Programación de un Sistema Job Shop-Open Shop por medio de una Red Neuronal

Omar D. Castrillón, Santiago Ruiz-Herrera y William Sarache

Universidad Nacional de Colombia, Sede Manizales, Facultad de Ingeniería y Arquitectura,
Departamento de Ingeniería Industrial, Campus la Nubia, CP 170001, Manizales - Colombia.
(e-mail: odcastrillong@unal.edu.co; sruizhe@unal.edu.co; wasarachec@unal.edu.co)

Recibido Ene. 8, 2016; Aceptado Mar. 15, 2016; Versión final Abr. 1, 2016, Publicado Oct. 2016

Resumen

Se propone una metodología basada en redes neuronales para la programación de pedidos en un ambiente *job shop-open shop*. El problema de programación de la producción en sistemas *job shop* consiste en definir la mejor secuencia de pedidos para minimizar el tiempo total de proceso (*makespan*) u otras variables relevantes. Diversas técnicas inteligentes se han aplicado para resolver este tipo de problema. Sin embargo, éstas presentan dificultades prácticas cuando se requieren cambios de programación, pues exigen reconfigurar nuevamente el problema para obtener una nueva solución. La metodología propuesta entrena la red mediante la combinación de los tiempos de proceso en cada nodo, por medio de funciones de transferencia multiplicadas por pesos que son generados por el algoritmo de programación de la red. La red neuronal propuesta permite obtener soluciones no solo para un problema en particular, sino para los nuevos que surjan sin necesidad de hacer una nueva reconfiguración. Las soluciones encontradas fueron comparadas con otras técnicas de programación de la producción, encontrándose desempeños superiores en el *makespan* del orden del 30% al 164%.

Palabras clave: programación de la producción; redes neuronales; *job shop- open shop*; *makespan*

Scheduling of a Job Shop-Open Shop System through a Neuronal Network

Abstract

A methodology based on neural networks is proposed for solving the *job shop-open shop* scheduling problem. The job shop scheduling problem consist of defining the best order sequence to minimize the total processing time (*makespan*) or other relevant variables. Several intelligent techniques have been applied to solve this kind of problem. However, when reprogramming is required these techniques present practical difficulties since they require to be restructured to find a new solution. The proposed methodology trains the network by combining the processing times at each node, through transfer functions that are multiplied by weights obtained from the algorithm of network programming. The proposed neural network obtains solutions not only for a particular problem but also for new situations without requiring a problem reconfiguration. When compared with other techniques, the obtained solution showed a superior performance ranging from 30% to 164% in terms of the *makespan*.

Keywords: production scheduling; neuronal networks; *job shop-open shop*; *makespan*

INTRODUCCIÓN

La programación de la producción en ambientes de manufactura *job shop*, es uno de los problemas más complejos de resolver debido al alto número de posibilidades que pueden ser consideradas en la secuenciación de pedidos. La secuenciación implica determinar el orden de entrada de los pedidos al sistema de manufactura con miras a lograr los mejores desempeños en cierto tipo de variables, entre las cuales, el tiempo de procesamiento (*makespan*) ha sido uno de los más abordados. El problema de programación en ambientes *job shop* se considera del tipo *NP-hard* debido a que la cantidad de posibles soluciones son del orden de $n!^m$ (Castrillón et al., 2009; Castrillón et al., 2011), donde n corresponde al número de pedidos y m al número de máquinas que conforman el sistema.

En el estado del arte es posible identificar diversas contribuciones soportadas en el uso de técnicas inteligentes orientadas a encontrar soluciones sub-óptimas en tiempos computacionales aceptables. Dentro de estas técnicas se destacan los algoritmos evolutivos, la búsqueda tabú, los algoritmos de enjambre y el recocido simulado, entre otras. Si bien estas técnicas han demostrado buenos resultados, desde la perspectiva de la gestión de los sistemas productivos una de sus falencias más notorias radica en que cada vez que se requiera programar la producción, es necesario configurar nuevamente todos los pedidos y máquinas con miras a realizar una nueva búsqueda hasta encontrar una solución adecuada. Esta situación ofrece dificultades prácticas, dada la necesidad de las exigencias del mercado que exigen reprogramar el sistema productivo de forma permanente.

