El problema de la programación de las órdenes de producción

Alfredo Abarca Rojas

Escuela de Ingeniería Industrial y de Sistemas, ITESM Campus Toluca, Eduardo Monroy Cárdenas 2000, CP 50110, San Antonio Buenavista, Toluca, Estado de México, México

fabarca@itesm.mx

La mayoría de los modelos de programación y control de producción orientados a la búsqueda de una asignación eficiente de diferentes órdenes de producción a distintos puestos de trabajo, es quizás el problema de la Ingeniería Industrial que ha demandado quehacer a investigadores en el campo. Muchos de los escenarios planteados se calalogan con la etiqueta de problemas "no polinomiales completos" (NP complete) lo que implica que modelo sigue sin una solución general óptima. El objetivo del presente artículo es la presentación del problema de la asignación de órdenes de trabajo a máquinas (the sequencing and scheduling problem).

Introducción

Uno de los problemas de la Ingeniería Industrial que más trabajo ha demandado a investigadores es, sin duda alguna, el problema de cómo asignar eficientemente órdenes de producción a puestos de trabajo, que, en términos generales, se denomina el problema de asignación de cargas de trabajo[1]. Específícamente, este problema se refiere a la definición de la programación de un conjunto de trabajos que esperan ser procesados por una o por un conjunto de máquinas dispuestas en correcta secuencia tecnológica.

A pesar del incuestionable avance de la Ciencia y la Tecnología en cuanto a la generación de nuevas herramientas de trabajo, más confiables y más rápidas para buscar soluciones, el estado del arte en esta categoría de problemas, en su planteamiento general, aún está en estados primitivos hasta el punto en que se duda si este problema podría solucionarse.

Dada esta última cualidad, muchos investigadores han concentrado sus esfuerzos en el diseño de algoritmos heurísticos que conducen a "buenas soluciones". De ahí la proliferación de metodologías que ofrecen prometedoras alternativas para resolver planteamientos que, *a priori*, se sabe que no tienen solución.

[1] En este trabajo se entenderá por "asignación de trabajos a máquinas" al problema que en la literatura anglosajona se conoce con el nombre de "*the sequencing and scheduling problem*". No confundirlo con otro problema clásico de la Ingeniería Industrial denominado el problema de asignamiento, "*the assigment problem*".

Un problema NP completo

El problema de asignar cargas de trabajo a máquinas se cataloga como "problema no-polinominal completo", *NP complete*.

Un problema "NP" implica que la solución óptima al problema en cuestión -si existiera- se alcanzaría en un tiempo no-polinomial, mientras que un problema "NP completo" indica que existe una cantidad de escenarios que, con buen sentido común e imaginación, es fácilmente transportable a otros problemas[2].

En el caso concreto del problema de asignación de cargas de trabajo, -y por su naturaleza de ser problema NP completo- la resolución de este problema también resolvería problemas no menos famosos como el problema del agente viajero y el problema de la mochila, entre otros -por sólo citar dos- los cuales están planteados desde hace mucho tiempo para su solución óptima y eficiente.

Aplicaciones al problema de asignación de cargas

El problema de asignar cargas de trabajo a máquinas no tiene aplicación únicamente en ambientes de "máquinas" y "manufactura", por lo que sería un error abstraerlo sólo con conceptos de "equipo industrial" y "líneas de producción". Así por ejemplo, la resolución de este problema solucionaría planteamientos, entre otros a:

- ☐ Instituciones de enseñanza, donde, semestre a semestre debe definirse cuáles aulas se deben asignar a determinados cursos, considerando, entre otras, variables como número de alumnos, afinidad de profesores para ciertos cursos, tipos de aula y requisitos colaterales que deben estar plenamente satisfechos.

- ☐ Administradores de sistemas portuarios quienes deben determinar, entre otras cosas, cual embarcación debe anclar o no en determinado lugar.

- ☐ Médicos y enfermeras que laboran dentro de sistemas hospitalarios, cuando se requieren decisiones como cuál paciente debe reposar en determinado lugar, o bien, cuando es fundamental prever "y asignar" un determinado quirófano para una operación específica en un tiempo y lugar apropiados.

- ☐ Encargados de centros de las tecnologías de la información cuando se hace necesario definir cuál programa debe ejecutarse en determinada máquina para que el rendimiento del sistema sea el mejor.

Con este gran potencial del planteamiento, la solución óptima o cercana al óptimo del problema de asignación de cargas de trabajo aumentaría significativamente la productividad de los sistemas de producción, de bienes y servicios.

[2] Técnicamente dicho, un problema NP completo se resolvería en tiempo polinomial si y sólo si otro problema NP completo puede ser resuelto en un tiempo polinomial. Dado que, obviamente en este trabajo no se puede ahondar en la teoría de la complenitud, para una introducción al tema, véase, entre otros, Horowith et al (1978) Capítulos 11 y 12.

