Comparando métodos heurísticos para secuenciar tareas en líneas de flujo

Mireya L. Valenzuela Luna
Departamento de Ingeniería de Sistemas, Instituto Tecnológico de Tepic
mireyalisset@yahoo.com.mx
Roger Z. Ríos Mercado
División de Posgrado en Ingeniería de Sistemas, FIME, UANL
roger@uanl.mx

RESUMEN

En este artículo se presenta un estudio computacional de una serie de métodos para encontrar secuencias de n tareas en un ambiente de líneas de ensamblado o flujo (Flow Shop) de m máquinas con el objetivo de minimizar el tiempo en el cual todas las tareas terminan de ser procesadas en el sistema. Este es un problema derivado del área de secuenciamiento en sistemas de manufactura, de los clasificados como difíciles de resolver. La evidencia computacional muestra que el Método Modificado de Palmer encuentra las mejores secuencias para el caso general de m máquinas (m>3).

PALABRAS CLAVE:

Investigación de operaciones, sistemas de manufactura, problema de secuenciamiento, línea de flujo, minimización de tiempo de terminación, heurística

ABSTRACT

In this article a computational study of several methods for solving the m-machine flow shop with makespan minimization objective is presented. Such problem arises from manufacturing systems, and it is regarded as difficult to solve. The empirical evidence shows that the Modified Palmer Sequence method finds solutions of better quality for the general m-machine case (m>3).

KEYWORDS:

Operations research, manufacturing systems, scheduling problem, flow shop, makespan minimization, heuristics

INTRODUCCIÓN

La ciencia de la toma de decisiones, mejor conocida como Investigación de Operaciones (IO),[2] nació hace más de 50 años por motivos de carácter militar, en la segunda guerra mundial, cuando George Dantzig inventó el método simplex para resolver problemas de optimización lineal. La IO se utiliza en todos los niveles y en todo tipo de industrias. En éstas se busca solucionar problemas donde se tome la mejor decisión sujeta a las restricciones tecnológicas existentes que reditúe en beneficios económicos mediante el uso de modelos matemáticos, que sirven para obtener una representación abstracta del problema a resolver, y computadoras, que ayudan a desarrollar e implementar las técnicas de solución.

Uno de los problemas que se presentan de forma natural en algunas empresas es el de líneas de flujo en sistemas de manufactura. El problema consiste en cómo secuenciar las operaciones en las máquinas de la forma más eficiente posible. Si no se hace inteligentemente, esto puede representar un costo de oportunidad (pérdida económica) para la empresa. Por lo tanto, es importante encontrar el orden en el cual se deben de procesar todas las tareas en todas las máquinas con el fin de reducir al mínimo el tiempo de terminación. Una excelente referencia sobre problemas de secuenciamiento es el texto de Pinedo.[3]

El objetivo de este estudio es evaluar computacionalmente algunas de las técnicas de solución que se usan para resolver este problema de secuenciamiento en líneas de flujo e ilustrar que hay técnicas de solución más efectivas que otras, las cuales deben tomar en cuenta la estructura matemática del problema.

En primera instancia se presenta una descripción detallada del problema de líneas de flujo y posteriormente se presentan algunas técnicas populares de solución como es el caso de la regla de Johnson, la cual encuentra la solución exacta en problemas con 2 máquinas, y dos heurísticas para problemas de *m* máquinas. Finalmente, se lleva a cabo una evaluación computacional de las técnicas estudiadas aplicándolas en la solución de varias instancias del problema.

Al evaluar empíricamente el comportamiento de cada heurística mediante un análisis comparativo, se observó que una de las heurísticas (MPS) resultó más efectiva que la otra (CDS) en relación a la calidad de las soluciones encontradas.

DESCRIPCIÓN DEL PROBLEMA

El problema de línea de flujo considerado consiste en un conjunto de *n* tareas que deben ser procesadas a través de un conjunto de *m* máquinas cada una. Cada tarea tiene el mismo ruteo a través de las máquinas. Es decir, cada tarea debe procesarse primero en la máquina 1, luego en la 2 y así sucesivamente hasta las *m* máquinas. El tiempo de procesamiento de la tarea j en la máquina i se denota por P_{ij}.