En contraste, el empleo de redes neuronales en la secuenciación de pedidos, permite construir y programar soluciones no solo para un problema en particular, sino para los nuevos que surjan sin necesidad de hacer una nueva reconfiguración del problema; es decir, las redes neuronales ofrecen soluciones de programación mucho más acordes con la dinámica que caracteriza a los sistemas de producción tipo *job shop*. Sin embargo, una revisión de 139 artículos científicos, realizada por Ruiz (2015), permite establecer que estas han sido poco exploradas en la solución de este tipo de problemas (Ver Figura 1), Aunque las mismas han sido empleadas en otras áreas (Henríquez et al., 2011; Mendes et al., 2010).

Algunas contribuciones identificadas bajo el enfoque de redes neuronales para la solución de problemas de programación de la producción en ambientes *job shop*, se resumen en la Tabla 1. La mayoría de las soluciones identificadas se enfocan en limitar el número de posibilidades, minimizar el tiempo de entrega, manejar un número fijo de máquinas, de pedidos o clasificar los problemas para ser resuelto por medio de una heurística determinada. Como se puede observar, ninguna de ellas ofrece una solución orientada al diseño y entrenamiento de una red neuronal en un sistema de producción Job Shop – Open Shop.

Relación Método Aplicado - Número de Publicaciones

Fig. 1: Técnicas inteligentes aplicadas en la solución de problemas de programación en ambientes job-shop

Tabla 1: Algunas contribuciones bajo el enfoque de redes neuronales para tratar el problema de programación en ambientes Job shop

Autor	Contribución principal
(Lo y Bavarian, 1993)	Partiendo del problema del agente viajero, se diseña una red neuronal para programación de múltiples trabajos.
(Watanabe et al., 1993)	Se estiman los tiempos de entrega en un problema Job Shop por medio de una red neuronal. Como resultado se mejora la tardanza promedio, pero no la máxima tardanza.
(Foo et al., 1995)	Se trabajan las propiedades de una red neuronal (<i>hopfield</i>) para solucionar un problema Job Shop. El trabajo se centra en el número de parámetros (amplificadores, interconexiones, etc.) que pueden ser manejados por una red neural construida con circuitos electrónicos, según el número de máquinas y trabajos. Se considera la precedencia lineal de las operaciones, lo que reduce las posibles soluciones, siendo menor que $n!^m$. Donde n son los pedidos y m las máquinas.
(Sabuncuoglu et al., 1996)	Se soluciona la tardanza promedio sobre una sola máquina. También se minimiza el <i>makespan</i> en un problema Job shop, con buenos resultados. En el problema Job Shop se considera que cada trabajo de un pedido, requiere una e máquina específica por un periodo de tiempo determinado.
(Luh et al., 1999)	Se estudia el diseño, arquitectura y uso de una red neuronal física (construida con circuitos electrónicos) para solucionar un problema Job Shop, encontrándose muy buenos resultados para estos problemas.
(Park et al., 2000)	Por medio de una red neuronal y una regla de prioridad, en este trabajo se minimiza la tardanza ponderada, en una secuenciación de un trabajo Job Shop con máquinas paralelas iguales. Se manejan secuencias, tiempos de configuración, fechas de entrega, tiempos de proceso y penalidad por retraso.
(El-Bouri et al., 2000)	En este trabajo se emplea dos redes neuronales para secuenciar trabajos sobre una máquina. La primera de ellas clasifica el problema. Acorde a esta clasificación, la segunda realiza la secuenciación de los trabajos.
(Fonseca et al., 2002)	Se construye una red neuronal para estimar el menor tiempo de entrega de una serie de pedidos sobre cuatro máquinas. Se realizó una rápida evaluación de alternativas sin necesidad de realizar simulaciones, con resultados similares a los generados por algunos programas de simulación conocidos.
(Kechadi et al., 2013)	Se diseñó un algoritmo para minimizar el mínimo tiempo de ciclo, incorporando restricciones de precedencia y partiendo de una programación inicial factible. Se logró aproximar los modelos teóricos a los prácticos, permitiendo obviar algunas limitaciones como el tamaño de los problemas con buenos resultados.
(Zeidi et al., 2013)	Se presenta un modelo de programación multiobjetivo en el cual se trata el plan de conversión de talleres de trabajo a sistemas de manufactura celular. Se emplea un algoritmo genético y una red neuronal, sobre ambientes dinámicos.
(Sun et al., 2015)	Se propone un algoritmo evolutivo híbrido con una red neuronal, para minimizar el tiempo de proceso en un problema <i>job shop</i> Flexible. Como resultado se produce un algoritmo más robusto y estable. Que incrementa el espacio de búsqueda y evita caer en óptimos locales.

