

Asignación de trabajos en máquinas en paralelo mediante un modelo de programación entera

Assignment of work on parallel machines using integer programming model

Harol Mauricio Gámez Albán

Escuela de ingeniería industrial y estadística, Universidad del Valle, Cali, Colombia

hghamagal@gmail.com

Resumen— Encontrar una manera óptima para la asignación de trabajos en distintas máquinas y la programación de las mismas es un aspecto importante en las industrias manufactureras. En este artículo se presenta un modelo de programación entera que decide los trabajos y máquinas a programar en un horizonte planeación de una semana. Las variables definidas están asociadas con la cantidad pedida por el cliente por los diferentes productos para una fecha determinada. Una particularidad del modelo es el tiempo en que debe estar listo el pedido del cliente, el cual para este caso se maneja como turnos de trabajo a la semana, donde el turno indica en que día estará listo del pedido del cliente. Se muestran los resultados de la formulación en términos de calidad de la solución y tiempo de ejecución, esto se hace para tres instancias. Al final, la formulación entrega resultados satisfactorios en los diferentes ítems de evaluación, alcanzando tiempos de ejecución razonables para instancias medianamente grandes.

Palabras clave — Programación entera, programación de máquinas, asignación de trabajos.

Abstract— Find an optimal way for the allocation of work in different machines and programming them is an important aspect in manufacturing. This paper presents an integer programming model that decides the work and machinery to programming in a planning horizon of one week. Defined variables are associated with the amount requested by the customer for each product for a specific date. A peculiarity of the model is the time to be ready the customer's order, which for this case is handled as shifts a week, where the shift indicates that the day will be ready by request. Shows the results of the formulation in terms of solution quality and run time, this is done for three instances. In the end, the formulation delivered satisfactory results in the different assessment items, reaching reasonable execution time for medium-large instances.

Key Word — Integer programming, machines programming, assignment of works.

I. INTRODUCCIÓN

Las industrias manufactureras afrontan día a día el problema de programación de la producción, enfrentándose a restricciones como de capacidad de planta, fecha de entrega, máquinas, personal y entre otras que hacen de este un problema difícil de resolver a simple vista. Desde la perspectiva de la investigación de operaciones, este tipo de problemas se enmarcan dentro del área conocida como *production programming*. Básicamente los problemas de esta área consisten en la asignación de trabajos en diferentes máquinas respetando una serie de condiciones y requerimientos. Este problema se adapta al caso de una empresa manufacturera ubicada en la ciudad de Cali, que se encarga de producir insumos para calzado.

Esta programación debe satisfacer una serie de requerimientos impuestos por políticas de la compañía. Actualmente la programación se hace de manera manual, quedando expuesta a posibles errores como exceder la capacidad e incumplimiento de las fechas de entrega generando tiempos de ociosidad entre otros más.

Este artículo presenta un modelo de programación entera, para la asignación de trabajos y programación de máquinas, que busca maximizar la producción satisfaciendo todos los requerimientos y restricciones de la planta. Este modelo incorpora por un lado los requerimientos y condiciones deseables y por otro los objetivos perseguidos por la compañía.

Dada la dificultad práctica para resolver de forma exacta los problemas de programación entera, se han desarrollado algoritmos que proporcionan soluciones factibles (es decir que satisfacen las soluciones del problema), los cuales aunque no proporcionan soluciones óptimas, al menos se acercan a dicha solución con un esfuerzo computacional razonable.

Hoy en día existe suficiente bibliografía donde se proponen diferentes alternativas para dar solución a este tipo de problemas, tal es el caso que proponen Zhi y Warren [1], donde presentan un problema de programación entera para la programación de máquinas en paralelo utilizando el algoritmo

de generación de columnas. Luego Lee y Chen [2], proponen una formulación para la programación de trabajos en diferentes maquinas que funcionan en paralelo pero considerando que en el horizonte de planeación al menos una vez cada máquina estará en mantenimiento con el objetivo de minimizar el tiempo total en que debe estar listo cada trabajo, adicional proponen el algoritmo Branch and Bound basado en un enfoque de generación de columnas con el cual encuentran soluciones buenas para instancias medianas en tiempos de computacionales normales.

Dai [3], propone una heurística para la programación de trabajos en distintos talleres, la heurística propuesta utiliza inventarios de seguridad y mantiene la maquina cuello de botella ocupada en todo momento y las otras funcionando a ritmo de ella. El autor hace un análisis probabilístico de la heurística, donde el makespan de la heurística supera el makespan optimo.