Descripción del problema

En primer lugar es prudente diferenciar las expresiones anglosajonas de *sequencing* y la de *scheduling*, que en castellano se acostumbra traducir indistintamente (y sin pensar en aglisismos) por "secuenciación" y/o "asignación".

Se entiende por *sequencing* la determinación de la apropiada permutación u ordenamiento de un conjunto de trabajos que deben ser procesados en más de una máquina. Por ejemplo, si se tienen - en un determinado proceso productivo- los trabajos "A", "B", "C", y "D", entonces algunas permutaciones factibles podrían ser {A-D-C-B}, {A-B-C-D} ó {A-B-D-C}. La búsqueda de *la permutación* que optimiza alguna medida de efectividad es lo que se denomina *sequencing*.

Por otro lado, la expresión *scheduling* se refiere a las especificaciones exactas en el tiempo en que ciertos eventos han de tomar parte en el proceso. Por ejemplo, la determinación del inicio de un determinado trabajo en una máquina dada es una respuesta típica de *scheduling*. En otras palabras, cuando dada una permutación, si el interés del analista se centra en el conocimiento de cuándo una actividad finaliza o empieza, o cuándo una actividad estaría en proceso en determinada máquina, o cálculos de tiempos de ocio, entre otros, entonces se dice que al analista le interesa el *scheduling*.

En este trabajo se utiliza la expresión castellana de *asignamiento* para referirse indistintamente a *sequencing* y al *scheduling*. Es importante anotar que, por lo general, en castellano se insinúa un significado más cercano a la última expresión que a la primera.

El modelo

Elmaghraby (1968) es quizás quien mejor define el problema de asignamiento: aquel planteamiento en donde se hace necesario definir un orden (en el sentido de prioridad, rango o importancia) dado un conjunto de trabajos (en el sentido amplio de la palabra) que esperan ser procesados por una o varias máquinas (o puestos o centros) en correcta secuencia tecnológica.

Bajo esta definición y en el contexto que nos interesa[3], el problema adquiere dos grandes ramas de estudio:

- En primer lugar, en escenarios netamente industriales donde el trabajo literalmente fluye de máquina a máquina en forma secuencial.
- En segundo lugar, en situaciones donde ese flujo de trabajo, por alguna restricción intrínseca al sistema productivo, el flujo de producto semiacabado se retroalimenta o se adelanta una o varias máquinas de la línea[4].

Al estudio de escenarios esbozados como primer área de estudio se le conoce con el nombre de *flow shop*. A los modelos de la segunda área, se les denomina *job shop*.

[3] Cualquier estudioso de la Ingeniería Industrial entenderá que la división es un tanto arbitraria porque bajo el marco de la asignación perfectamente pueden incluirse una gran cantidad de áreas de estudio como balance de líneas, CPM/PERT, transporte, etc.

[4] Es de rigor hacer notar que esta separación de conceptos, hoy ampliamente aceptados, no se considera en textos tan rewspetados –y tan excelentemente escritos- como Johnson y Montgomery (1974) ; véase la página 322 del referido texto.

Con el contexto anterior, el esquema que presenta la Figura 1 muestra un ambiente *flow shop* donde el trabajo semiacabado que ingresa a la máquina "i" proviene siempre de la máquina "i-1", mientras que la Figura 2 presenta al *job shop*, una variación bastante significativa al esquema anterior para un trabajo en cuatro, máquinas.