El objetivo de este artículo es exponer una estructura metodológica para entrenar una red neuronal orientada a la programación de pedidos en un sistema de producción Job shop – Open shop. En la metodología propuesta se manejan restricciones tales como la precedencia de los pedidos, el número de máquinas y el número de procesos, aspecto que incrementa considerablemente el número de posibilidades de secuenciación. Con base en el entrenamiento de la red, se construyen soluciones para nuevos problemas, tendientes a minimizar el tiempo total de proceso, sin necesidad de que esta sea entrenada (programada) nuevamente. Las soluciones generadas fueron comparadas con otras técnicas de programación de la producción tales como FIFO (primera en entrar, primera en salir), LIFO (última en entrar, primera en salir), LPT (mayor tiempo de procesamiento) y SPT (menor tiempo de procesamiento). La solución encontrada mediante la red neuronal propuesta, ofreció desempeños superiores en relación con el tiempo de procesamiento de los pedidos (*makespan*).

METODOLOGÍA

La estructura de la metodología propuesta consta de 5 pasos: (1) identificación del problema de programación; (2) construcción del algoritmo; (3) validación; (4) comparación; y (5) análisis de estabilidad

Paso 1: Identificación del problema de programación

El problema de programación se basa en la identificación de la configuración productiva a programar en términos de las máquinas disponibles y del número de pedidos. Este se representa mediante una matriz de tamaño $n \times m$. En esta matriz, las filas representan los pedidos a programar y las columnas representan las máquinas. Por su parte, cada posición T_{ij} representa el tiempo que el pedido i , requiere en la máquina j (ver Tabla 2).

Tabla 2: Representación del problema de programación Job shop – Open shop N*M.

Pedido/Máquina	Maq 1	Maq 2		Maq n
P1	$T_{(1,1)}$	$T_{(1,2)}$		$T_{(1,n)}$
P2				
:		$T_{(i,j)}$		
Pn				

Paso 2: Construcción del algoritmo

El algoritmo propuesto se divide en dos fases: fase 1, programación de la red neuronal y fase 2, construcción del programa detallado (diagrama de gantt). En la fase 1, la red neuronal permite combinar varias entradas, multiplicadas por un peso que debe ser calculado. La combinación de entradas se hace por medio de una suma a la cual se le aplica una función de transferencia para generar la salida deseada (ver figura 2). Cada una de las salidas de la función de transferencia constituirá la programación de cada máquina, siempre y cuando esta sea válida, para lo cual se emplea otra función de transferencia. En particular, la fase 1 se lleva a cabo a través del siguiente procedimiento:

a) Cada una de las columnas de la Tabla 2, representará los valores $T_{i,1} \dots T_{i,m}$ que ingresarán a cada uno de los nodos M_i descritos en la Figura 2. Posteriormente, se debe definir una función de pesos $W_{i,1} \dots W_{i,m}$. Estos pesos serán los valores a ajustar una vez se programe la red.

b) para cada uno de los nodos $M_1 \dots M_n$ de la red, se define la función de transferencia que se expone en la Ecuación (1), donde, n es el número de pedidos, $T_{i,j}$ es el tiempo de proceso del pedido i en la máquina j , $W_{i,j}$ es el peso calculado para el pedido i en la máquina j . El redondeo es una función de aproximación al entero más cercano y el residuo es el valor restante de la división entera. Por su parte i es una variable que toma valores entre 1 y n y j toma valores entre 1 y el número máximo de máquinas (m). Con el ajuste de los pesos de forma adecuada, la función de transferencia debe generar un número entre 1 y el número máximo de pedidos a ser atendidos por una máquina

$$Fi(x) = \text{Residuo}(\text{Redondeo}(\sum_{i=1}^n W_{i,j} * T_{i,j}), n + 1) \quad (1)$$

c) La reunión de todos los valores de las funciones de transferencia $F_1(x)$ hasta $F_n(x)$, sobre el nodo denominado "Máquina M" en la Figura 2, determinará la programación de dicha máquina, siempre y cuando la salida sea válida. Es decir, una salida es válida si se cumple que los valores entregados por las funciones $F_1(x)$ hasta $F_n(x)$, representen una secuencia de pedidos (1...n) a ser procesados por la máquina M. Lo anterior es validado por medio de la Ecuación (2), donde n es el número máximo de pedidos.