En general existen diferentes herramientas para resolver este tipo de problemas, como lo son las técnicas de descomposición de Benders y Danzig Wolf. También es posible ver algoritmos aplicados como el de Generación de Columnas, Branch and Bound y Branch Price, todos en busca de una mejor solución para problemas de programación entera. Y cuando se quieren cotas maximales o minimales se encuentra la Relajación Lineal y Lagrangeana.

El objetivo de este paper está dirigido a encontrar una formulación para la programación de maquinas y asignación de trabajos, que de una buena solución en un tiempo computacional relativamente razonable.

Este artículo está organizado de la siguiente manera: se describe el problema de asignación de trabajos y programación de máquinas, luego se presenta la formulación inicial, posterior se evalúa la formulación para tres instancias mostrando resultados computacionales y por último se muestran algunas conclusiones del trabajo.

II. CONTENIDO

A. Planteamiento del problema

El desarrollo de este trabajo se basa en una empresa manufacturera ubicada en la ciudad de Cali, que se dedica a la producción de insumos para calzado. La situación problemática encontrada en la empresa está en la planificación y programación de la producción; dicha programación se basa en la asignación de trabajos en una serie de máquinas que funcionan en paralelo. Dado que existen restricciones en el número máquinas, fecha en la que se necesita que esté listo el trabajo, horas a la semana que están disponibles las máquinas y trabajos que no se pueden hacer en cierto tipos de maquinas; debido a esto se hace necesario el planteamiento de un modelo de optimización que se encargue de asignar dichos trabajos cumpliendo todas las restricciones.

A continuación en las siguientes tablas 1 y 2 se detallan algunas de las restricciones más relevantes del problema. La tabla 1, presenta si el producto k se puede hacer en la maquina j , toma valor 1 en caso afirmativo y 0 de lo contrario y la tabla muestra el tiempo de ciclo para elaborar una unidad del producto k .

La instancia del problema se plantea bajo los siguientes parámetros que son los inputs para el modelo de programación entera:

- Pedido del cliente por cierto producto para una fecha determinada
- Tiempos de ciclo de la máquina para hacer el producto
- Tiempos de alistamiento
- Tipos de productos que se pueden hacer en ciertas maquinas

Con estos requerimientos en la siguiente sección se presente el modelo de optimización.

B. Formulación del problema

El problema descrito anteriormente se puede enunciar de la siguiente manera:

	<i>I</i>	<i>2</i>	.	.	<i>j</i>	.	.	<i>J</i>
<i>I</i>	0	1	0	1	1	0	0	0
<i>2</i>	1	1	0	0	0	0	0	1
.	1	0	0	0	1	0	1	1
.	0	1	0	0	1	1	1	1
<i>k</i>	0	1	1	1	0	1	0	1
.	0	0	0	1	0	0	0	1
.	0	0	1	0	0	0	0	0
<i>K</i>	1	0	0	0	1	1	1	0

Tabla 1. Restricción maquina producto.

<i>1</i>	TC_1
<i>2</i>	TC_2
.	TC_3
.	TC_4
<i>k</i>	TC_5
.	TC_6
.	TC_7
<i>K</i>	TC_8

Tabla 2. Tiempos de ciclo.

Dado un conjunto de pedidos de diferentes clientes por ciertos productos y tallas para determinadas fechas de entrega, estos se deben programar en un número fijo de máquinas que se tienen para un horizonte de planificación de una semana de modo que se máxime el número de pedidos a cumplir en la semana cumpliendo una serie de condiciones y requerimientos. Dicho modelo se conoce como formulación estándar.

1. Modelo inicial: Formulación Estándar

Conjuntos

- $i \in I$ Conjunto de clientes
- $j \in J$ Conjunto de maquinas
- $k \in K$ Conjunto de productos
- $l \in L$ Conjunto de tallas
- $t \in T$ Conjunto de Turnos en que opera la planta

Parámetros

- $pedido_{ikl}$ Pedido del cliente i por el producto k talla l
- $tcliclo_{kl}$ Tiempo de ciclo para producir el producto k talla l
- rm_{kj} Si el producto k se puede hacer en la maquina j
- $timclien_{ik}$ Turno en que debe estar listo el producto k para el cliente i
- $talista$ Tiempo de alistamiento
- $horturno_{jt}$ Horas disponibles de la maquina j en el turno t

Variables

- X_{ijklt} Cantidad a producir del producto k talla l para el cliente i en la maquina j en el turno t .
- U_{jkl} Cantidad agregada a producir del producto k talla l en la maquina j en el turno t .
- Y_{ijklt} Variable binaria que indica si el producto k talla l se puede hacer en la maquina j para el cliente i en el turno t .
- V_{jkl} Variable binaria que indica si el producto k talla l se puede hacer en la maquina j en el turno t .

