

REDES DE PETRI Y ALGORITMOS GENÉTICOS, UNA PROPUESTA PARA LA PROGRAMACIÓN DE SISTEMAS DE MANUFACTURA FLEXIBLE*

Juan Pablo Caballero Villalobos**

Gonzalo Mejía***

Resumen: este artículo propone el uso conjunto de las redes de Petri y de los algoritmos genéticos como nuevo enfoque para modelar sistemas de manufactura flexible y generar programas de producción activos orientados a la minimización de la tardanza ponderada de los trabajos. Se consideraron algunas restricciones propias de este tipo de sistemas de producción, tales como tiempos de alistamiento dependientes de la secuencia de operaciones, estaciones con máquinas en paralelo no necesariamente idénticas y capacidad de almacenamiento temporal limitado en las estaciones. Las fortalezas de las redes de Petri y de los algoritmos genéticos para modelar sistemas discretos y solucionar problemas combinatorios, respectivamente, son ampliamente reconocidas y permiten considerar su integración como tema de interés en la programación de la producción. Se proporcionan resultados computacionales que hacen prever este enfoque como promisorio para futuros trabajos.

Palabras claves: Redes de Petri, programación de la producción, algoritmos genéticos, sistema de manufactura flexible, programas de producción activos, tardanza ponderada.

Abstract: this paper uses the Petri networks and genetic algorithms strengths for modelling and solving manufacturing systems and combinatorial problems, respectively. This article generates active schedules for flexible manufacturing systems that minimize weighted tardiness. The algorithm works for problems involving aspects such as: setups dependent on the operations sequences, stations with non identical parallel machines and stations buffers with finite capacity.

* Fecha de recepción: 14 marzo de 2006. Fecha de aceptación para publicación: 27 de abril de 2006.

** Ingeniero Industrial, Pontificia Universidad Javeriana. Magíster en Ingeniería Industrial, Universidad de los Andes. Profesor, Departamento de Procesos Productivos, Pontificia Universidad Javeriana, Bogotá, Colombia. Correo electrónico: juan.caballero@javeriana.edu.co

*** Ingeniero Mecánico, Universidad de los Andes. Magíster en Ingeniería Mecánica, Universidad de los Andes. Ph.D. in Industrial Engineering, Lehigh University. Profesor asistente, Departamento de Ingeniería Industrial, Universidad de los Andes, Bogotá, Colombia. Correo electrónico: gmejia@uniandes.edu.co

Some of the obtained computational results make as believe it could be considered as a promise field for future works.

Key Words: Petri Networks, production programming, genetic algorithms, flexible manufacturing systems, active schedules, weighted tardiness.

1. Introducción

En diversos sectores productivos, la programación de actividades reviste especial importancia por su incidencia en la productividad y en la competitividad de las empresas. De manera general, el problema de programación de producción que se estudia en este artículo corresponde a los sistemas de manufactura flexible (SMF), los cuales deben adaptarse rápidamente para producir gran diversidad de clases de productos en pequeñas y medianas cantidades. Estos sistemas agrupan sus recursos en unidades funcionales –estaciones– en donde se procesan diversas operaciones de los trabajos. Un SMF se compone de una serie de máquinas-herramientas de control numérico controladas por un computador, las cuales pueden procesar un grupo de trabajos. El sistema cuenta con un manejo de inventario en proceso y de materia prima controlado por computador, lo que permite balancear dinámicamente la utilización de los recursos y que el sistema se adapte automáticamente a los cambios en la producción, en la mezcla de productos y en los niveles de producción [Lee, DiCesare, 1994].

Para este tipo de sistema de manufactura es importante contar con una herramienta de modelado que describa apropiadamente sus características. Una de las herramientas de modelado que puede ser utilizada para este propósito son las Redes de Petri (RdP) las cuales “... permiten modelar fácilmente sistemas con eventos concurrentes, asíncronos, distribuidos, paralelos y/o estocásticos” [Murata, 1989]. En la literatura se encuentra un buen número de aplicaciones que han utilizado las RdP para modelar sistemas cuyos eventos tengan alguna o varias de las características ya mencionadas. Murata [1989] y Desrochers y Al-Jaar [1995] citan en sus trabajos una amplia variedad de aplicaciones en las cuales las RdP han sido utilizadas

exitosamente y mencionan por qué el modelado y análisis de SMF y de sistemas de control industrial se incluyen dentro de las áreas más prometedoras en su aplicación.

A pesar de sus evidentes ventajas para modelar SMF, las RdP no han sido muy utilizadas en problemas de programación (*scheduling*) debido al tamaño enorme de su espacio de estados para sistemas de tamaño real. Adicionalmente, la obtención de la programación de producción óptima en los sistemas de manufactura puede ser muy compleja o imposible de encontrar en tiempos razonables para dichos sistemas. Esta dificultad se evidencia claramente en el caso de los SMF, donde se reconoce por diversos autores que es un problema combinatorio clasificado como de tipo NP-Hard [Garey, Jonson 1979], [Stecke 1981].

Uno de los enfoques más exitosos para resolver este tipo de problemas es el de los algoritmos genéticos (AG). Los AG han sido ampliamente utilizados por su capacidad para brindar buenas soluciones a muchos de los problemas típicos de optimización combinatoria tales como asignación, transporte, agente viajero, corte de materiales (*cutting stock*), distribución de planta, pronósticos, clasificación de inventarios, planeación de la producción y programación de la producción. Según Chen et al., “para este tipo de problemas, los AG exploran eficientemente el espacio de soluciones, sin requerir enormes cantidades de memoria como otros algoritmos de búsqueda tradicionales como la búsqueda A*, ramificación y acotación (*Branch & Bound*) o la programación matemática” [Chen et al., 2001].

Este artículo presenta un nuevo enfoque de programación de producción para SMF que combina las ventajas de modelado de las RdP con la capacidad de los AG para resolver problemas de programación de producción en los SMF. En el artículo, después de efectuar la revisión bibliográfica sobre el tema, se presentan las bases de modelado de las RdP aplicadas a SMF y se propone la estructura cromosómica para el AG. A continuación se explican los fundamentos de la implementación del algoritmo desarrollado, denominado REPAG, basado en AG. Finalmente se presentan las pruebas realizadas y los resultados computacionales que muestran la validez del enfoque.

2. Revisión bibliográfica

El problema de programación de SMF ha sido estudiado desde variados enfoques que se pueden clasificar en las siguientes categorías: programación matemática [Aanen et al., 1993], [Sawik 1990]; inteligencia artificial y sistemas expertos [Bensana et al., 1988], [Chryssolouris et al., 1988]; inteligencia artificial - redes neuronales [Jain 2001], [Sabuncuoglu, 1998]; técnicas de simulación y reglas de despacho [Li et al., 1997], [Min et al., 1998]; lógica difusa y reglas de despacho [Custodio et al., 1994], [Grabot, Geneste, 1994]; búsqueda tabú [Basnet, 1996], [Logendran, Sonthinen, 1997]; métodos heurísticos [Kimemia, Gershwin, 1983], [Liu, MacCarthy, 1999]. A pesar del evidente interés que despierta este problema, la programación de SMF es aún un problema abierto. Los anteriores enfoques presentan dificultades bien sea en capturar todas las características salientes de dichos sistemas o en presentar métodos de solución eficientes [Lee, DiCesare, 1994].