Una máquina no puede procesar más de una tarea a la vez. El objetivo es encontrar una secuencia de tareas que minimice el tiempo en el que termina de ser procesada la última tarea en el sistema, o, equivalentemente, en la última máquina. A este tiempo de terminación se le conoce como *makespan*. Es un problema de optimización combinatoria ya que el número total de secuencias posibles es *n!*, las cuales resultan imposibles de enumerar para valores de *n* relativamente grande.

La figura 1 muestra una secuencia de tres tareas S=(1,2,3) que se procesan en cada una de las tres máquinas. Por ejemplo, la tarea 1, ocupa 5 unidades de tiempo en la máquina 1, posteriormente pasa a la máquina 2 donde ocupa 4 unidades de tiempo, para finalmente pasar a la máquina 3 donde usa 2 unidades de tiempo terminando al tiempo 11. En este ejemplo, el *makespan* para la secuencia mostrada es 21.

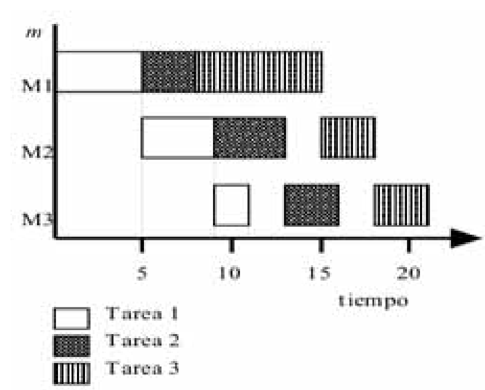

Fig. 1. Ejemplo de una secuencia de 3 tareas en una línea de flujo de 3 máquinas.

Este problema, como muchos otros en el campo de secuenciamiento de sistemas de manufactura es sumamente difícil de resolver ya que está clasificado técnicamente como NP-completo. Esto significa que cualquier algoritmo de solución emplea un tiempo de ejecución que aumenta, en el peor de los casos, exponencialmente con el tamaño del problema. Esto no significa que el problema no se pueda resolver, sino que uno debe de proponer algoritmos de solución que exploten favorablemente la estructura matemática del problema para que sean capaces de resolver la mayoría de las instancias del problema en tiempos de ejecución relativamente pequeños. La otra alternativa es recurrir a heurísticas, las cuales son procedimientos que encuentran soluciones aproximadas en tiempos de ejecución relativamente razonables, lo cual constituye el enfoque de este trabajo. Véase[4,5] para una descripción más detallada del problema.

MÉTODOS DE SOLUCIÓN

Caso de 2 Máquinas

El caso especial de este problema donde se consideran dos máquinas ha sido resuelto por Johnson.[3] La regla de Johnson consiste en un algoritmo exacto, que da la secuencia óptima para el problema de 2 máquinas con *n* tareas. El algoritmo consiste primeramente en la división de los trabajos en dos conjuntos. El conjunto 1 contiene todas las tareas donde $P_{1j}<P_{2j}$ y el conjunto 2 tiene todas las tareas con $P_{1j}>P_{2j}$. Las tareas donde $P_{1j}=P_{2j}$ pueden ir en cualquiera de los dos conjuntos. Posteriormente, las tareas en el conjunto 1 se ordenan de forma creciente de P_{1j} (regla SPT, *shortest processing time*), y las tareas del conjunto 2 van en orden decreciente de P_{2j} (regla LPT, *largest processing time*). La unión de estas dos subsecuencias (las del conjunto 1 más las del conjunto 2) proporciona la secuencia óptima al problema.