Formulación

Función objetivo: Maximizar la cantidad de producción semanal

$$Max Z = \sum_i \sum_j \sum_k \sum_l \sum_t X_{ijklt}$$

Restricciones

Cumplimiento de la demanda total o parcial para cada cliente

$$\sum_j \sum_t X_{ijklt} \leq pedido_{ikl} \quad \forall i, k, l \quad (1)$$

La suma por todos los clientes por el producto k talla l en el turno t , se fabrica en un solo bloque agregado

$$\sum_i X_{ijklt} = U_{jkl} \quad \forall j, k, l, t \quad (2)$$

Se produce el producto k talla l en la maquina j en el turno t , si el producto ya fue asignado a dicha máquina.

$$\sum_t Y_{ijklt} \leq V_{jkl} card(I) \quad \forall j, k, l \quad (3)$$

Se produce el producto k talla l en la maquina j turno t , si la variable binaria ya fue activa.

$$U_{jkl} \leq V_{jkl} M \quad \forall j, k, l, t \quad (4)$$

Se produce el producto k talla l para el cliente i en la maquina j turno t , si la variable binaria ya fue activa.

$$X_{ijklt} \leq Y_{ijklt} M \quad \forall i, j, k, l, t \quad (5)$$

Todo lo que se produzca en la maquina j , no debe superar las horas disponibles del turno.

$$\sum_k \sum_l (U_{jkl} tcliclo_{kl} + V_{jkl} talista) \leq horturno_{jt} \quad \forall j, t \quad (6)$$

Todo producto k talla l se produce en la maquina j turno t si ese producto se puede hacer en dicha maquina.

$$\sum_t V_{jkl} \leq rm_{kj} card(T) \quad \forall j, k, l \quad (7)$$

El producto k talla l se hace en el turno t , en una única maquina.

$$\sum_j V_{jkl} \leq 1 \quad \forall k, l, t \quad (8)$$

Se debe cumplir el turno en el que debe estar listo el pedido del cliente.

$$t \sum_j Y_{ijklt} \leq timclien_{ik} \quad \forall i, k, l, t \quad (9)$$

Todo pedido del cliente i , se asigna a lo más en dos turnos diferentes.

$$\sum_{jt} Y_{ijklt} \leq 2 \quad \forall i, k, l \quad (10)$$

Todo pedido del cliente i , se asigna en una sola maquina.

$$Y_{ijklt} + Y_{ij^*kl^*t^*} \leq 1 \quad \forall i, k, l, j \neq j^*, t, t^* \quad (11)$$

Si el pedido del cliente i se produce en 2 lotes diferentes, dichos lotes deben estar en turnos consecutivos.

$$Y_{ijklt} + Y_{ijkl^*t^*} \leq 1 \quad \forall i, k, l, j, t^* \leq t - 2, t > 2 \quad (12)$$

A lo más un producto se particiona por cambio de turno.

$$V_{jkl} + V_{jkl-1} + V_{jk^*lt} + V_{jk^*lt-1} \leq 3 \quad \forall k, l \neq l^*, j, t > 1 \quad (13)$$

A lo más un producto se particiona en 2 turnos consecutivos.

$$V_{jkl} + V_{jkl^*t^*} + V_{jkl^*t^*-1} \leq 2 \quad \forall k, l, j, t > 2, t^* = t - 1, t^* = t^* - 1 \quad (14)$$

Naturaleza de las variables

$$X_{ijklt} \in \mathbb{Z} \quad \forall i, j, k, l, t \quad (15)$$

$$U_{jkl} \in \mathbb{Z} \quad \forall j, k, l, t \quad (16)$$

$$Y_{ijklt} \in \{0,1\} \quad \forall i, j, k, l, t \quad (17)$$

$$V_{jkl} \in \{0,1\} \quad \forall j, k, l, t \quad (18)$$

El problema se resuelve como un modelo de programación entera (IP), en la sección de resultados se muestran la ejecución del modelo y la comparación de resultados en términos del valor de la función objetivo y tiempo de ejecución del modelo.

C. Resultados computacionales

Las instancias utilizadas para comparar los resultados computacionales de la formulación propuesta se muestran a continuación.

