

Un enfoque de optimización y simulación utilizando algoritmo genético para el problema de secuenciamiento dinámico con demanda estocástica

A simulation-optimization approach using genetic algorithm for dynamic Job Shop scheduling with stochastic demand

Marilda de Fátima de Souza da Silva¹ Fernanda Morán Menezes Pereira¹
Patricia Chambal Rodriguez² Fabio Henrique Pereira^{1, 2, 3}

Recibido 19 de septiembre de 2017, aceptado 2 de junio de 2018
Received: September 19, 2017 Accepted: June 2, 2018

RESUMEN

El problema de programación de la producción en ambientes job shop ha sido ampliamente estudiado en las últimas décadas debido a su importancia y complejidad computacional. Sin embargo, una gran parte de los trabajos abordan la versión estática y determinista del problema. En este trabajo se presenta la integración de un modelo de simulación con un método de optimización para resolver el problema de programación dinámica con demanda estocástica. La integración del modelo propuesto se logra utilizando componentes fuera de proceso, a través de la tecnología *ActiveX Automation* y *Visual Basic for Application*, en la que un Algoritmo Genético simple se ejecuta como una aplicación autónoma. Los resultados del método propuesto se compararon con algunas normas comunes de expedición y muestran que el método de optimización y simulación pueden resolver el problema de programación eficientemente. Además las soluciones generadas por el AG son menos sensibles a las variaciones de la demanda, lo que es muy significativo en estos entornos.

Palabras clave: Simulación, programación, secuenciamiento dinámico, Job Shop, demanda estocástica.

ABSTRACT

The Job Shop scheduling problem has been extensively studied in the last decades due to its importance and computational complexity. However, a large part of the papers address the static and deterministic version of the problem. This work presents an integration of a simulation model with an optimization method in order to solve the dynamic job shop scheduling problem. The proposed model integration is accomplished using out-of-process components, through the ActiveX Automation technology and the Visual Basic for Application, in which a simple Genetic Algorithm runs as a free-standing application. Results of the proposed method were compared with some common dispatching rules and showed that the proposed approach the scheduling problem efficiently. Moreover, the solutions generated by GA are less sensitive to variations in demand, which is quite significant in such environments.

Keywords: Modelling, simulation, dynamic scheduling, Job Shop, stochastic demand.

¹ Programa de Pós-graduação em Informática e Gestão do Conhecimento da Universidade Nove de Julho. Rua Vergueiro 235. São Paulo, Brasil. E-mail: marilda.silva@uni9.pro.br; fernandamorán@uninove.edu.br

² Diretoria dos Cursos de Informática da Universidade Nove de Julho. Rua Vergueiro 235. São Paulo, Brasil. E-mail: patriciachambal@yahoo.com.br

³ Programa de Pós-graduação em Engenharia de Produção da Universidade Nove de Julho. Rua Vergueiro 235. São Paulo, Brasil. E-mail: fabiohp@uni9.pro.br

* Autor de correspondencia

INTRODUCCIÓN

Uno de los principales problemas que enfrentan las empresas de manufactura es el secuenciamiento de la producción, también llamado de scheduling, que consiste en identificar cuál o cuáles son las mejores maneras de ordenar el programa de producción en las máquinas para una mejora en la eficiencia [1].

En entornos de producción del tipo Job Shop, los recursos son representados como máquinas y un job es definido como un conjunto de una o más actividades, también llamadas de tareas, y cada tarea es única, con rutas preestablecidas y diferentes, procesada al menos una vez en cada una de las máquinas. Los problemas dentro de ese entorno son conocidos como Problemas de Secuenciamiento de la Producción en Job Shop (del inglés, JSSP).

Debido a su importancia y complejidad computacional, el JSSP ha sido extensivamente estudiado. Sin embargo, la gran mayoría de los artículos aborda el problema determinístico y/o estático, y no tienen en cuenta los tiempos de procesamiento estocástico y/o la llegada dinámica de tareas que reflejan de manera más adecuada a las situaciones de la industria del mundo real. Además, muchos de los artículos tratan el problema de programación en diferentes entornos de producción [2]. De esta forma, es posible ver una gran brecha entre la práctica y la teoría de secuenciamiento en Job Shop [3].

Como, en general, cada abordaje de integración específica utiliza su propia representación del sistema, se hace difícil el intercambio de datos y exige el desenvolvimiento de interfaces dedicadas [4].