Caso de *m* Máquinas

Como es bien conocido en el área, cuando el problema es mayor o igual a 3 máquinas, no existe ninguna regla simple que proporcione una solución óptima al problema. En este trabajo estudiamos un par de métodos heurísticos que, de alguna forma, intentan transformar el problema a uno de 2 máquinas para usar posteriormente la regla de Johnson y encontrar soluciones de buena calidad. En esencia, una heurística es un procedimiento que no garantiza obtener la solución óptima global del problema, sin embargo, si se hace de forma inteligente, es capaz de encontrar buenas soluciones en tiempos razonablemente pequeños.

Método CDS

El método CDS (propuesto por Campbell, Dudek y Smith)[1] consiste de dos etapas. Primero, se transforma el problema original en uno de dos máquinas mediante una partición de las máquinas en dos conjuntos. En el primer conjunto se agrupan las primeras *q* máquinas *{1, 2, ..., q}* y en el otro las restantes *{q+1, ..., m}*. Luego, se calculan los tiempos de procesamiento Q_{ij} (de la tarea *j* en la máquina *i*) del problema transformado (de 2 máquinas) de acuerdo a esta partición para cada tarea *j* de la siguiente forma:

$$Q_{1j} = \sum_{i=1}^{q} P_{ij} \quad \text{y} \quad Q_{2j} = \sum_{i=q+1}^{m} P_{ij}$$

Una vez que se tiene la transformación se emplea el algoritmo de Johnson para dos máquinas obteniendo una secuencia de tareas. Nótese que en el algoritmo CDS original, se generan *m*-1 problemas de 2 máquinas (uno por cada valor de *q*, *q*=1, ..., *m*-1) y se toma como solución la mejor de las *m*-1 secuencias. En nuestra implementación nos limitamos a generar solo una, tomando *q* = *m*/2.

Método MPS

La heurística MPS (Modified Palmer Sequence) propuesto por Hundal y Rajgopal,[1] intenta también reducir el problema a una de dos máquinas para posteriormente utilizar la regla de Johnson para obtener una secuencia. Sin embargo, a diferencia del método anterior, los tiempos de procesamiento S_{ij} del problema reducido para cada *j* se calculan así:

$$S_{1j} = \sum_{i=1}^{m} (m-2i)P_{ij}$$

$$S_{2j} = \sum_{i=1}^{m} (m-2i+2)P_{ij}$$

La lógica en éste método es secuenciar las tareas de tal forma que las tareas que vayan de tiempos menores a mayores en la secuencia de operaciones se procesen más temprano. A este problema reducido se le aplica el algoritmo de Johnson para obtener la secuencia.

EVALUACIÓN COMPUTACIONAL

Los procedimientos fueron codificados en lenguaje C en una estación de trabajo Sun Ultra 10 con sistema operativo Solaris 7, usando la opción de compilación –o de Sun Forte v.6.0.

Para llevar a cabo la evaluación de cada heurística se generaron 90 problemas con 2, 5 y 10 máquinas para 50, 100 y 500 tareas. Los P_{ij} fueron generados mediante una distribución uniforme en [10,100]. Para cada posible combinación m x n se generaron 10 instancias. Cada una de las 90 instancias generadas fue intentada resolver por los tres métodos (Johnson, CDS, MPS). Además, para fines comparativos se utilizó también un método para generar secuencias totalmente aleatorio (MA).

Las tablas I y II muestran los resultados de la comparación entre MA y CDS y entre CDS y MPS, respectivamente. En cada una de las celdas se muestra el porcentaje de veces que un método encontró mejores soluciones que el otro. Por ejemplo, el 100 en la celda 5×500 de la tabla I significa que CDS fue mejor que MA en el 100% de las instancias probadas de 5 máquinas y 500 tareas .

Como puede apreciarse en la tabla I vemos que en casi todos los casos la heurística CDS mejoró en un 100% al MA lo cual era de esperarse. En la tabla II vemos que la heurística MPS dominó totalmente a la CDS en las instancias probadas. En el caso de 2 máquinas, es evidente que el Algoritmo de Johnson resuelve óptimamente el 100% de las instancias probadas. La tabla III nos muestra el intervalo de optimalidad relativo entre MA y el Método de Johnson. Por ejemplo, en el caso de 100 tareas el MA aleatorio está a un 3.81% del óptimo del problema.