Instancia	Clientes	Productos	Maquinas	Pedido Total
A	7	8	2	7111
B	16	11	2	12260
C	22	12	3	18225

Tabla 3. Descripción de las instancias.

Ambas formulaciones fueron modeladas utilizando la herramienta Gams 23.6 y resuelto mediante el Solver CPLEX 10.0 en un ordenador con procesador Intel Core i3 2.53 GHz con 2 GB de memoria RAM.

A continuación se detalla los resultados computacionales, donde se mide la calidad de la solución y el tiempo de ejecución de la formulación evaluada en las tres instancias ya mencionadas.

Instancia	Solución			
	Objetivo	Entera	Gap	Tiempo
A	7111	7111	0%	00:13.984
B	12260	11976	0%	11:15.552
C	Out of Memory			

Tabla 4. Resultados computacionales de las tres instancias.

Se detalla que el modelo para instancias grandes presenta tiempos de ejecución más elevados, mientras que para resolver el problema no tiene inconvenientes y lo hace a optimalidad, como se puede observar con un Gap del 0%. Para el caso de instancias grandes (instancia C) el modelo presenta problemas de out of memory, debió a que el problema cae dentro de la familia de problemas combinatoriales complejos, conocidos como NP-Hard o Np Completo.

Para estos casos que es necesario implementar algoritmos que busquen soluciones con buenos tiempos de ejecución.

III. CONCLUSIONES

La utilización de la modelación matemática para problemas de programación de maquinas es una herramienta útil que permite asignar de manera más eficiente los distintos productos en las maquinas.

Se alcanzaron buenos resultados con la formulación presentada para el caso de instancias normales, mientras que para instancias grandes para más de 20000 unidades de producto el modelo presenta problemas por falta de memoria. Es de notar que este inconveniente se debe en gran medida al equipo computacional que se utiliza.

La implementación de algoritmos eficientes que busque soluciones con un número limitado de iteraciones, es la propuesta a utilizar para las instancias de mayor tamaño. Se recomienda para trabajos futuros la utilización del algoritmo de generación de columnas u otro tipo de algoritmo como el Branch and Price, capaces de resolver problemas con un número grande variables y un número pequeño de restricciones.

RECOMENDACIONES

Se sugiere plantear una formulación extendida que busque soluciones en tiempos de ejecución razonables.

Para esto se pueden basar en los siguientes temas:

- Eliminar restricciones redundantes
- Reducir la extensión de conjuntos
- Tratar de implementar algoritmos sofisticados para encontrar mejores soluciones

AGRADECIMIENTOS

Se le agradece a la empresa Formiplass S.A. por permitir llevar a cabo esta investigación, partiendo desde el Gerente General hasta todo el personal de planta.

REFERENCIAS

- [1] Z. Chen, W. Powell "Solving Parallel Machine Scheduling Problems by Column Generation", *Journal on computing*, Vol 11, pp. 78-94, 1999
- [2] C. Lee, Z. Chen "Scheduling of Jobs and Maintenance Activities on Parallel Machines", *Naval Research Logistics*, Vol 47, pp. 145-165, 2000

[3] J. Dai, G. WEISS “A Fluid Heuristic for Minimizing Makespan in Jobs Shops”, *Operation Research*, Vol 50, pp. 692-707, 2002

[4] J. Akker, C. Hoesel, M. Salvendy “A Polyhedral Approach to Single Machine Scheduling Problems”, *Mathematical Programming*, Vol 85, pp. 541-572, 1999

[5] J. Akker, C. Hurkens, M. Salvendy “Time Indexed Formulations for Machine Scheduling Problems: Column Generation”, *INFORMS Journal Computing*, Vol 12, pp. 111-124, 2000

[6] J. Lenstra, A. Rinnooy, P. Brucker “Complexity of Machine Scheduling Problems”, *Annals of Discrete Mathematics*, Vol 1, pp. 343-362, 1977

[7] Z. Chen, W. Powell “A Column Generation Based Decomposition Algorithm for a Parallel Machine Just in Time Scheduling Problem”, *European Journal of Operational Research*, Vol 116, pp. 221-223, 1999

[8] A. Ramudhin, J. Bartholdi, J. Calvin, J. Vande, G. Weiss “A Probabilistic Analysis of 2 Machine Flowshops”, *Operation Research*, Vol 44, pp. 899-908, 1996

[9] B. Hunsaker, A. Kleywegt, M. Salvendy, C. Tovey “Optimal Online Algorithms for Minimax Resource Scheduling”, *Journal on Discrete Mathematics*, Vol 16, pp. 555-590, 2003