De este modo, este trabajo tiene como objetivo encontrar, por medio de la integración entre el modelo de simulación con el método de optimización, una o más soluciones satisfactorias para el problema de secuenciamiento dinámico y estocástico de la producción en un entorno Job Shop. Por lo tanto, la busca por la mejoría continua en el proceso de producción, específicamente en el secuenciamiento de la producción (scheduling) hace con que la contribución de este trabajo, con una propuesta de abordaje de uso simple, pueda ser utilizado en la planificación de la producción, pues puede ofrecer respuestas más eficientes en relación al medio real de cambio constante y al

mismo tiempo puede ayudar estratégicamente los tomadores de decisiones en el proceso diario de una empresa de manufactura.

PROBLEMA DE SECUENCIAMIENTO DINÁMICO Y ESTOCÁSTICO DE LA PRODUCCIÓN EN JOB SHOP

En el problema de secuenciamiento estático de la producción, todas las tareas están listas para empezar a momento cero, mientras que en la secuenciación de tareas dinámicas pueden llegar a algunos momentos futuros conocidos o desconocidos. Además, si los tiempos de procesamiento son aleatorios, entonces, el problema es también clasificado como estocástico. De este modo, según [5], el problema de secuenciamiento dinámico y estocástico puede ser definido de acuerdo con las dos características siguientes: (1) El tiempo entre la llegada de los pedidos es considerado como una variable aleatoria, lo que significa que la llegada de las tareas al sistema es dinámica; (2) Los tiempos de procesamiento de las tareas varían estocásticamente en cada máquina.

ENFOQUE TEMÁTICO

Módulo de simulación

La simulación es la imitación de la operación de un sistema del mundo real a lo largo del tiempo. En este caso, la representación del sistema es definida solamente en puntos discretos en el tiempo en que ocurren las alteraciones de estado del sistema [6]. Uno de los principales objetivos del módulo de simulación consiste en evaluar la solución para un módulo de optimización [7].

En este trabajo, el módulo de simulación fue desarrollado con base en simulación de eventos discretos, por medio del software ARENA 14.5 y tiene como objetivos:

- (1) evaluar la solución candidata para el problema de programación generada por el módulo de optimización;
- (2) construir un programa teniendo en cuenta todas las limitaciones;
- (3) evaluar el desempeño de las reglas de secuenciación comunes para condiciones dinámicas y estocásticas del piso de fábrica (para la comparación).

El modelo de simulación fue construido a partir de una escenografía dinámica y estocástica de la producción compuesta de 8 máquinas y 10 tipos de jobs con rutas predefinidas, y estimados tiempos de producción determinados de acuerdo con la Tabla 1.

Tabla 1. Escenografía dinámica y estocástica de la producción: jobs con rutas predefinidas y tiempos de producción estimados.

	Ruta	Tiempo (min)		Ruta	Tiempo (min)
1	1, 2, 3, 6	16,9	6	1, 2, 4, 3, 1, 2, 4, 5, 7	36,0
2	1, 3, 7, 8	16,4	7	2, 6, 5, 6, 8	20,7
3	1, 5, 8	11,6	8	1, 2, 4, 5, 7	19,7
4	1, 2, 3, 5, 6, 7, 8	27,3	9	2, 3, 4, 6, 7	21,0
5	3, 4, 6, 7	16,3	#	1, 2, 4, 5, 8	20,6

Otros parámetros del modelo de simulación propuesto se presentan en la Tabla 2. Los detalles del desenvolvimiento y evaluación se presentan en [7], realizado de acuerdo con [8].

El formato de soluciones de secuenciamiento

El módulo de simulación desmenuado reconoce una solución del problema de programación por medio de una secuencia de prioridades de las tareas en cada una de las máquinas. Entonces, una solución es representada en el interior del módulo como una lista de número entero k , donde k es el número de operaciones. Es importante destacar que él es independiente de la representación adoptada por el módulo de optimización, en que puede ser utilizado en cualquier enfoque disponible. Además, como diferentes tareas deben tener diferentes valores de prioridad, en una determinada máquina los

valores son quitados de la lista teniendo en cuenta la ocurrencia en vez de la posición.

Dentro del modelo, cada tarea tiene un plano de proceso definido como una secuencia de acuerdo con la ruta en la Tabla 1. Cada paso del plano de proceso para todas las tareas tiene una prioridad de atribución de nombre, cuyo valor es almacenado como un objeto del modelo en un módulo específico, y ella es cambiada durante el proceso de solución.