Los AG han sido ampliamente utilizados para la programación de producción (*scheduling*), principalmente en sistemas tipo taller (*job shop*). Los enfoques utilizados para abordar este problema mediante los AG pueden ser clasificados esencialmente por la estructura utilizada en la representación de los cromosomas [Gen, Cheng, 1997], [Ponnambalam, Aravindan, 2001]. Las categorías identificadas por dichos autores basan la representación de los cromosomas en los siguientes aspectos: en la representación común de grafo disyuntivo [Pinedo, Chao, 1999], en la relación entre pares de trabajos, en el tiempo de finalización de los trabajos, en las máquinas, en las operaciones, en los trabajos, en listas de preferencia y en reglas de despacho, entre otros. Como se explica más adelante, la representación más relevante por su simplicidad y aplicabilidad al problema por tratar es la basada en llaves aleatorias, en la que los individuos están constituidos por una cadena de tantos genes como número de operaciones se han de programar. Cada gen está compuesto por varios números enteros que representan, según el enfoque, la prioridad, orden o máquina donde deben programarse los trabajos [Ponnambalam, Aravindan, 2001]. Este tipo de representación tiene la ventaja de que cualquier individuo generado aleatoriamente codifica una solución factible al problema y que cada individuo generado como producto de la aplicación de los operadores genéticos también codifica una solución factible. Algunos autores han trabajado este tipo de representación [Fang, Ross, Cone, 1993], [Gonçalves, Mendes, Resende, 2002].

Otros autores han utilizado RdP para el modelaje de SMF y han aplicado un método heurístico para su programación. Lee y DiCesare [1994], así como Lloyd y Conostas [1995], utilizaron una búsqueda de ramificación y acotación para minimizar el tiempo de flujo total (*makespan*). Saitou et al. [2002] utilizaron RdP para modelar la asignación de los recursos y controlar la programación de la producción con el objeto de lograr programas robustos en SMF. Tiwari et al. [2001] modelan mediante RdP temporizadas el sistema de vehículos guiados automáticamente de un SMF para asegurar que no se presenten conflictos y bloqueos en el sistema. Jain [2001] propone un *software* controlador de asignación de recursos para SMF usando RdP y reglas de despacho. En los trabajos mencionados, las RdP se han utilizado para modelar los SMF o controlar su evolución. La programación de los trabajos en el SMF se realiza de manera independiente y sin tener la representación de la red de Petri. Este último aspecto diferencia los trabajos encontrados en la revisión bibliográfica con respecto al enfoque presentado en este artículo.

A continuación se explican los fundamentos del modelado y de la implementación del algoritmo REPAG basado en AG.

3. Fundamentos

3.1. Las Redes de Petri (RdP)

Las RdP son una clase de grafo dirigido y bipartito, conformado por dos tipos de nodos denominados plazas y transiciones. Las plazas representan, por lo general, acciones o condiciones y las transiciones representan eventos. Las plazas y las transiciones se conectan mediante arcos. En las plazas residen entidades llamadas *tokens* o puntos negros, que representan la verdad de la condición o acción asociada con la plaza. El estado de la RdP se determina por un vector M de enteros no negativos denominado marcaje que contiene el número de *tokens* que se encuentran en cada sitio de la RdP.

Formalmente la RdP se define como una:

$$PN=6\text{-tupla } \{ P, T, \{ T \times P \}, \{ P \times T \}, M_0, W \} \quad (1)$$

Donde:

P: conjunto de plazas de la RdP

T: conjunto de transiciones de la RdP

$T \times P$: conjunto de arcos de salida de las transiciones a las plazas

$P \times T$: conjunto de arcos de entrada de las plazas a las transiciones

M_0 : arcaje inicial de la RdP

W: conjunto de pesos asociados a los arcos

La evolución de una RdP hace referencia al movimiento de los *tokens* en la red, que define el vector de marcaje y sus cambios y que debe cumplir las condiciones estipuladas por la regla de disparo de transiciones. Una transición t está habilitada si el número de *tokens* en sus plazas de entrada es mayor o igual al peso del arco conectando la plaza con la transición.

3.2. Los algoritmos genéticos

Según Goldberg, "...los Algoritmos Genéticos son técnicas de búsqueda estocástica basados en los mecanismos de evolución natural" [Goldberg, 1989]. Los AG utilizan un conjunto de individuos –por ejemplo, soluciones– sobre el cual se aplican operadores y estrategias para encontrar soluciones mejoradas al problema. El conjunto de individuos que usa el AG se denomina población. Los AG no buscan una única solución, sino que están orientados a que, en promedio, los individuos de la población busquen una mejora en la función objetivo de las soluciones, de manera que en posteriores generaciones se enriquezcan (evolucionen), preservando sólo las mejores características de los individuos de generaciones anteriores.

Un AG es una técnica iterativa basada en una población, que usa una estrategia de selección y operadores –esencialmente recombinación y mutación–, para generar y explorar nuevos puntos en el espacio de soluciones que originen una nueva población, la cual estará constituida por un conjunto de individuos denominados cromosomas que están compuestos por elementos menores

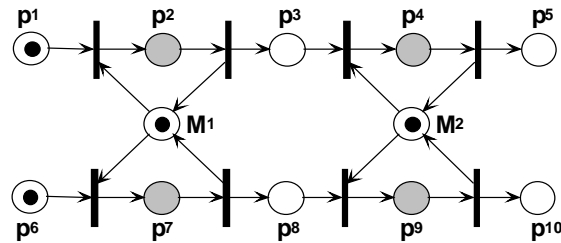
llamados genes, los cuales codifican las instrucciones, parámetros y demás elementos usados para construir una solución al problema por solucionar.

Para la conformación de cada nueva generación, se utiliza una estrategia de selección que estipula las reglas y condiciones para constituir una población intermedia a partir de los cromosomas de la población actual. Esta población intermedia es usada para la obtención de nuevos individuos mediante el uso de la recombinación y de algunos operadores genéticos – mutación, inversión y otros más especializados según las características del problema afrontado. Para evaluar la conveniencia de un cromosoma se utiliza la función de adaptación, que calcula el valor de la función objetivo correspondiente a la solución codificada en él. Este valor puede ser el resultado de la evaluación de la función objetivo o de una estimación parcial de la misma; por este motivo cobra especial importancia que se pueda calcular rápidamente usando la representación del cromosoma.