Tabla I. Porcentaje (%) mejorado del método CDS en contraste con el método aleatorio.

m / n	**50**	**100**	**500**
5	100	90	100
10	100	100	100

Tabla II. Porcentaje (%) mejorado de la heurística MPS en contraste con el CDS.

m / n	**50**	**100**	**500**
5	100	100	100
10	100	100	100

Tabla III. Promedios de optimalidad relativa para problemas de 2 máquinas.

n	**MA**	**J**
50	3.67	0.00
100	3.81	0.00
500	1.40	0.00

Ahora bien, en la tabla IV observamos los promedios del valor de la solución obtenida por cada método para los casos de 50, 100 y 500 tareas. Comparando estos valores, se observó una notable disminución en los valores reportados por la heurística CDS y mayor aún en la heurística MPS.

En el caso de 5×50 las soluciones encontradas por el MPS fueron superiores en un rango de 5-32% relativo con respecto al CDS. De forma similar en

Tabla IV. Promedio del tiempo de terminación.

50 Tareas			
m	**MA**	**CDS**	**MPS**
5	3426.5	3231.9	2767.9
10	4001.6	3726.0	2685.5
100 Tareas			
m	**MA**	**CDS**	**MPS**
5	6324.2	6040.2	5373.4
10	7167.3	6710.1	5468.1
500 Tareas			
m	**MA**	**CDS**	**MPS**
5	29308.9	28488.9	27242.1
10	30435.6	29412.0	27304.6

los casos 5×100 y 5×500, MPS fue superior sobre CDS con rangos de 2-33% y 1-8%, respectivamente. Para los casos de mayor tamaño (10 máquinas), la diferencia fue aún más dramática. MPS exhibió mejoras de 27-48%, 14-37% y 5-10%, en las instancias de 50, 100 y 500 tareas, respectivamente. A su vez, puede apreciarse que ambas fueron muy superiores al compararse con el MA.

CONCLUSIONES

En este trabajo se ilustró la efectividad del Método de Johnson al resolver óptimamente instancias de 2 máquinas y de hasta 500 tareas en una fracción de segundos.

Además, los resultados computacionales mostraron contundentemente que la heurística MPS fue mejor que la CDS para el problema general de línea de flujo de *m* máquinas.

A su vez, ambos métodos (CDS y MPS) diseñados después de un estudio de la estructura del problema, resultaron mejores que uno de los que ignoran a la misma, en este caso el método aleatorio, por lo que tenemos múltiples alternativas en las que se puede elegir la que mayores beneficios reditúe y así encontrar secuencias de buena calidad.

La implicación práctica del presente trabajo es que la heurística CDS puede usarse para obtener secuencias de buena calidad que pueden recomendarse en el sistema de manufactura estudiado.

AGRADECIMIENTOS

El trabajo de Mireya Lisset Valenzuela Luna fue apoyado por una beca del Programa de Verano de la Investigación Científica del Pacífico (DELFIN). También agradecemos a Conrado Borraz y José Luis Rivera, estudiantes de posgrado y licenciatura de la FIME, por sus sugerencias en la realización de este proyecto.

REFERENCIAS

1. T. S. Hundal y J. Rajgopal. An extension of Palmer´s heuristic for the flow shop scheduling problem International Journal of Production Research, 26(6): 1119-1124,1988.
2. K. Mathur y D. Solow. Investigación de Operaciones: El arte de la toma de decisiones. Prentice-Hall, México,1996.
3. M. Pinedo. Scheduling: Theory, Algorithms, and Systems. Prentice-Hall, Englewood Cliffs, New Jersey, EUA, 1995.
4. R. Z. Ríos Mercado y J. F. Bard. Heurísticas para secuenciamiento de tareas en líneas de flujo. Ciencia UANL, 3(4):420-427, 2000.
5. R. Z. Ríos Mercado y J. F. Bard. Secuenciando óptimamente líneas de flujo en sistemas de manufactura. Ciencia UANL, 4(1):48-54, 2001.