$$\text{Factorial}(n) + \frac{n * (n + 1)}{2} = \left(\sum_{i=1}^n Fi(x) + \prod_{i=1}^n Fi(x) \right) \quad (2)$$

El anterior proceso genera una secuencia de fabricación de los pedidos en cada una de las máquinas que se están programando.

d) dado que pueden existir varias combinaciones de pesos W , los cuales generarán varias salidas validas en el sistema, es necesario realizar el respectivo diagrama de Gantt, con el fin establecer la secuencia de pesos que arroje el mejor desempeño; es decir, aquella secuencia de programación valida que produzca el menor tiempo total de procesamiento (*makespan*). Terminada la fase 1, el diagrama de Gantt que sustenta la fase 2, comprende los siguientes pasos. I) Se parte de un gen inicial, el cual representa la secuencia a ser programa (ver Tabla 3). II) Dado que pueden existir varias alternativas de programación de las máquinas que influyen en el resultado final del proceso, todas estas posibilidades deben ser consideradas al momento de construir el respectivo diagrama.

Tabla 3: Secuencia para generar el diagrama de Gantt.

Máquina 1				Máquina 2				Máquina m			
2	1		n	3	1	...	N	1	2	...	n

Paso 3. Validación

Una vez programada la red, se procede a tomar un problema de programación, el cual representará las entradas $T_{i,1} \dots T_{i,m}$ de la red propuesta y se procederá a validar la respectiva salida generada por la red. No obstante, si la salida es inválida los pesos son ajustados nuevamente.

Paso 4. Comparación

Finalmente los resultados obtenidos en el paso 3 son comparados con otras técnicas de programación de la producción tales como FIFO (primera en entrar, primera en salir), LIFO (última en entrar, primera en salir), LPT (mayor tiempo de procesamiento) y SPT (menor tiempo de procesamiento). Lo anterior con el fin de validar su aplicabilidad y su desempeño.

Paso 5. Análisis de estabilidad

Con el fin de analizar la estabilidad del sistema en los resultados entregados, un análisis de varianza usando la ecuación (3) debe ser realizado con el fin de establecer si existen diferencias significativas en los resultados. En este análisis, y_i representa la respuesta a las variables; T_i es el efecto causado por el tratamiento i -ésimo y ε_i el i -ésimo error experimental. La información recolectada debe cumplir los requerimientos de independencia y normalidad requeridos. El análisis de varianza es realizado bajo un intervalo de confianza del 99.5%.

$$y_i = \mu + T_i + \varepsilon_i \quad (3)$$

Fig. 2: Modelo de la una red neuronal para un sistema Job Shop – Open Shop. Fuente: elaboración propia.

RESULTADOS

Para la experimentación de la anterior metodología se proponen dos problemas de programación; el primero de ellos para el entrenamiento de la red, definido en la Tabla 4 y el segundo para la validación de la red definido en la Tabla 5.

Pasos 1- 2

Como resultado de los dos primeros pasos, se obtuvo la programación del problema propuesto en la Tabla 4 y la programación de la red. La mejor secuenciación encontrada en las máquinas (*Processors*) en la solución de este problema, es ilustrada en la Figura 3.

Tabla 4: Datos para Entrenamiento.

Pedidos	Máq 1	Máq 2	Máq 3	Máq 4	Máq 5
P1	7	2	6	8	7
P2	6	5	5	2	6
P3	5	8	4	10	5
P4	4	7	3	8	3

Tabla 5: Datos para validación

Pedidos	Máq 1	Máq 2	Máq 3	Máq 4	Máq 5
P1	21	6	18	24	21
P2	18	15	15	6	18
P3	15	24	12	30	15
P4	12	21	9	24	9

Fig. 3: Mejor secuenciación al programar la red. Makespan = 40.