Modelo lógico

Las soluciones criadas por el módulo de optimización son insertadas en el modelo utilizando el Visual Basic for Application (VBA) en el ARENA. El proyecto VBA tiene acceso al modelo de simulación por medio del objeto This-Documents que contiene muchos eventos VBA integrados en el ARENA donde el código es activado. Estos eventos integrados dividen en tres categorías que definen el exacto momento durante la simulación de que el código VBA es ejecutado [9]: (1) eventos Pre-ejecutable (por ejemplo, DocumentOpen); (2) eventos de ejecución Arena-iniciado (por ejemplo, RunBegin, RunEndReplication); (3) eventos de ejecución modelo / usuario iniciado (por ejemplo, UserFunction, VBA_Block_Fire).

Una ilustración de la secuencia de estos eventos VBA, con instrucciones para en el establecimiento de una secuencia de prioridades en el modelo, se presenta en la Figura 1. En la Figura 1, la cursiva indica los eventos VBA integrados, al tiempo que subraya representan las instrucciones programadas en el código VBA desarrollado en el modelo. Los valores de prioridad se introducen en el campo apropiado antes de que comience la simulación y los resultados se guardan después de cada replicación del modelo.

Tabla 2. Parámetros del modelo de simulación propuesto (std = standard deviation).

Parámetros	Expresión	Descripción
Tiempo entre llegadas	EXPO (mean)	Descripción exponencial: 8, 10, 12, 14, 16 y 18 minutos media.
Tiempo de procesamiento	Normal (4, 0,4)	A distribución normal de media de 4 min. y 0,4 min. std
Factor de confianza	$k = 0,3$	Factor de confianza en relación a data de vencimiento
Tiempo de producción estimado	τ_0	Estimado por <i>processing time x number of operation</i> de tareas
Fecha de vencimiento	NORMAL ((1+k) * τ_0 , 0,1* (1+k)* τ_0)	Distribución normal de (1+k) * τ_0 media y 0,1* (1+k) * τ_0 std

1. *RunBegin* (ARENA comprueba el modelo y se inicia la simulación)
2. *RunBeginSimulation*
 - Define un tipo de variable 'ARENA model'
 - Define un tipo de variable 'ARENA module'
 - Define un archivo de entrada (*Input*)
 - Guardan en un vector de las prioridades de los valores creados por el módulo de optimización
 - Asigna el modelo actual en la variable 'ARENA model'
 - Encontrar el módulo en el que se deben almacenar los valores de prioridad
 - Asigna el módulo encuentra la variable 'ARENA module'
 - Asigna los valores de las prioridades de todos los *Jobs*
3. *RunBeginReplication*
4. *RunEndReplication*
 - Define un objeto SIMAN en el que los resultados están disponibles
 - Manejar los resultados de objeto SIMAN (*makespan*, tiempo de retardo y el número de tareas atrasadas)
 - Guarda los resultados en un archivo de salida
5. *RunEndSimulation*
6. *RunEnd*

Figura 1. Ilustración de la secuencia de eventos VBA.

Módulo de optimización

El Algoritmo Genético (AG) es una técnica meta-heurística - método estocástico, no determinístico de busca y optimización. Él imita el proceso evolutivo que ocurre con organismos biológicos en la naturaleza, basado en el proceso de selección natural. Sin embargo, el AG presenta aspectos peculiares en relación a otros métodos de optimización. Mitchell hace destaque a tales aspectos [10]: (i) trabajar con los parámetros de codificación, en vez de variables originales del problema, (ii) pesquisar las soluciones optimas a partir de un conjunto de soluciones potenciales y no apenas de un punto de partida, (iii) usar una función de evaluación para las diferentes soluciones y (iv) el uso de reglas probabilísticas en encontrar nuevas soluciones.

A partir de esta teoría, el módulo de optimización es desarrollado con base en la biblioteca Galib, componentes del Algoritmo Genético en C++ [11].

Tabla 3. Parámetros utilizados en las experiencias.

Parámetros	Valor Adoptado
Representación cromosoma	<i>Ordered operation-based</i>
Selección	<i>SteadyState</i>
Tasa de reemplazo	80%
Tamaño de la población	50
Cruce	<i>Partial Match</i> (90%)
Mutación	<i>Swap</i> (5%)
Numero total de Generaciones	30
Criterio de parada	
Función de evaluación	Módulo de simulación

Sin embargo, como la propuesta de integración es realizado utilizando componentes fuera del proceso, en que el método de optimización es ejecutado como un aplicativo independiente, el módulo de simulación puede ser fácilmente integrado con otras aplicaciones desenvueltas en cualquier lenguaje de programación como C++, Visual Basic o Java.

La Tabla 3 resume los parámetros del AG que fueron adoptados para ejecutar la propuesta de integración.