4. El modelado de SMF con Redes de Petri

En este artículo se utiliza la metodología descrita por Zhou y DiCesare [1993] para modelado de sistemas de manufactura usando RdP. En esencia la RdP consiste de un número (n) de sub-redes acíclicas que modelan las relaciones de precedencia de los n trabajos asignados al sistema de manufactura. Los recursos se añaden como plazas que representan condiciones para el procesamiento de dichos trabajos. La RdP contiene plazas temporizadas y no temporizadas: las primeras modelan las operaciones de los trabajos, mientras que las segundas lo hacen con las colas de almacenamiento temporal (*buffers*) localizadas antes de las estaciones. Las transiciones modelan bien sea el inicio o el fin de una operación. La disponibilidad de las máquinas es modelada mediante plazas sin temporizar. Estas plazas, denominadas de recurso, están conectadas a las transiciones de las subredes de los trabajos y controlan que el correspondiente recurso no pueda estar utilizado por más de una máquina. La Figura 1 ilustra el modelado con RdP. Las plazas en color gris son temporizadas. Las plazas M1 y M2 representan la disponibilidad de las máquinas 1 y 2 respectivamente. En la Tabla 1 se describe cada una de las plazas de la Figura 1.

Figura 1. Red de Petri que modela dos trabajos y dos máquinas



Fuente: presentación propia de los autores.

Tabla 1. Descripción de plazas en la RdP de la Figura 1

p ₁	Trabajo 1 disponible	p ₆	Trabajo 1 disponible
p ₂	Trabajo 1 procesado por la máquina 1	p ₇	Trabajo 2 procesado por la máquina 1
p ₃	Operación 1 de trabajo 1 terminada	p ₈	Operación 1 de trabajo 2 terminada
p ₄	Trabajo 1 procesado por la máquina 2	p ₇	Trabajo 2 procesado por la máquina 2
p ₅	Operación 2 de trabajo 1 terminada	p ₈	Operación 2 de trabajo 2 terminada
M ₁	Máquina 1 disponible	M ₂	Máquina 2 disponible

Fuente: presentación propia de los autores.

5. REPAG (Redes de Petri y Algoritmos Genéticos)

REPAG es un algoritmo que utiliza las fortalezas de las RdP y los AG para modelar sistemas de manufactura y solucionar el problema de programación en los SMF. REPAG implementa un AG que utiliza la estructura de la RdP asociada con el SMF para obtener programas de producción factibles, es decir, que en ellos se cumplen todas las restricciones de precedencia de los trabajos y de capacidad de las máquinas. El AG utiliza un cromosoma que codifica una secuencia de disparos de transiciones que transforma el marcaje inicial hasta alcanzar el marcaje final deseado. La evolución del marcaje inicial de la RdP define implícitamente el programa de producción. El

criterio de desempeño utilizado para medir la conveniencia de la programación resultante es la tardanza ponderada de los trabajos. El aporte fundamental del enfoque propuesto es proveer una herramienta para programación de SMF que, por una parte, es capaz de manipular las características salientes de dichos sistemas y, por otra, supera en desempeño a otros algoritmos y reglas de despacho.

5.1. El problema

El problema de estudio en este artículo es programar la producción de un número n de trabajos en un SMF con el fin de minimizar el objetivo de la tardanza ponderada total T . La definición de SMF utilizada se enuncia a continuación:

Índices:

i trabajos, j operaciones, k máquinas, e estaciones

Conjuntos:

M máquinas, E estaciones, J trabajos,

M_e máquinas de la estación e ,

$O_{i,e}$ operaciones del trabajo i que deben ser procesadas en la estación e

Parámetros:

$a_{j,i,k}$: tiempo de alistamiento de la operación j del trabajo i en la máquina k

d_j : fecha de entrega o de compromiso del trabajo j

$p_{j,i,k}$: tiempo de proceso de la operación j del trabajo i en la máquina k

Un SMF se compone de un conjunto de máquinas M y de un conjunto de trabajos J que serán programados. Cada máquina pertenece a un único subconjunto M_e de máquinas de acuerdo con su funcionalidad que forman estaciones. Cada trabajo i se compone de un conjunto de operaciones $O_{i,e}$, cada una de las cuales debe procesarse en la estación e , de manera ininterrumpida en cualquiera de las máquinas del subconjunto de máquinas M_e de la estación e . El tiempo de proceso de las operaciones de los trabajos puede variar dependiendo de la máquina en la cual se programe, es conocido con anterioridad y se identifica como $p_{j,i,k}$. El tiempo de

alistamiento en las máquinas depende de la secuencia de operaciones o de la configuración inicial de las máquinas y se denota por $a_{j,i,k}$.

Los supuestos del problema son los siguientes:

1. Cada trabajo j tiene una fecha de compromiso o de entrega d_j .
2. Puede existir recirculación de trabajos en las estaciones.
3. Las máquinas agrupadas en las estaciones no son necesariamente iguales, por lo que los tiempos de proceso de las operaciones en las diferentes máquinas de una estación pueden ser diferentes.
4. Cada máquina está disponible desde tiempo cero.
5. Las máquinas no presentan fallas.
6. El procesamiento de una operación no puede ser interrumpido y eliminado de la máquina correspondiente antes de que finalice.
7. Se programan trabajos conocidos en tiempo cero y no llegan nuevos trabajos; sin embargo, se pueden programar trabajos cuya disponibilidad sea posterior al tiempo cero.
8. Los tiempos de alistamiento son determinísticos y dependientes de la secuencia de operaciones.
9. Los tiempos de alistamiento son iguales para todas las máquinas que pertenecen a una estación.

5.2. El funcionamiento de REPAG

El funcionamiento de REPAG se puede esquematizar mediante los siguientes pasos:

1. Crear una población inicial de individuos generados de forma aleatoria y calcular su valor de adaptación con el criterio de la tardanza ponderada de los trabajos, T .
2. Si hay k individuos idénticos en la población se penaliza el valor de adaptación de $k-1$ de ellos con un valor muy grande para evitar que existan genes dominantes que puedan causar convergencia prematura.

3. Asignar la probabilidad de ser seleccionado a cada uno de los individuos de la población proporcionalmente al inverso del valor de adaptación.
4. Seleccionar parejas de individuos de la población actual según las probabilidades calculadas en el paso 3, aplicar la estrategia de reproducción para generar los hijos y calcular su valor de adaptación con el criterio de tardanza ponderada.
5. Escoger los mejores cromosomas entre padres e hijos, es decir, aquellos que tengan los menores valores de adaptación.
6. Escoger los genes de los individuos de la población que serán mutados, de acuerdo con la probabilidad de mutación fijada en el algoritmo.
7. Si se ha alcanzado el número de generaciones definido como criterio de detención, parar y devolver la población actual; en caso contrario, regresar al paso 2.

