

14. Comparación de técnicas de secuenciación de producción para aplicación en industria salvadoreña

D. E. Zamora y F. J. Gómez, Departamento de Operaciones y Sistemas-UCA

Resumen

La investigación consiste en la comparación de diversas técnicas de secuenciación de producción y sus requisitos tecnológicos para determinar qué tipo de algoritmo pueda ser útil ante diversas condiciones de configuraciones de taller. El objetivo de indagar en el tema es identificar las generalidades de los algoritmos que se puedan utilizar y reconocer la mejor forma de seleccionarlos, dependiendo de características locales y condiciones de recursos que tenga una empresa para su aplicación.

En el estudio, se utiliza una comparación de resultados obtenidos de cada herramienta en un Taller Gráfico. La comparación de los diversos modelos recopila diversas investigaciones sobre algoritmos existentes para diversas configuraciones y si éstos han obtenido resultados con diferencias significativas. Su comparación incluye una serie de requisitos tecnológicos, a modo que la elección pueda ser hecha a través de varios criterios de factibilidad de implementación. Por la naturaleza de los algoritmos, se ha previsto la comparación del tiempo de programación de producción como criterio a la prueba independiente de cada método.

Índice de términos

Algoritmos de secuencias, Job Shop, Makespan, Secuencia de producción.

I. INTRODUCCIÓN

En la comparación de sistemas de producción aplicados en las industrias salvadoreñas, se puede advertir una forma muy marcada en base a mantener una alta variabilidad de productos con pocos volúmenes de cada uno. Esta sección, considera que existe un buen número de aplicaciones de una producción de tipo Job Shop o conocida como configuración de taller.

En el problema sobre la resolución del orden de cómo hacer la programación de diversas tareas en diferentes máquinas, reconociendo que cada tarea puede tener una secuencia diferente y se puede atender

diferentes cantidades vuelve una situación que no se suele modelar matemáticamente. En un trabajo día a día, cada lugar de trabajo se encuentra con el dilema de reordenar su producción para cumplirle al cliente. La situación exige en obtener un cálculo complejo que deba realizarse en un corto tiempo.

En la configuración de taller, se considera que no exista una solución a través de una función polinómica cerrada y con un alto grado de dificultad del modelado estocástico, denominado como NP-difícil [1].

El segundo problema encontrado es qué método conviene seleccionar si sabe que cada planteamiento no logrará siempre la mejor respuesta pero



se desconocen las mejores condiciones productivas para ello. Existen diversos enfoques para proponer una solución de programación, pero cada técnica puede ser muy buena solución para una característica de tareas pero no ser adecuada cuando existen flujos muy distintos. En las soluciones, se han desarrollado respuestas en base a algoritmos de colonias de hormigas, algoritmos genéticos, métodos de mejora y diversas técnicas [2].

A la interrogante de diferentes algoritmos, se decidió comparar resultados entre cuatro métodos de secuenciación usados en la configuración de taller y decidir sobre las condiciones de aplicación para aplicarlos según la realidad que exista entre cada empresa. La comparación trata de arrojar una breve idea sobre cuándo sería más adecuado utilizar un algoritmo determinado.

Entre los cuatro métodos a comparar están: algoritmo de colonia de hormigas, algoritmo de Johnson de múltiples máquinas, algoritmo de Jackson y General SB Heuristic de secuenciación Job Shop. La comparación se realiza con los datos obtenidos en el Taller Gráfico UCA, por su comportamiento de procesos de taller marcado ante la división de cada algoritmo.

II. MATERIALES Y MÉTODOS

Los algoritmos procesados se basan en obtener una respuesta aceptable para el tiempo que existe una solución en un tiempo determinado. Los métodos trabajan con funciones objetivos similares pero con un enfoque completamente distinto para relacionar las variables. Sin embargo, los arreglos muestran caracterizan detalles similares que permiten una comparación.

A. Algoritmo de colonia de hormigas (ACO)

Su base es debido a la comparación matemática donde existe una cantidad infinita de soluciones y a través de un parámetro de feromona, la hormiga (variable de decisión) toma un camino más corto [3]. La programación de producción está basada en representar los caminos

como posibles soluciones de secuencia y tomar el makespan como el largo del camino. Se añaden restricciones adicionales por los caminos previos que debe tomar una hormiga y sus parámetros de revisión para establecer la intensidad de la feromona [4]. Su forma versátil logra describir el algoritmo en base a la programación.