Entorno de integración

La integración fue realizada utilizando las instalaciones del control dentro de las aplicaciones, a través de la tecnología ActiveX Automation y del Visual Basic for Application. ActiveX es una especie de Component Object Model (COM), que permite una comunicación entre aplicaciones de forma independiente del lenguaje de programación, esto es, permite que otros métodos sean fácilmente integrados más tarde por el usuario, a través de otro código de propósito general para hacer frente con eventos externos en una simulación en tiempo real. El control del módulo de simulación es realizado por la aplicación C++. Además de realizar las operaciones normales del método de optimización, tales como los operadores genéticos, la aplicación hace control a ejecución del modelo siempre que una nueva solución candidata tiene que ser evaluada.

RESULTADOS NUMÉRICOS

Los experimentos que utilizan un AG simples fueran realizados principalmente para ilustrar la operación de la integración propuesta. Además, seis niveles de demanda diferentes fueran testados a fin de evaluar sus efectos sobre las reglas de secuencia de acuerdo con el *makespan*, tiempo total de retardo total y número de trabajos atrasados.

Resultados para el tiempo de retardo se muestran en la Figura 2. Los resultados del método propuesto fueron comparados con algunas reglas de secuenciación comunes: *Shortest Processing Time* (SPT), *Longest Processing Time* (LPT), *Earliest Due Date* (EDD), *First In First Out* (FIFO) e *Last In First Out* (LIFO), para diferentes niveles de demanda. Cuanto más corto sea el tiempo entre llegadas más largo son las demandas.

Se ve que, los valores de las medidas de desempeño empeoran mientras la demanda aumenta. Por otra

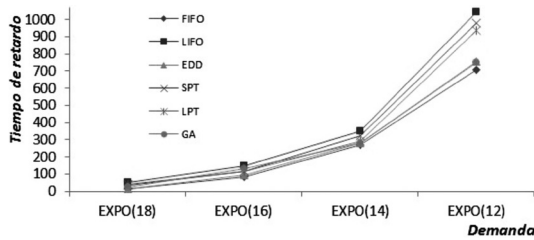


Figura 2. Resultados para el tiempo de retardo.

parte, los efectos de las reglas de secuenciación son más significativos cuando el nivel de la demanda es mayor. En general, a pesar del pequeño número de individuos y el número de generaciones adoptadas, la simulación basada en el AG podría alcanzar buenos resultados. Es importante señalar que la integración propuesta fue aplicada como un enfoque multiobjetivo, lo que significa que el objetivo es minimizar las medidas de desempeño simultáneamente, mismo que los resultados de AG son, en la mayoría, similares a los das reglas de expedición.

Además de promover una reducción significativa en el número de retardos, en particular para las demandas más altas, como se muestra en las Figuras 2 y 3, las soluciones generadas por el AG son menos sensibles a las variaciones de la demanda. Se observa que los números obtenidos en las soluciones generadas por el AG a las demandas EXPO(6) y EXPO(8) son casi iguales y alrededor de 77 unidades, en comparación con más de 115 unidades en el mejor caso para las reglas de expedición (LPT).

En este caso, los datos de reglas de secuenciación tienen una variación mínima de 73% (regla EDD) cuando cambia la demanda EXPO(8)-EXPO(6) mientras que la variación AG es inferior a 10% (aumento de 77,1 a 84,7). Esto muestra que la integración puede producir igualmente buenos resultados en situaciones de demandas diferentes. Estos resultados se muestran en la Figura 3. Las soluciones generadas por el AG son menos sensibles a las variaciones de la demanda.

Además de la optimización, la integración con simulación permite evaluar otras medidas de rendimiento del sistema e inferir/ver correlaciones importantes entre estas mediciones. Se observa en la Figura 4, por ejemplo, el tamaño máximo de la

cola y la tasa de utilización en cada máquina, lo que permite identificar los puestos de trabajo que limitan la producción (cuello de botella, ejemplo la máquina 2) y ver el aumento esperado en las colas y las tasas de utilización debido al aumento de la demanda.

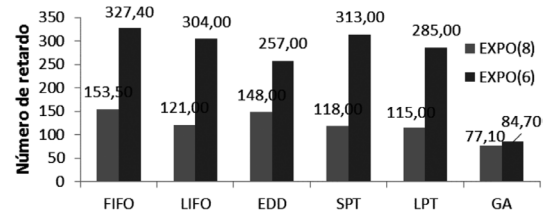


Figura 3. Números de retardos para los diferentes enfoques.