La RdP con la cual REPAG modela el SMF tiene asociada una matriz de incidencia de $2 \sum_{i \in \mathbf{J}} \sum_{e \in \mathbf{E}} \|\mathbf{O}_{i,e}\| \|\mathbf{M}_e\|$ filas por $\|\mathbf{E}\| + \|\mathbf{M}\| + \|\mathbf{J}\| + \sum_{i \in \mathbf{J}} \sum_{e \in \mathbf{E}} \|\mathbf{O}_{i,e}\| \|\mathbf{M}_e\|$ columnas, donde cada fila corresponde a los cambios que se deben realizar sobre el marcaje si esa transición es disparada. Las primeras $\|\mathbf{E}\|$ columnas de la matriz de incidencia representan las plazas de almacenamiento temporal de las estaciones del SMF. Las siguientes $\|\mathbf{M}\|$ columnas están asociadas a los recursos del SMF. Las $\|\mathbf{J}\|$ columnas siguientes corresponden a las plazas de finalización de cada uno de los trabajos programados en el SMF. Finalmente, las últimas columnas están vinculadas a las plazas que representan el procesamiento de un trabajo en algún recurso.

5.3. La estructura del cromosoma

Los cromosomas utilizados en REPAG se definen como una secuencia de un número de genes igual al número total de operaciones de los trabajos por programar en el sistema. Cada gen se compone de dos alelos (g_1, g_2) que son números enteros, cuyos valores posibles se encuentran en un rango entre 1 y un número suficientemente grande como para garantizar que cualquier máquina en cualquier estación pueda ser seleccionada para procesar cualquier operación dada.

Este número suficientemente grande debe ser al menos el mayor de los siguientes números: número de trabajos a programar, máximo número de operaciones de cualquier trabajo por programar y máximo número de máquinas en cualquier estación del sistema. Cada gen representa de forma codificada un trabajo inconcluso y la máquina correspondiente a la siguiente operación que se realizará en dicho trabajo. Esto es equivalente a una transición que se debe disparar en la RdP para iniciar una de las operaciones disponibles.

El cromosoma se va interpretando de manera secuencial y el trabajo cuya operación se va a programar se escoge de la lista de trabajos inconclusos (aquellos que al menos tengan una operación por programar), de acuerdo con el valor obtenido como residuo entero de la división entre el primer alelo del gen que se está interpretando y el número de trabajos inconclusos.

La transición a disparar en la RdP implica la selección de una máquina donde puede programarse la operación del trabajo ya seleccionado. Esta transición se escoge de la lista de transiciones utilizando como criterio el residuo entero de la división entre el segundo alelo del gen que se está interpretando y el número de transiciones o máquinas de la estación donde se debe procesar esa operación. Este número obtenido identifica de manera única la transición que se encuentra en dicha posición en la lista de posibles transiciones por disparar para la operación del trabajo seleccionado mediante la interpretación del primer número del gen.

Expresado formalmente, si al momento de interpretar un gen determinado, se tiene un conjunto P de trabajos inconclusos, se escoge el m -ésimo trabajo de P si:

$m = \frac{g_1}{\|P\|} + 1$ y se programa en la r -ésima máquina de la estación e , disparando en la RdP la

transición asociada si: $r = \frac{g_2}{\|O_e\|} + 1$

5.4. La función de adaptación

La función objetivo utilizada en REPAG es la minimización de la tardanza ponderada total de los trabajos por programar. Dicha medida de desempeño se simboliza mediante T y se define como

la suma ponderada de los retrasos de un conjunto de trabajos. Este indicador, según Pinedo y Chao [1999], ofrece la ventaja de permitir priorizar los trabajos más urgentes sobre los que no son tan críticos, penalizando la función objetivo, tanto por el número de periodos de tiempo adicional que debe usar cada trabajo para su finalización después de la fecha de entrega establecida, como por el factor de prioridad asignado al trabajo.

$$T = \sum_{i \in J} w_i \max \{0, (c_i - d_i)\} \quad (2)$$

Donde c_i y w_i representan el tiempo en el cual se finaliza el trabajo i y el peso o importancia asignada al trabajo i , respectivamente.

El registro de los tiempos en el que finaliza un trabajo, se calcula usando un vector que contiene los tiempos del último arribo de un *token* a cada plaza en la RdP.

Cuando se dispara una transición t asociada a una máquina k , se actualiza la variable $t_{sistema}$ de la siguiente forma:

$$t_{sistema} = \max \left\{ \max \{a_{j,i,k} + p_{j,i,k}\}, t_{sistema} \right\} \quad (3)$$

Donde:

t^* : mayor tiempo de alistamiento de los recursos (máquinas) necesarios para procesar la operación

Una vez el tiempo del sistema es actualizado, se asigna su valor a las plazas de salida de la transición t en el vector de tiempos. Es de notar que el valor de t^* depende de la anterior operación programada en cada recurso o del alistamiento inicial del mismo.

Las entradas del vector de tiempos correspondientes a las plazas, asociados a la terminación de cada trabajo en la matriz de incidencia de la RdP del SMF, contienen el tiempo en el cual se finaliza el trabajo.

5.5. La estrategia de selección

La selección de un individuo de la población para la recombinación se realiza mediante un mecanismo de ruleta en el cual los individuos con mejor valor de adaptación y menor tardanza ponderada total tienen mayor probabilidad de ser escogidos. Estas probabilidades son calculadas y asignadas a cada individuo antes de empezar el proceso de selección.

5.6. La recombinación

REPAG utiliza la recombinación de un punto para crear dos hijos de cada pareja de padres seleccionados. Para cada pareja de padres se selecciona un punto de intercambio cuyo valor está en el rango de 1 a la longitud del cromosoma, que sirve para segmentar los cromosomas en dos partes e intercalarlos para la obtención de los hijos.

REPAG utiliza la recombinación de un punto para minimizar el efecto de ruptura de esquemas en los cromosomas durante la aplicación de la recombinación. Este efecto hace referencia al teorema fundamental de los AG, el Teorema del Esquema, y a la probabilidad de separación de genes cuyo valor es muy importante dentro del cromosoma cuando se aplica este operador [Whitley, 1993].

5.7. La mutación

El mecanismo de mutación implementado en REPAG selecciona aleatoriamente un número de genes que corresponde a un porcentaje fijo del número total de genes de la población. Los valores de los genes seleccionados se reemplazan por nuevos valores enteros generados de manera aleatoria y que pertenecen al rango válido ya mencionado en la sección de estructura del cromosoma.