B. Algoritmo de Johnson

Está basado en reglas empíricas sobre ordenar las tareas de menor duración en las iniciales y en las de mayor duración en las últimas secuencias [5]. Su complicación es en obtener diferentes máquinas mientras se respeta la secuencia de las tareas que debe llevar el proceso.

Su diferencia radica en el uso donde sus reglas de decisión están basadas localmente en la duración de las tareas y no tanto en la formación del tiempo total utilizado para la elaboración de todo el programa de producción [6].

C. Algoritmo de Jackson

Su base se realiza según la secuencia de producción donde se define el orden de los trabajos que deberán entrar al centro de procesamiento. El objetivo es una asignación óptima realizada según secuencia y se reconoce como una extensión del algoritmo de Johnson [7].

Su ventaja es la forma de abordaje en la entrega, aunque se asume que su resultado fue óptimo hasta 3 máquinas. Con una cantidad mayor no se obtenía una certeza de sus resultados por las interacciones que realizan las tareas precedentes.

D. General SB Routine

Se propone la reducción de la tardanza mínima de todos los trabajos involucrados por medio de crear una secuencia directa con el recurso declarado cuello de botella. Ese cuello de botella se considera dinámico en el algoritmo y permite su estabilización en las diferentes pruebas [8].

El cuello de botella dinámico permite disminuir el tiempo de carga del resto de recursos en una planta de producción. Su regla de detención es basada en el incremento o aumento posible de makespan, al desarrollar una nueva solución.

E. Metodología de comparación entre los métodos

Los cuatro métodos fueron probados con diferentes escenarios para las órdenes del Taller Gráfico UCA. Cada método logró seleccionar un número distinto de órdenes en pedidos de una o dos semanas, a fin de obtener los datos necesarios de la comparación.

Entre los diferentes trabajos de Taller Gráfico, se logró un acuerdo con utilizar entre 12 y 16 órdenes de producción que reflejaran una semana de probabilidad de funcionamiento del Taller. En ciertos casos, se les dio un seguimiento con la segunda semana, dada la condición del taller y sus restricciones de capacidad. Para los trabajos, se disponía en obtener sus mediciones hasta completar la entrega, ya fuese avanzar una tercera semana de seguimiento para conseguir la secuencia y tiempos del trabajo. Las fechas de los trabajos fue aleatoria entre métodos y se desarrollaron en momentos diferentes para no repetir estudios entre los algoritmos.

Los datos fueron tomados directamente de los reportes de órdenes de producción con los tiempos estimados que el Taller Gráfico maneja para su proceso. En algunas tareas, se logró una verificación por medio de un estudio de tiempos, para verificar puntos de control que permitieran una mejor confiabilidad de resultados para cada método.

En tres de los algoritmos, se decidió por agrupar las máquinas en cinco grandes departamentos: diseño, fotomecánica, impresión, acabado y producto terminado. La agrupación se asumió con la idea de mantener ciertas tareas previamente agrupadas por el tipo de trabajadores que se mantienen dentro del Taller Gráfico. En el método General SB Routine, se modeló máquina por máquina por la diferencia que tiene cada forma de aprender sobre el negocio. Ver la TABLA I

Tabla I. Diseño de comparación de los métodos

Método a comparar	Órdenes de producción	Fechas	Estaciones (departamentos)
ACO	19	Inicio de noviembre 2012	5
Johnson	15	Final de noviembre 2012	5
Jackson	15	Final de noviembre 2012	5
SB Heuristic	18	Inicio de julio 2013	11

Los métodos se compararon respecto al criterio de la minimización del Makespan como función objetivo principal de comparación. Los otros valores objetivos fueron exclusivamente para mantener un parámetro de referencia, como el tipo de software que fue necesario utilizar, el esfuerzo realizado para obtener una solución y otros parámetros que sólo fueron posibles medirlos en métodos específicos

F. Cálculo del makespan

El makespan se considera como el tiempo de procesamiento necesario desde la primera tarea del primer trabajo iniciado en la secuencia hasta terminar con la última tarea en el último recurso. Como definición es el tiempo de procesamiento necesario para realizar las n tareas que se disponga a programar.