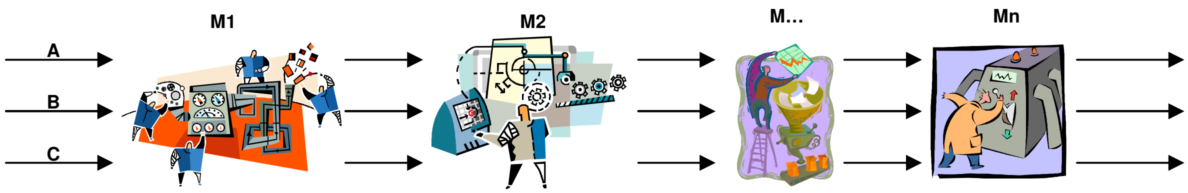


Figura 1: Un ambiente flow shop

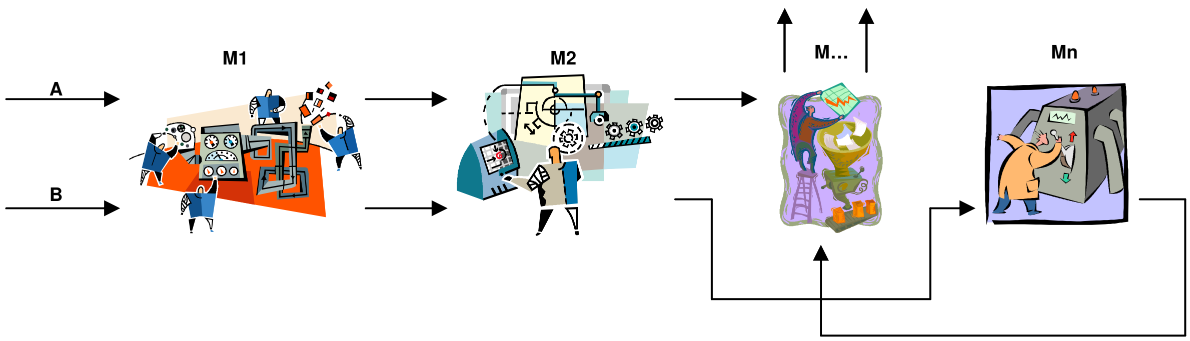


Figura 2: Un ambiente job shop

Para ambos escenarios, -y emulando a la teoría de las líneas de espera- se acostumbra una nomenclatura del tipo "n/m/l/k", donde "n" es el número de trabajos a procesarse, "m", la cantidad de máquinas del sistema, "l" una letra que distingue el escenario[5] , y "k" alguna medida de efectividad.

Por otra parte, y dentro del área del *flow shop*, se acostumbra discriminar, además, dos subconjutos de modelos:

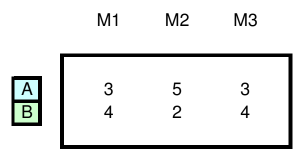
- ☐ Aquellos donde el orden requerido para procesar todos los trabajos es el mismo en todas las máquinas, y,

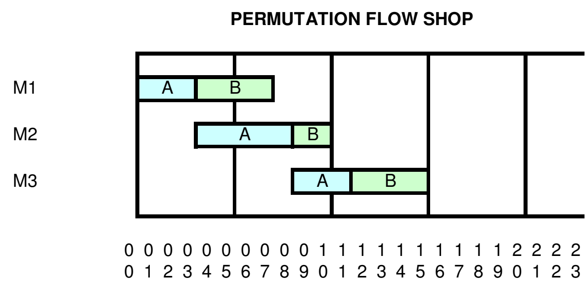
- ☐ Aquellos donde, además de requerirse el orden para procesar todos los trabajos, se exige, para todas las máquinas, que se proceda con un mismo orden para todos los trabajos.

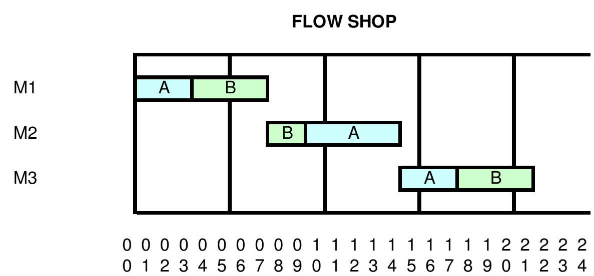
Para diferenciar estos dos escenarios de *flow shop*, el primero se reserva el nombre de *flow shop* a secas mientras que el segundo se acostumbra denominarlo comom *permutation flow shop*.

[5] Se acostumbra "F" para *flow shop*, "P" para *permutation flow shop* -como adelante se verá-, "G" para *job shop*,

Las figuras siguientes muestran dos ejemplos de programación del *flow shop* de tres órdenes de producción a dos máquinas. De ambos diagramas únicamente la primera es un auténtico *permutation flow shop*. Para efectos de claridad, en este trabajo no se hará la distinción entre los *flow shop* descritos.







Indistintamente de la división entre ambientes *flow shop* o *job shop*, existen modelos de trabajo que ayudan a concentrar la atención en la búsqueda de una o varias soluciones. Específicamente, se discriminan modelos determinísticos y estocásticos; modelos estáticos y dinámicos, o mezclas de modelos. Como los primeros modelos se definen por sí solos, a continuación sigue una breve descripción de los segundos.

- Los modelos estáticos son aquellos donde la información referente a "tiempos de procesamiento", "tiempos de entrega", "pesos ponderados", "tiempos entre-arribos" y demás variables que delimitan un sistema no cambian -o cambian en forma insignificativa- con el trascurrir del tiempo. Por ejemplo, si el quinto trabajo en la octava máquina dura veinte minutos, la escogencia de un modelo estático implica que, indistintamente al día de estudio, la duración "veinte minutos" continúa constante.

- Los modelos dinámicos son aquellas situaciones opuestas a las "estáticas", es decir, situaciones donde el transcurso del tiempo es atributo esencial para el cálculo de parámetros o variables.