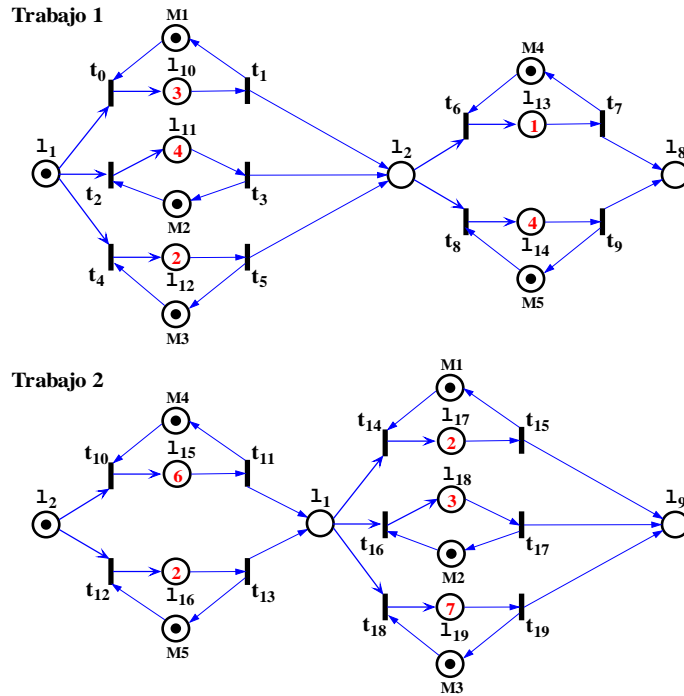
5.8. El constructor de programas de producción

Para la construcción de un programa de producción a partir de un cromosoma de REPAG se debe interpretar cada uno de los genes para identificar y disparar la transición t correspondiente. Una vez se dispara esta transición t (correspondiente tanto al inicio de una operación como a la adquisición de recursos requeridos por dicha operación), se agrega la transición $t+1$ a una lista de transiciones pendientes por disparar ($t+1$ corresponde a la finalización de una operación). Al disparar $t+1$ se liberan los recursos (máquinas) requeridos por la operación. Para poder disparar cualquier transición t se comprueba si los recursos que necesita para ser disparada se encuentran disponibles en la RdP. Si algunos de éstos recursos no se encuentran disponibles, REPAG inspecciona la lista de transiciones pendientes por disparar y dispara aquellas que liberan los recursos que se necesitan para disparar t , generando así secuencias de disparos de transiciones válidas.

5.9. Un caso de ejemplo

A manera de ejemplo de la implementación de REPAG, en la Figura 2 se muestra la RdP asociada a un SMF de dos estaciones, con tres y dos máquinas, y dos trabajos por programar con tiempos de entrega 12 y 10 respectivamente.

Figura 2. Ejemplo de modelaje de un SMF con REPAG



Fuente: presentación propia de los autores.

Los dos trabajos tienen dos operaciones de diferentes tipos (A, B ó C) según el tipo de alistamiento requerido. La primera operación del trabajo uno es de tipo A y debe procesarse en la estación 1; la segunda operación es de tipo A y debe procesarse en la estación 2. La primera operación del trabajo dos es de tipo B y debe procesarse en la estación 2 mientras la segunda operación es de tipo C y debe procesarse en la estación 1. Los tiempos de proceso asociados a cada plaza se indican en su interior. El alistamiento inicial de las máquinas es el siguiente: Máquina 1 C, Máquina 2 A, Máquina 3 B, Máquina 4 C, Máquina 5 B. Los tiempos de alistamiento son los siguientes: (A,B,2) (A,C,2) (B,A,3) (B,C,1) (C,A,1) (C,B,2). En la Tabla 2 se relaciona la lista de transiciones candidatas para procesar cada operación en el ejemplo propuesto.

Tabla 2. Transiciones candidatas

Operación	Transiciones candidatas
T1 - Op1	t_0, t_2, t_4
T1 - Op2	t_6, t_8

T2 - Op1	t_{10}, t_{12}
T2 - Op2	t_{14}, t_{16}, t_{18}

Fuente: presentación propia de los autores.

Para este ejemplo, $\bar{N} = \max\{2,2,3\} = 3$ es el valor máximo entre el número de trabajos por programar, el mayor número de operaciones que componen un trabajo y el mayor número de máquinas que componen una estación. Un cromosoma válido para este SMF es (1,2) (2,1) (1,1) (2,3) y su interpretación se muestra en la Tabla 3.

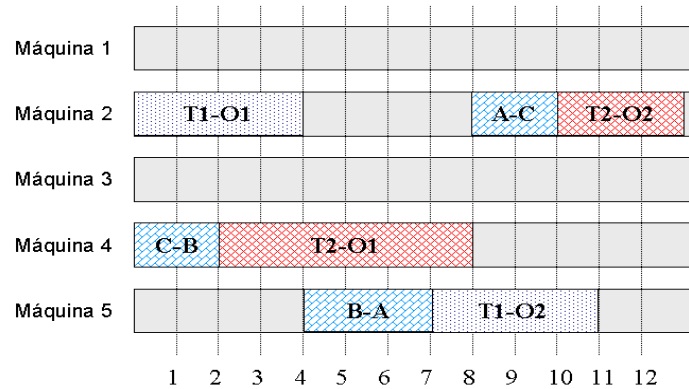
Tabla 3. Interpretación de los genes del cromosoma REPAG

Gen	Trabajos inconclusos	Trabajo escogido	Operación a programar	Transición seleccionada	Disparos
(1,2))	T1, T2	$(1 \bmod 2) + 1 = 2$	T2-O1	$(2 \bmod 2) + 1 = 1$	t10
(2,1))	T1, T2	$(2 \bmod 2) + 1 = 1$	T1-O1	$(1 \bmod 3) + 1 = 2$	t2
(1,1))	T1, T2	$(1 \bmod 2) + 1 = 2$	T2-O2	$(1 \bmod 3) + 1 = 2$	t3 , t11 , t16
(2,3))	T1	$(2 \bmod 1) + 1 = 1$	T1-O2	$(3 \bmod 2) + 1 = 2$	t8
Finalmente					t9 , t17

Fuente: presentación propia de los autores.

De lo anterior resulta la secuencia de disparos de transiciones $t_{10} . t_2 . t_3 . t_{11} . t_{16} . t_8 . t_9 . t_{17}$. Esta secuencia de disparos de transiciones produce la programación que se muestra en la Figura 3.

Figura 3. Programación de operaciones de ejemplo de modelaje de un SMF con REPAG



Fuente: presentación propia de los autores.

6. Resultados computacionales y análisis

Con el fin de comparar el desempeño de REPAG se utilizó un conjunto de veinticinco problemas generados aleatoriamente. Los valores de los parámetros de estos problemas se obtuvieron usando distribuciones uniformes discretas. Los parámetros que describen la configuración del taller se encuentran en un archivo que contiene la información correspondiente al número de estaciones en el taller, el número de máquinas en cada estación, los tiempos de alistamiento dependientes de la secuencia de proceso para cada pareja de operaciones y el tipo de configuración inicial en cada una de las máquinas. De igual manera se especifica el número de trabajos a programar, el tiempo de disponibilidad de cada trabajo, el tiempo de entrega último establecido para el trabajo, el peso o importancia de finalizar el trabajo antes de la fecha última establecida, la ruta que debe seguir el trabajo dentro del taller, el tiempo de proceso en cada estación y el tipo de operación. En la Figura 4 se ilustra parte del formato típico de uno de los problemas generados.