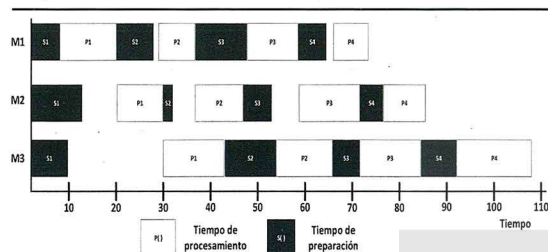
Por varios autores, se conoce que se estructura como la sumatoria de los tiempos de inicio de cada tarea, más el tiempo de preparación que exista entre las estaciones, más el tiempo de procesamiento de las tareas en la última secuencia y el tiempo de finalización en el último recurso [9].

Para ilustrar el concepto, la fig.1 tiene una muestra de 4 tareas con 3 estaciones [10]. El tiempo del makespan se consideraría desde el S1 en



el tiempo 0 hasta la finalización de la tarea P4 al tiempo 108. En caso de tomar el makespan sólo para las dos tareas, llegaría hasta el tiempo 85 con el P4 del recurso M2.

Fig. 1. Forma de cálculo del makespan.



Un supuesto importante a recalcar en el estudio es que no se consideran los inicios ni las fechas donde el cliente inicia el proceso de petición de producción. Este inicio, ha permitido escalar la programación de una forma más suavizada, aunque se ha probado el criterio de utilizar el algoritmo cuando la planta reciba muchos trabajos que no se logre definir el orden de ingreso a producción.

Un segundo supuesto es que los recursos podrían trabajar las 8 horas laborales de un día y que ningún recurso estaría falto de personal para operar. No se considera que trabajen horas extras aunque no se consideró como una restricción dentro del modelo, sólo para obtener la secuencia de la producción.

III. RESULTADOS OBTENIDOS PARA CADA MÉTODO

Según la configuración de cada algoritmo utilizado, se logra obtener una propuesta que permite la reducción del makespan ideal en base al enfoque de cada método.

A. Propuesta en Algoritmo de Colonia de Hormigas

Dado los 19 trabajos obtenidos para el Algoritmo, se decidió en utilizar una restricción adicional para modelar la secuencia. Se encontró que la secuencia por tareas podía ser representada sólo si las estaciones de trabajo asumidas podrían tener un recurso y no ser vistas como estaciones en paralelo. Esta idea básica, restringe el uso del método cuando se quiera transformar hacia 11 estaciones o más para medir su secuencia.

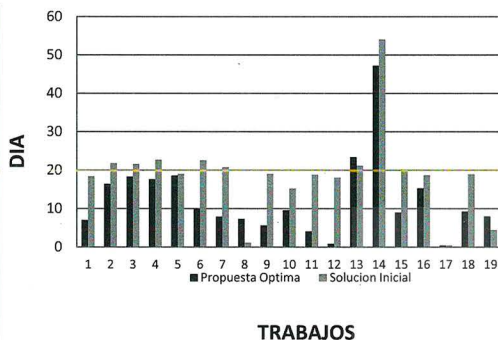
En el resultado, se logró comparar frente a la regla primero en entrar y primero en salir (FIFO). De 19 operaciones, 17 lograrían reducir su tiempo de makespan compensando el atraso sólo en dos trabajos. Ver Fig. 2.

Fig. 2. Comparación de los tiempos de duración de las tareas para Algoritmo de Colonia de Hormigas.