Paso 3

Una vez entrenada la red se procedió a generar la solución, para el respectivo problema mostrado en la Tabla 5. Como resultado de este paso se obtuvo la secuencia de programación que se expone en la Fig. 4.

Fig. 4: Solución encontrada por medio de la red neuronal. Makespan = 99

Tomando, como referencia la forma como han sido solucionado algunos problemas similares en la literatura (Castrillón, 2009), el problema fue solucionado usando 10 réplicas y 4 corridas. Los resultados obtenidos se exponen en la Tabla 6.

Tabla. 6: Solución del problema para 10 réplicas y 4 corridas

	<i>Réplicas</i>										
<i>Corrida</i>	1	2	3	4	5	6	7	8	9	10	<i>Suma</i>
Primera	123	126	111	126	120	114	105	126	114	123	1188
Segunda	117	123	120	132	99	105	114	120	126	120	1176
Tercera	120	111	114	105	117	114	120	123	117	111	1152
Cuarta	114	120	126	114	105	120	99	123	117	111	1149
Promedio General											117

Paso 4

Al solucionar el problema planteado en la Tabla 5 por medio de algunas de las técnicas tradicionales de programación de la producción, se obtienen los resultados ilustrados en la Tabla 7. La tercera columna calcula el porcentaje de ineficiencia de estas técnicas respecto al promedio general obtenido por la solución propuesta (ver Tabla 6).

Tabla 7: Desempeño usando técnicas tradicionales

Técnica	Makespan	Ineficiencia
Pilas (FIFO)	153	30%
Colas (LIFO)	153	30%
Mayor tiempo de proceso (LPT)	291	148%
Menor tiempo de proceso (SPT)	309	164%

Paso 5

Tomando como referencia las soluciones encontradas en el paso 3 (Tabla 6), se realizó el análisis de varianza, bajo un intervalo de confianza del 99.5%. Dado que en la Tabla 8, el F tabulado (Columna F_{Tab}) es mayor que el F calculado (Columna F_{Cal}) se puede expresar que no existen diferencias significativas en los resultados entregados por el sistema; es decir, bajo un intervalo de confianza del 99.5% los datos son estadísticamente iguales.

Tabla 8: Análisis de Varianza (Anova)

<i>Fuente Variación</i>	<i>GL</i>	<i>SC</i>	<i>CM</i>	<i>F_{cal}</i>
Sum C Total		2307,38		
Tratamiento	3,00	106,88	35,63	0,58
Error Exper	36,00	2200,50	61,13	
Total	39,00	2307,38	96,75	

DISCUSIÓN

Con la metodología propuesta se obtuvo la mejor secuencia de fabricación de los 5 pedidos analizados, (Figura 4) para un *makespan* o tiempo de procesamiento total de 99 minutos. Sin embargo, los datos empleados para calcular el análisis de varianza (Tabla 6) muestran que el promedio de los resultados obtenidos es un poco más alto (117). No obstante lo anterior y los diferentes resultados encontrados, en las cuatro corridas con 10 réplicas cada una, el análisis de varianza muestra que no existe diferencia significativa en estos resultados. Se demuestra que, bajo un intervalo de confianza del 99.5%, los resultados son estadísticamente iguales. Por lo tanto el promedio general obtenido (117) es un buen indicador para comparar esta metodología frente a otras técnicas tradicionales de programación de la producción.

De otro lado, debido que este problema tiene 207.360.000 posibles soluciones ($5!^4$), es difícil encontrar el valor óptimo, más aún cuando existen tiempos muertos en las diferentes líneas de proceso. No obstante, en lo que respecta al *makespan*, los resultados encontrados muestran una mejora significativa respecto a los sistemas tradicionales empleados, lo cual es altamente relevante para los propósitos de entrega rápida que generalmente persiguen este tipo de sistemas de fabricación. Al mismo tiempo, y en relación a los sistemas inteligentes de programación de la producción, se evita el recalcule de soluciones como típicamente sucede con las dichas técnicas. En contraste, dada su naturaleza, las redes neuronales pueden ser programadas en una primera fase cuyos resultados facilitan, en una segunda fase, el abordaje de nuevos problemas sin ser reestructurada de nuevo. Finalmente se resalta que aunque estas técnicas producen muy buenos resultados, su aplicación ha sido escasa y no existen trabajos similares como se ilustró en la tabla 1 y la figura 1 de la introducción.