Figura 4. Estructura de archivo de configuración del taller y de sus trabajos

Problema 1068499577

8 Estaciones, 16 Mquinas, 20 Trabajos (id_estacin,mquinas)

(1,1) (2,2) (3,3) (4,3) (5,2) (6,2) (7,2) (8,1)

Estacin 1 (tipo_anterior, tipo_siguiete, t_alistamiento)

(A,B,2) (A,C,2) (A,D,1) (A,E,5) (B,A,1) (B,C,5) (B,D,1) (B,E,5) (C,A,4) (C,B,4)

(C,D,1) (C,E,1) (D,A,5) (D,B,1) (D,C,4) (D,E,4) (E,A,3) (E,B,2) (E,C,3) (E,D,3)

Alistamiento inicial de mquinas de la estacin (id_mquina,tipo)

(1,E)

.
.
.

Trabajos: (id,t_disp,t_entrega,peso) Operaciones (estacion,t_proceso,tipo)

(1,0,87,2) (8,9,C) (2,9,B) (7,10,C) (4,4,E) (1,6,E) (6,1,E) (5,4,C) (3,8,A)

(2,0,86,4) (4,6,C) (1,7,C) (8,1,D) (6,5,E) (3,3,A) (5,6,A) (7,10,B) (2,6,E)

(3,0,89,1) (8,9,E) (6,9,A) (5,9,E) (2,1,D) (1,7,A) (7,9,A) (3,2,D) (4,3,C)

(4,0,92,6) (4,6,C) (3,8,B) (5,5,D) (7,5,E) (8,3,B) (2,10,D) (1,9,C) (6,7,A)

(5,0,97,5) (1,0,D) (6,10,D) (0,2,C) (1,1,D) (5,0,E) (7,10,D) (2,10,D) (2,1,E)

.
.
.

Fuente: presentación propia de los autores.

La configuración del taller y el tamaño de los problemas usados para la comparación y análisis del desempeño de REPAG se compendian en la Tabla 4. En dicha tabla la columna tipos de operación contiene el número de categorías consideradas para calcular el tiempo de alistamiento en función de la secuencia de tipos de operaciones consecutivas.

Tabla 4. Dimensión de los problemas usados para el análisis de REPAG

Problema	Estaciones	Máquinas	Trabajos	Tipos de Operación	Tiempo de ejecución promedio (segundos)*	
					Heurística cuello de botella	REPAG
99577	8	16	20	5	113	215
00662	8	17	20	5	136	216
06684	10	20	20	5	216	328
11473	6	12	25	5	98	395
27215	6	17	25	5	107	398
11538	7	11	25	5	113	525
11485	7	12	25	5	142	509
11486	7	13	25	5	156	517
06657	7	14	25	5	197	250
11541	7	14	25	5	168	503
06665	7	17	25	5	266	240
21688	7	17	25	5	309	248
98068	8	13	25	5	165	339
06523	8	15	25	5	226	329

06564	8	17	25	5	332	327
06644	10	13	25	5	164	521
21291	10	14	25	5	205	542
21177	10	15	25	5	292	825
06581	10	15	25	5	246	532
05520	5	10	30	5	114	382
01466	5	11	30	5	392	182
01986	5	13	30	5	337	185
21615	6	12	30	5	242	278
21610	6	12	30	5	492	260
21592	8	12	30	5	329	492

* Los tiempos de ejecución promedio fueron obtenidos en un computador Pentium III de 450 MHz con 128 Mb de memoria RAM con sistema operativo Windows 2000. No se listan los tiempos correspondientes a la ejecución de las reglas de despacho debido a que presentan gran homogeneidad para todos los problemas y son cercanos a un quinto de segundo.

Fuente: presentación propia de los autores.

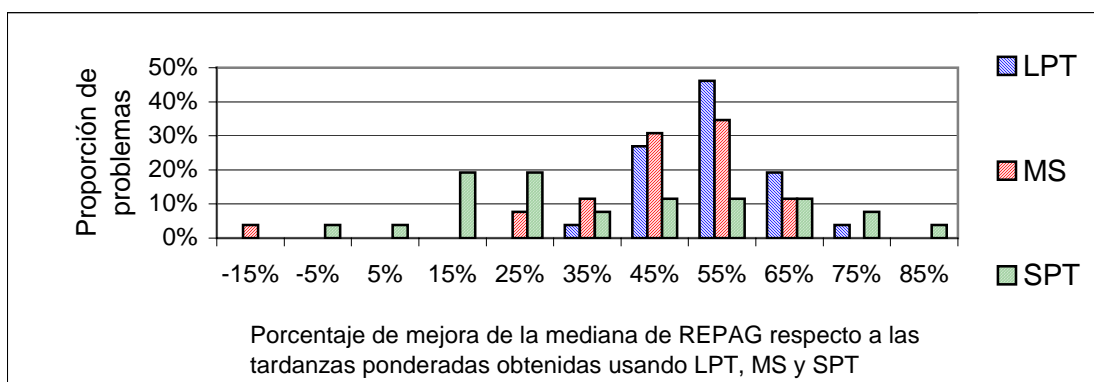
Para cada uno de los problemas se obtuvo la programación de sus trabajos utilizando el algoritmo general de cuello de botella (SB) para tardanza ponderada y algunas reglas de despacho implementadas en el software LEKIN® [Pinedo, Chao, 1999]. Las reglas de despacho utilizadas como parámetro de comparación son las siguientes: *Longest Processing Time (LPT)*, *Minimum Slack (MS)*, *Short Processing Time (SPT)*, *Weighted Shortest Processing Time (WSPT)*, *Apparent Tardiness Cost with Setups (ATCS)*.

Para determinar la estrategia de evolución, la estrategia de selección y el tipo de recombinación más conveniente y poder calibrar el modelo, se realizó un diseño de experimentos 2 factorial con bloques para determinar la significancia de cada uno de los factores y sus interacciones. Los factores significativos (al 95% de confianza) fueron el tamaño de la población, el porcentaje de mutación y el tipo de recombinación utilizado. Cada uno de los problemas fue resuelto en REPAG en diez ocasiones con los siguientes parámetros: población de cien individuos, cincuenta hijos en cada generación, escogencia de mejores entre padres e hijos como estrategia de evolución, torneo como estrategia de selección, porcentaje de mutación de 0,5% del total de genes de la población, recombinación de un punto y mil generaciones.

Del conjunto de tardanzas ponderadas obtenidas por REPAG para cada uno de los problemas, se calculó la mediana y los valores mínimos y máximos, para describir de mejor forma el comportamiento de los datos y comparar de forma más clara estos resultados con los obtenidos

con las reglas de despacho y el algoritmo de cuello de botella. Estos resultados se compendian en las figuras 5, 6 y 7, en dónde se muestra la proporción de los problemas y el porcentaje en el cual el valor de la mediana de REPAG mejora los valores obtenidos al usar los otros métodos. En la Figura 5 se observa que en un alto número de los problemas el valor de mediana obtenido por REPAG produce mejoras importantes respecto a las tres reglas de despacho y que sólo en pocos casos estas reglas de despacho han producido mejores valores que REPAG.