Nombre	Duración actual	Duración anterior	Duración dif.
* J1	6.99 días	18.41 días	-11.43d
* J2	16.42 días	21.77 días	-5.35d
* J3	18.31 días	21.61 días	-3.3d
* J4	17.67 días	22.74 días	-5.07d
* J5	18.57 días	19.04 días	-0.46d
* J6	9.99 días	22.8 días	-12.71d
* J7	7.92 días	20.75 días	-12.83d
* J8	7.31 días	1.17 días	6.14d
* J9	5.61 días	19.07 días	-13.46d
* J10	9.58 días	15.22 días	-5.63d
* J11	4.13 días	18.83 días	-14.7d
* J12	0.92 días	18.14 días	-17.22d
* J13	28.41 días	21.2 días	7.21d
* J14	47.22 días	54.06 días	-6.84d
* J15	8.99 días	20.24 días	-11.25d
* J16	15.3 días	18.59 días	-3.29d
* J17	0.46 días	0.46 días	0d
* J18	9.24 días	18.98 días	-9.75d
* J19	8 días	4.51 días	3.5d

En la Fig. 3, se nota la mejora total realizada en la diferencia del valor del makespan obtenida en el ACO que con la regla FIFO. Cabe notar, que la solución propuesta necesitó el uso de software Matlab para calcular la nueva posición de secuencia y se auxilió de MS Project para estimar el diagrama de Gantt en precedencia y calcular la carga de los recursos. El tiempo reducido por el algoritmo fue de 121 horas en las producción.

Fig. 3. Comparación del makespan en el ACO para los trabajos mostrados.

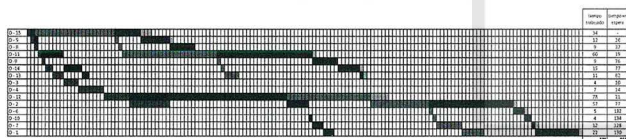


B. Propuesta en Algoritmo de Johnson

En los 15 trabajos, se notó una mayor presencia de tareas con mayor duración que mostrarían un valor más elevado en el makespan general. Se comparó el resultado del algoritmo contra la regla FIFO, por ser la que utilizan de preferencia en el Taller Gráfico.

En la fig. 4, se observa la propuesta generada por el algoritmo, tomando a consideración de ser diferentes tareas comparadas en el ACO.

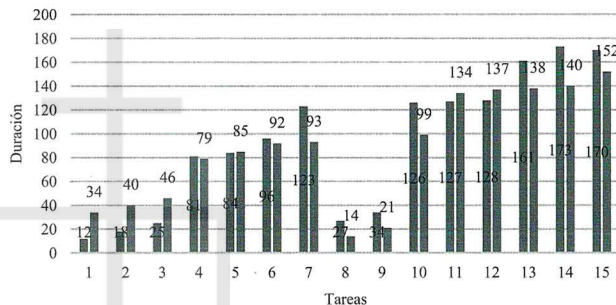
Fig. 4. Propuesta en algoritmo de Johnson.



Para el algoritmo, fue necesario obtener la codificación en Matlab®, para incluir la forma en que cada secuencia pueda ser vista dentro de cada máquina. Su tiempo de propuesta de solución es más corto dado que puede generar soluciones sin una regla de entrada.

En la fig. 5, se observa que el makespan reducido a todos los trabajos fue de 81 horas. Las últimas tareas consideran un mayor tiempo de ejecución por la espera que deben tener respecto a las primeras tareas.

Fig. 5. Comparación de makespan para algoritmo de Johnson.



En el algoritmo, también se advierte el uso de una gráfica de Gantt realizada con una base de la hoja de cálculo para entender su secuencia. No se modeló una herramienta automática para determinar las precedencias por su condición.

C. Propuesta para algoritmo de Jackson

En la propuesta generada, se logra visualizar que mantiene un rango de disminución entre 75 y 90 horas. El rango es por la explicación del algoritmo en el inicio que permite una variabilidad en la forma de comenzar en el algoritmo. La regla interna para el desempate fue la del tiempo más corto como inicio.