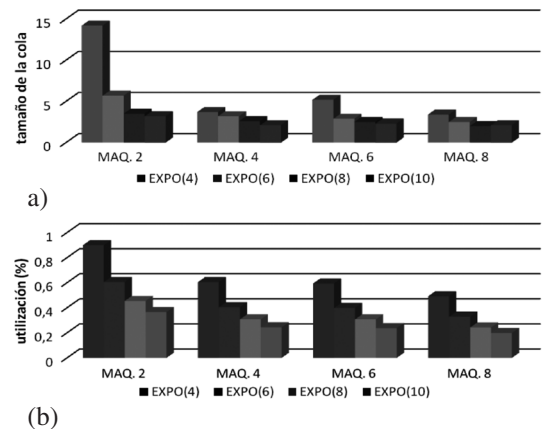


Figura 4. Tamaño máximo de la cola (a) y Tasa media de utilización por máquina (b).

CONCLUSIONES

En este trabajo proponemos una integración entre el modelo de simulación y el método de optimización a fin de solucionar el problema de secuenciamiento dinámico e estocástico de la producción en un entorno *Job Shop*.

A pesar de usar un método de AG sencillo, los resultados numéricos muestran que el método de optimización y simulación pueden resolver el problema de programación eficientemente, logrando resultados hasta 40% mejores en comparación con las reglas de secuenciación comunes. Además las soluciones generadas por el AG son menos sensibles

a las variaciones de la demanda, lo que es muy significativo en entornos *job Shop* con demanda estocástica variable.

El único inconveniente observado fue con el tiempo de procesamiento, principalmente debido al procedimiento de cambio de mensajes. Sin embargo, esto se puede reducir mediante el uso de una solución *Message broker* que es un módulo de programa intermediario que traduce un mensaje del protocolo de mensajería formal del remitente para el protocolo de mensajería formal del receptor, lo que permite intercambiar mensajes entre aplicaciones.

AGRADECIMIENTOS

Los autores desean agradecer a la Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP, Grant 2014/08688-4), al Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Grant 155934/2015-5) y el Programa de Pós-graduação em Informática e Gestão do Conhecimento en la Universidade Nove de Julho - PPGI/UNINOVE por su apoyo financiero.

REFERENCIAS

- [1] P.P. Lukaszewicz. "Metaheuristics for job shop scheduling problem, comparison of effective methods". Tesis para optar al grado de doctor. Aarhus School of Business/Aarhus. 2005.
- [2] R.M. Durán, L.P. Rojas and V.P. Daza. "A genetic algorithm for the Flexible Job Shop problem". *Ingeniare. Revista chilena de ingeniería*. Vol. 19 N° 1, pp. 53-61. 2011. ISSN: 0718-3305. DOI: 10.4067/S0718-330-5201-1000100006.
- [3] D. Ouelhadj and S. Petrovic. "A Survey of Dynamic Scheduling in Manufacturing Systems". *Journal of Scheduling*. Vol. 12, pp. 417-431. Octubre 2009. ISSN: 1094-6136. DOI: 10.1007/s10951-008-0090-8.
- [4] S. Iassinovski, A. Artiba and V. Bachelet, "Integration of simulation and optimization for solving complex decision making problems". *International Journal of Production Economics*. Vol. 85 N° 1, pp. 3-10. Julio 2003. ISSN: 0925-5273. DOI: 10.1016/S0925-5273(03)00082-3.
- [5] B. Zhang, L-X. Yi and S. Xiao. "Study of stochastic job shop dynamic scheduling". *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*. pp. 18-21. Guangzhou, China. 2005.
- [6] J. Banks. "Introduction to Simulation". *Proceedings of the Winter Simulation Conference*. Orlando, FL, USA. 2000.
- [7] E.B. Silva, M.G. Costa and M.F.S. Silva. "Simulation study of dispatching rules in stochastic job shop dynamic scheduling". *World Journal of Modelling and Simulation*. Vol. 10 N° 3, pp. 231-240. Mayo 2014. ISSN: 1746-7233.
- [8] F. Leal, R.F.S. Costa and J.A.B. Montevechi. "A practical guide for operational validation of discrete simulation models". *Pesquisa Operacional*. Vol. 31 N° 1, pp. 57-77. Abril 2011. ISSN: 0101-7438. DOI: 10.1590/S0101-74382011000100005.
- [9] W.D. Kelton, R.P. Sadowski and D.A. Sadowski. "Simulation With ARENA". McGraw Hill. 2ª ed. Boston, USA. pp. 385-396. 2000. ISBN: 978-0071122399.
- [10] T.M., Mitchell. "Machine Learning". McGraw-Hill. 1ª ed. New York, USA. pp. 249-273. 1997. ISSN: 978-0070428072.
- [11] M. Wall. GALIB: A C++ library of genetic algorithm components. Mechanical Engineering Department, Massachusetts Institute of Technology. 1996. <http://lancet.mit.edu/ga/dist/>