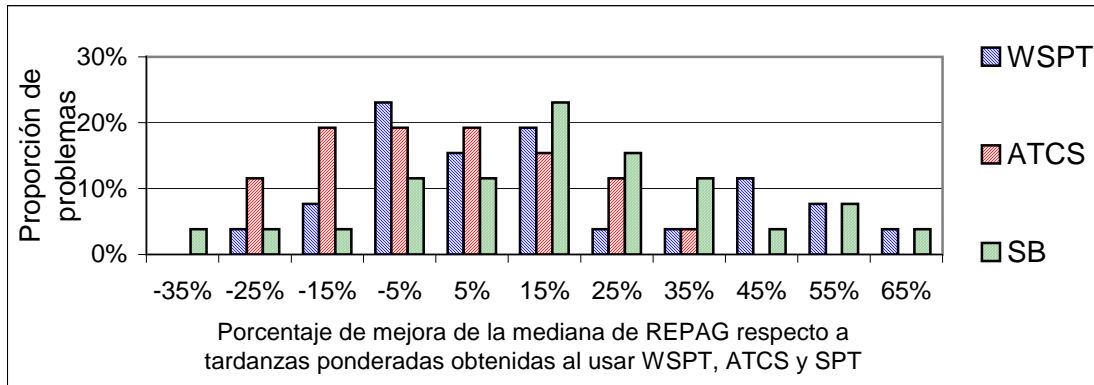
Figura 5. Comparativo de los resultados de REPAG con los de reglas de despacho LPT, MS y SPT



Fuente: presentación propia de los autores.

La Figura 6 muestra que existe una buena proporción de los problemas para los cuales el valor de la mediana obtenido por REPAG representa una mejora porcentual interesante sobre los resultados proporcionados por estas reglas de despacho, a pesar de que se encuentren problemas para los cuales no se presenten mejora.

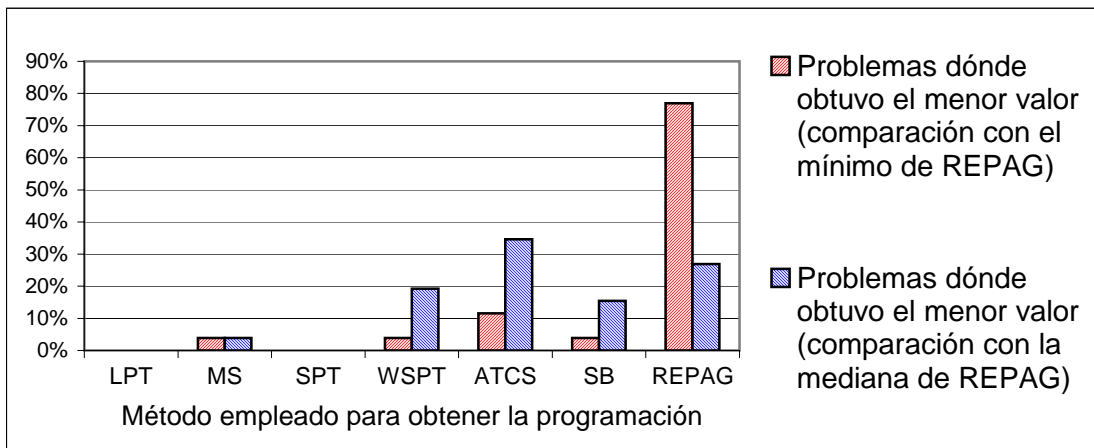
Figura 6. Comparativo de los resultados de REPAG con los de reglas de despacho WSPT, ATCS y SB



Fuente: presentación propia de los autores.

La Figura 7 permite observar conjuntamente y comparar los resultados al utilizar tanto el valor mínimo como la mediana obtenidos por REPAG respecto a los que se obtienen con las reglas de despacho y el algoritmo cuello de botella.

Figura 7. Comparativo de los resultados obtenidos con REPAG con los obtenidos mediante otras reglas de despacho



Fuente: presentación propia de los autores.

En la Figura 7 se observa que en cerca del 80% de los problemas se obtuvieron los mejores valores de tardanza ponderada con REPAG con lo cual su utilización resulta prometedora si se tiene en cuenta que las características de los problemas que puede manipular REPAG lo acercan más a la realidad.

La configuración de los problemas utilizados se estableció de acuerdo con las siguientes reglas: número de estaciones del SMF (establecido por el usuario) y el número de trabajos (establecido por el usuario). Los demás parámetros son números enteros generados aleatoriamente en un rango determinado por parámetros que son ingresados en el momento de generar el problema y que se relacionan entre llaves cuadradas. Número de máquinas en cada estación [mínimo_número_máquinas, máximo_número_máquinas]; peso máximo asignado a la terminación a tiempo de los trabajos [1, máxima_importancia]; número de clases de operaciones para contemplar alistamiento [mínimo_clases_operaciones, máxima_clases_operaciones]; tiempo de proceso de las operaciones en cada estación [1, máximo_tiempo_proceso]; fecha de entrega de los trabajos [90% tiempo_proceso_mínimo, 110% tiempo_proceso_mínimo]; clasificación de cada operación para contemplar el alistamiento [1, número_clases_operaciones]; tipo de alistamiento inicial de las máquinas en cada estación [1, número_clases_operaciones].

7. Conclusiones y futuras investigaciones

El uso conjunto de las RdP y AG para modelar y optimizar la programación de la producción en un SMF muestra resultados prometedores tal como se muestra en los resultados reportados. El modelaje del SMF mediante RdP permite capturar sus características salientes en una forma clara y concisa. Por ejemplo, el modelo de RdP permite representar recirculación de trabajos, máquinas en paralelo no necesariamente idénticas en las estaciones, tamaño de almacenamiento temporal limitado en las estaciones, diferentes tiempos de disponibilidad de los trabajos y el uso de más de un recurso en el procesamiento de las operaciones.

REPAG utiliza el modelo de las RdP para la programación de tareas basado en AG. El cromosoma propuesto en REPAG representa indirectamente la secuencia de disparos de transiciones en la RdP utilizada para modelar un SMF. Lo anterior permite que la programación del SMF pueda ser controlada a través de esta secuencia de disparos de transiciones obtenidas por REPAG. Este cromosoma permite generar todo el espacio posible de soluciones, garantiza siempre una solución factible después de ser recombinado o mutado y no necesita de estrategias de reparación.

El desempeño de REPAG depende en gran medida de la sofisticación usada en su implementación y de la calibración de los parámetros que se utilicen. En consecuencia, el mecanismo de construcción de los programas activos es un punto clave para disminuir el tiempo de ejecución de REPAG cuando se consideran tiempos de alistamiento dependientes de las secuencias de operaciones; esta consideración debe hacerse porque es muy frecuente que, teniendo un programa activo al programar una nueva operación, ésta modifique los valores de los tiempos de alistamiento y conduzca a programas no activos.