En esta propuesta sólo se identificó que la primera solución tiende a una forma de Johnson, pero logró considerar los hechos de las prioridades de cada trabajo dentro del Taller Gráfico. El resultado se observa mejor



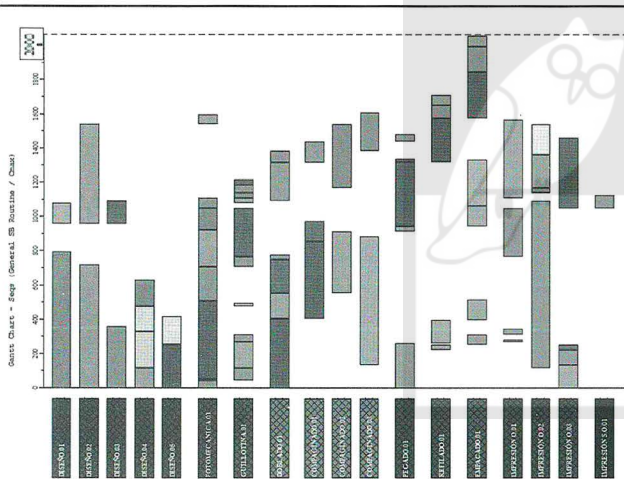
en la fig. 5 de la propuesta de Johnson, pero con la probabilidad de cambiar según las reglas del criterio interno.

D. General SB Heuristic

Su identificación en la propuesta era determinar el cuello de botella cambiante entre los recursos y máquinas utilizadas, a forma de programar el menor tiempo posible en la máquina declarada como cuello de botella.

El resultado de la fig. 6, muestra las tareas a través de colores que pasan por las diferentes máquinas, mostradas como barras independientes en el gráfico. La secuencia es almacenada en otro punto del programa y no se muestra en esta gráfica.

Fig. 6. Propuesta por SB Heuristic



Un aporte sobre el método es encontrar que fotomecánica obtiene la mayor cantidad de trabajos en cola y su línea de espera es la más grande del resto de recursos.

Sin embargo, el resultado especial sobre los productos estudiados en la metodología, mencionó proponer una forma igual a la regla FIFO. En comparaciones, sería obtener una reducción de cero horas en el makespan; aunque se aclara que el algoritmo puede comprobar que en esa condición, la propuesta FIFO iguale a la propuesta del método tratado.

La condición de uso para el SB Heuristic es el uso del software Lekin © desarrollado para comparar las reglas tradicionales de secuenciación contra el método heurístico de cuello de botellas dinámicas en las configuraciones de tipo taller.

IV. CONCLUSIONES

Según el algoritmo comparado, una solución directa considerando la información adicional sobre la programación, sería el uso del método del General SB Heuristic por declarar el recurso del cuello de botella. Se necesitaría el software Lekin © para dar una mayor aplicación del algoritmo en entornos de decisiones rápidas.

El algoritmo de colonia de hormigas posee un mejor resultado relativo de disminuir el makespan propuesto. Su desventaja en el esfuerzo es el modelado de secuencias múltiples que el algoritmo pueda no reaccionar frente a restricciones de retornos considerables. Es una técnica que deberá revisarse en el modelado de obtener secuencias factibles y lógicas, como una forma de mejora al algoritmo usado en este artículo. Su implementación, requeriría de un software programable en base a matrices y funciones como el programa Matlab ©.

En el algoritmo de Johnson, su reducción de makespan fue directa frente a la comparación del resto de algoritmos. Este permite incluir una propuesta en menor tiempo de respuesta que los demás algoritmos y puede ser considerado como una solución inicial para tomar la base de ahorrar tiempo en los otros algoritmos. Su desarrollo en varias máquinas, implicaría el uso del software Matlab © por las condiciones de secuenciar en base a la regla de menor duración a la mayor duración.

El algoritmo de Jackson puede tender a la propuesta en Johnson, salvo que permite establecer un rango de respuesta por la forma en que se puedan considerar las prioridades de una tarea. En ese sentido, permitiría comparar los resultados de los otros modelos para reconocer si sería útil incluir la prioridad en la planta de producción para la toma de decisiones y su efecto en los makespan del resto de productos.

En la secuenciación, en los cuatro métodos se encontró que debe existir un desarrollo de comunicación de otros programas para los módulos que controlen sus algoritmos. El objetivo de esa plataforma de traducción estaría dada por el beneficio de controlar las órdenes de producción con el sistema actual y generar una propuesta de secuencia que permita analizar los efectos de la decisión final del jefe de producción. Sin la conexión, resultará muy complicado tratar de implementar los algoritmos dado que poseen reglas avanzadas de programación en las secuencias.