CONCLUSIONES

Si bien en la literatura existen diferentes técnicas para realizar la programación de la producción, la novedad de este artículo se centra en la metodología propuesta para obtener buenos resultados en el desempeño del sistema, sin necesidad de recalcular las soluciones cada vez que se deban definir fechas de entrega. Igualmente los diferentes análisis de varianza realizados muestran que el sistema es estable, aun cuando en las corridas y replicas se obtengan resultados ligeramente diferentes. No obstante, la metodología propuesta es fácil de aplicar, pues solo es necesario definir la estructura de red propuesta en la Figura 2 y seguir cada uno de los pasos propuestos en la metodología.

Como futuras líneas de investigación se puede explorar la modificación de la metodología propuesta para abordar la solución de problemas de secuenciación en sistemas más complejos como el *job shop* flexible y sistemas híbridos. Igualmente es viable realizar experimentos para comparar la metodología propuesta con otras técnicas de inteligencia artificial. Aunque resultados preliminares en esta materia muestran que los resultados pueden ser similares, es necesario un estudio más profundo para corroborar esta tendencia.

REFERENCIAS

- Castrillon, O., W. Ariel y J. Giraldo, Metodología Job Shop basada en una colonia de hormigas, *Dyna*, 159, 177-184 (2009)
- Castrillón, O., W. Sarache y J. Giraldo, Aplicación de un Algoritmo Evolutivo en la Solución de Problemas Job Shop-Open Shop, *Inf. Tecnol*, 22(1), 83-92 (2011)
- El-Bouri, A., S. Balakrishnan y N. Popplewell, Sequencing jobs on a single machine: A neural network approach, *European Journal of Operational Research*, 126, 474 - 490 (2000)
- Fonseca, D., y D. Navarrese, Artificial neural networks for job shop simulation, *Advanced Engineering Informatics*, 16, 241–246 (2002).
- Foo, S., Y. Takefuji y H. Szu, Scaling properties of neural networks for job-shop scheduling, *Neurocomputing*, 8, 79-91 (1995)
- Henríquez, M., P. Palma y A. Patricio, Control Automático de Condiciones Ambientales en Domótica usando Redes Neuronales Artificiales, *Inf. tecnol.*, 22(3), 125-139 (2011)
- Kechadi, T., K. Low y G. Goncalves, Paper Recurrent neural network approach for cyclic job shop scheduling problem, *Journal of Manufacturing Systems*, 32, 689-699 (2013)
- Lo, Z., y B. Bavarian, Multiple Job scheduling with artificial neural networks, *computer and electrical engineering*, 19(2), 87-101 (1993)
- Luh, P., X. Zhao y otros 4 autores, Architectural Design of Neural Network Hardware for Job Shop Scheduling, *Annals of the CIRP*, 48(1), 373-376 (1999)
- Mendes, V., F. Leta y otros 2 autores, Detección de Posición Angular de Embarcaciones, utilizando Técnicas de Visión Computacional y Redes Neurales Artificiales, *Inf. tecnol.*, 21(6), 177-188 (2010)
- Park, Y., S. Kim y Y. Lee, Scheduling jobs on parallel machines applying neural network and heuristic rules, *Computers & Industrial Engineering*, 38, 189-202 (2000)
- Ruiz, S., Metodología multiobjetivo basada en un comportamiento evolutivo para programar sistemas de producción Job Shop, aplicaciones en la industria metalmecánica, p-38. Tesis Doctoral en Ingeniería, Universidad Nacional de Colombia (2015)
- Sabuncuoglu, I., y G. Burckan, A neural network model for scheduling problems, *European Journal of Operational Research*, 93(1), 288-299 (1996)
- Sun, L., L. Lin y otros 3 autores, A Bayesian Optimization-based Evolutionary Algorithm for Flexible Job Shop Scheduling, *Procedia Computer Science*, 61, 521-526 (2015)
- Watanabe, T., H. Tokumaru y Y. Hashimoto, Job shop scheduling using neuronal networks, *control engineering practice*, 1(6), 957-961 (1993)
- Zeidi, J., N. Javadian y otros 2 Autores, A hybrid multi-objective approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system, *Computers & Industrial Engineering*, 66, 1004-1014 (2013)