Los resultados computacionales muestran que este enfoque híbrido (RdP y AG) es una alternativa interesante para el problema de programación de sistemas flexibles de manufactura. Los resultados muestran que REPAG compite favorablemente con otros algoritmos para programación de estos sistemas.

Como continuación de este trabajo se sugiere estudiar el uso de este enfoque para afrontar el problema de reprogramación en SMF y la consideración de casos donde se presenten tiempos de alistamiento dependientes e independientes de la secuencia de operaciones.

Referencias

- Aanen, E., Gaalman, G.J., Naijn, W.M. "A Scheduling Approach for a Flexible Manufacturing System". En: *International Journal of Production Research*, 31(10), 1993, 2369-2385.
- Basnet, C. "Tabu Search Heuristic for a Loading Problem in Flexible Manufacturing Systems". En: *International Journal of Production Research*, 34, 1996, 1171-1174.
- Bensana, E., Bel, G., Dubois, D. "A Multi-Knowledge-Based System for Industrial Job Shop Scheduling". En: *International Journal of Production Research*, 26, 1988, 795-819.
- Chen, J.-H., Fu, L.-C., Lin, M.-H., Huang, A.-C. "Petri Net and GA-Based Approach to Modeling, Scheduling, and Performance Evaluation for Wafer Fabrication". En: *IEEE transactions on robotics and automation*, 17(5), 2001, 619-636[U1]

- Chryssolouris, G., Wright, K., Pierce, J., Cobb, W. "Manufacturing Systems Operations: Dispatch Rules vs. intelligent Control". En: *Robotics and Computer Integrated Manufacturing*, 4, 1988, 531-544.
- Custodio, L.M.M., Sentieiro, J.J.S., Bispo, C.F.G. "Production Planning and Scheduling using a Fuzzy Decision System". En: *IEEE transactions on robotics and automation*, 10, 1994, 160-167.
- Desrochers, A.A., Al-Jaar, R. *Applications of Petri Nets in Manufacturing Systems: Modelling, Control and Performance Analysis*. Piscataway: IEEE Press, 1995.
- Fang, H.-L., Ross, P., Corne, D. "A promising Genetic Algorithm Approach to job-shop Scheduling, Rescheduling, and Open Shop Scheduling Problems". En: Forrest, S. (ed). *Fifth International Conference on Genetic Algorithms*, San Mateo: Morgan Kaufmann, 1993, 375-382.
- Garey, M.R., Johnson, D.S. *Computer and Intractability: A Guide to the theory of NP completeness*. San Francisco: Freeman, 1979.
- Gen, M., Cheng, R. *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons, 1997.
- Gonçalves, J., Mendes, J., Resende, M. *A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem. Report td-5eal6j*. AT&T Labs, 2002.
- Grabot, B., Geneste, L. "Dispatching Rules in Scheduling: a Fuzzy Approach". En: *International Journal of Production Research*, 32, 1994, 903-915.
- Jain, A.S., Meeran, S. "Job Shop Scheduling using Neural Networks". En: *International Journal of Production Research*, 36, 1998, 1249-1272.
- Jain, P. "Solving Resource Contention Problem in FMS using Petri nets and rule-based Approach". En: *International Journal of Production Research*, 39, 2001, 785-808.
- Kim, S., Lee, Y. "Enhancement of a Job Sequencing Rule using an artificial Neural Network". En: *Proceedings of the Second Industrial Engineering Research Conference*. Norcross: IIE, 1993, 842-846.
- Kimemia, J., Gershwin, S.B. "An Algorithm for the Computer Control of a Flexible Manufacturing System". En: *IIE Transactions*, 15, 1983, 353-362.

- Lee, D., DiCesare, F. "Scheduling Flexible Manufacturing Systems using Petri Nets and Heuristic Search". En: *IEEE Transaction on Robotics and Automation*, 10, 1994, 123-132.
- Li, D.C., Wu, C., Torng, K.Y. "Using an unsupervised Neural Network and decision Tree as knowledge Acquisition Tools for FMS scheduling". En: *International Journal of Systems Science*, 28, 1997, 977-985.
- Liu, J., McCarthy, B.L. "General Heuristic Procedures and Solutions Strategies for FMS Scheduling". En: *International Journal of Production Research*, 37, 1999, 3305-3333.
- Lloyd, S., Yu, H., Konstas, N. "FMS Scheduling using Petri Net Modeling and a Branch and Bound Search". En: *Proceedings of the IEEE International Symposium on Assembly and Task Planning*. Washington: IEEE, 1995, 141-146.
- Logendran, R., Sonthinen, A. "A Tabu search-based Approach for Scheduling Job Shop type Flexible Manufacturing Systems". En: *Journal of Operation Research*, 48, 1997, 264-277.
- Min, H.S., Yih, Y., Kim, C.O. "A competitive Neural Network Approach to multiobjective FMS Scheduling". En: *International Journal of Production Research*, 36, 1998, 1749-1765.
- Murata, T. "Petri Nets: Properties, Analysis and Applications". En: *Proceedings of the IEEE*, 77 (4), 1989, 541-580.
- Pinedo, M., Chao, X. *Operations Scheduling with Applications in Manufacturing and Services*. New York: Mc Graw-Hill, 1999.
- Ponnambalam, S.G., Aravindan, P. "Comparative Evaluation of Genetic Algorithms for Job-Shop Scheduling". En: *Production Planning and Control*, 12, 2001, 560-574.
- Rossi, A., Gino, D. "Dynamic Scheduling of FMS using a real-time Genetic Algorithm". En: *International Journal of Production Research*, 38, 2000, 1-20.
- Sabuncuoglu, I. "Scheduling with Neural Networks: a Review of the Literature and new Research Directions". En: *Production Planning and Control*, 9, 1998, 2-12.
- Saitou, K., Malpathak, S., Qvam, H. "Robust Design of Flexible Manufacturing Systems using colored Petri Net and Genetic Algorithm". En: *Journal of Intelligent Manufacturing*, 13, 2002, 339-351.
- Sawik, T. "Modelling and Scheduling of Flexible Manufacturing Systems". En: *European Journal of Operational Research*, 45, 1990, 177-190.

Stecke, K.E., Solberg, J.J. "Loading and control Problem for a Flexible Manufacturing System".

En: *International Journal of Production Research*, 19, 1981, 481-490.

Tiwari, M.K., Chandrasekaran, M., Mohanty, R.P. "Use of Timed Petri Net and Activity Cycle Diagram Methodologies for Modelling Tandem AGVs in FMSs and their Performance Evaluation". En: *International Journal of Computer Integrated Manufacturing*, 14, 2001, 399-408.

Zhou, M.C., DiCesare, F. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston: Kluwer Academic Publishers, 1993.