Se recomienda experimentar las cuatro herramientas como opciones de propuesta, considerando diferentes tiempos de inicio y una cantidad más grande de tareas estudiadas para definir su eficiencia computacional.

V. RECONOCIMIENTOS

Los autores agradecen las contribuciones de:

René Guinea, José Magaña, Nelson Marcía y Walter Olivares en la realización del trabajo de graduación de Algoritmos de Colonias de Hormigas para la Programación de Planta de Producción.

Mauricio Carranza, Andrea Ponce y Emely Recinos por la realización del trabajo de graduación Propuesta de algoritmo determinístico de secuenciación para un taller de producción, donde se hizo la comparación del algoritmo de Johnson y el de Jackson.

Alfonso Rosa, Adriana Orellana, Morena Recinos y Natalia Rosa por la elaboración del trabajo de graduación Propuesta del algoritmo para programar producción en Talleres Gráficos UCA, usando Teoría de Restricciones, donde se comparó el SB Heuristic.

VI. REFERENCIAS

- [1] .unalmed.edu.co/~pruebasminas/index.php?. [Último acceso: 21 Abril 2013].
- [2] E. Vendramini, «Optimización del problema de cargamento de contenedores usando una metaheurística eficiente.» UNESP, Ilha Solteira, Brasil, 2007.
- [3] M. G. L. Dorigo, «Ant Colony System: A cooperative learning approach to Traveling Salesman Problem.» Evolutionary Computation, IEEE Transactions on, vol. 1, n° 1, pp. 53-66, 1997.
- [4] D. Das, Ant colony optimisation: application to scheduling problem, India: National Institute of Yourkela, 2009.
- [5] A. Ramos, «Universidad Pontificia de Comillas.» 20 Enero 2008. [En línea]. Available: www.iit.upcomillas.es/aramos/simio/transpa/t_ts_ar.pdf. [Último acceso: 4 octubre 2012].
- [6] L. Krajewski, L. Ritzman y M. Malhotra, Administración de operaciones, 8va ed., México D.F.: Editorial PEARSON, 2008.
- [7] M. Vélez, C. Castro y J. Toro, «Algoritmo de búsqueda aleatoria para la programación de la producción en un taller de fabricación.» REVIS-TA Universidad EAFIT, vol. 39, n° 131, pp. 76-86, 2003.
- [8] M. Pinedo, Scheduling: Theory and Algorithms, Editorial Springe, 2008, p. 671.
- [9] A. D. d. Barros. y V. M. João , «Simulated Annealing de Busca do Gargalo Fluctuante.» 2002.
- [10] M. Agreda, J. Álvarez, R. Estrada y J. López, «Aplicación del algoritmo BGaFSA en el sistema productivo de "Cafeterías UCA",» de Trabajo de graduación para Ingeniería Industrial, San Salvador, UCA, 2013, p. 35.



VII. BIOGRAFÍAS

Fernando Gómez nació en San Salvador, el 31 de agosto de 1986. Se graduó de Ingeniero Industrial en la Universidad Centroamericana "José Simeón Cañas" de El Salvador.

Su experiencia laboral ha sido en impresión comercial, en La Prensa Gráfica como Ingeniero de Métodos y Supervisor de Almacén. Su área de especialidad es el uso de técnicas para optimización de producción.

Se desempeña como docente investigador de la Universidad Centroamericana, en la especialidad de Ingeniería Industrial.

Gómez recibió un reconocimiento de la Asociación Salvadoreña de Ingenieros Mecánicos, Eléctricos e Industriales (ASIMEI) en 2009 por desempeño académico en la universidad.

Dennis Zamora nació en Santa Ana, el 12 de diciembre de 1986. Se graduó de Ingeniero Industrial en la Universidad Centroamericana "José Simeón Cañas" de El Salvador.

Su experiencia laboral ha sido en el área bancaria, en la capacitación del personal en las finanzas, en el Banco Procredit y en la extrusión, soplado e inyectado de plásticos como jefe de producción en Artisa. Su área de especialidad es el uso de herramientas para la toma de decisiones en las operaciones.

Se desempeña como docente investigador de la Universidad Centroamericana, en la especialidad de Ingeniería Industrial.