Medidas de efectividad

Dado el esquema general de asignamiento, es de rigor especificar qué se desea optimizar. Por ejemplo, y entre otros,

- ☐ ¿Se desea una permutación que minimice el ocio de las máquinas?
- ☐ ¿Se desea minimizar el tiempo de permanencia de los trabajos en el sistema?
- ☐ ¿Se requiere que un conjunto de trabajos se procesen lo más pronto posible dado un período de tiempo?

Justamente la fijación de lo que se desea optimizar, es a lo que técnicamente se denomina "medida de efectividad". En ese contexto, Panwalkar y Iskander (1977) listan más de cien posibles medidas de efectividad, agrupándolas en categorías para claridad de exposición.

Para efectos ilustrativos, a continuación se describen cinco medidas de efectividad. Luego, con un ejemplo, se calcularían esas medidas.

Minimizando el tiempo máximo

Minimizar la "cantidad de tiempo" en que todos los trabajos son acabados en el sistema ("*the makespan*"), es quizás la medida de efectividad más estudiada sin que ello signifique que sea la medida de efectividad la más importante en todos los sistemas. Como su nombre lo indica, se pretende localizar la mejor permutación que asigne a las máquinas todos los trabajos de forma tal que el tiempo de finalización del último trabajo sea mínimo.

La suma de los tiempos de acabado

La "suma de los tiempos de acabado", *the sum of completion times*, es una medida de efectividad que optimiza la sumatoria de tiempos de finalización de todos los trabajos en todas las máquinas. Esta medida de efectividad es equivalente a la optimización del "promedio de tiempo" que duran los trabajos en un sistema dado.

Minimizar las esperas de los trabajos o minimizar la suma de las esperas de las máquinas

Dado un problema de asignación, pudiera ser que el sistema requiera minimizar las "esperas" que los trabajos han de tener merced a las disponibilidades de recursos. De igual forma, tambien pudiera ser que el sistema requiera minimizar el ocio de la máquinas para una mejor productividad del sistema de producción.

Miminizar la suma de los tiempos de tardanza

Por otro lado, un problema de asignación pudiera tener la restricción de "tiempos mínimos" para entregar el o los trabajos, por lo que se hace necesario minimizar las *tardanzas*[6] en brindar un servicio. En los sistemas hospitalarios, por ejemplo, se requiere optimizar esta medida de efectividad: el enfermo no puede esperar mucho tiempo por un servicio.

---

[6] Tardanza, en el contexto de asignación de cargas de trabajo, se refiere a la cantidad de tiempo que excede al momento en que se debe entregar el trabajo terminado.

Un ejemplo

Supóngase una situación *flow shop* con tres órdenes de producción y con dos máquinas, con tiempos de procesamiento de la orden "i" en la máquina "j" mostrados en la Tabla 1.

Tabla 1
Tiempo de procesamiento, T_{ij}, en minutos

Trabajos	Máquinas	
	M1	M2
A	6	1
B	4	5
C	3	2

Para efectos descriptivos, se parte de que una permutación aceptable es la {A-B-C}, esto es, primero se procesa la orden "A" luego la "B" y por último la "C"; con estas condiciones, se tiene una Gráfica de Gantt, tal como se muestra en la Figura 4. A continuación se determinarán las principales medidas de efectividad.

Figura 4
Gráfica de Gantt para la secuencia {A-B-C}. Tiempo en minutos

M1 A A A A A A B B B B C C C

M2 A B B B B B C C

5 10 15

La Figura 4 determina que la permutación {A-B-C} en un auténtico ambiente de *flow shop*, requiere de 17 minutos para que los tres trabajos completamente se procesen[7], por lo que se afirma que la "máxima cantidad de tiempo" que requiere esa permutación es de 17. La "suma de los tiempos de acabado", por otro lado, es 39 unidades de tiempo (7+15+17), lo que da un promedio de "entrega" de trabajos de 13 unidades (39/3).

La suma de "esperas para ser procesadas" en la máquina 1 es de 16 unidades de tiempo (0+6+10) lo que ocasiona un promedio de 5.33 minutos en promedio de espera a ser procesadas en el sistema de producción (en la máquina 1).

No existe tiempo de espera por parte de la primera máquina; la segunda máquina una vez que inicia su proceso está ociosa 3 minutso.

Por otra parte, la suma de tardanzas es de cero pues no se especifican tiempos de entrega. Pero si al mismo problema se le asigna un tiempos de entrega para todos los trabajos, de 15, por ejemplo, entonces la suma de tardanzas es de 2 minutos (0+0+2).

[7] En el Anexo, se presenta una forma eficiente de calcular estos resultados.

Lo pertinente es, por tanto, definir qué se desea optimizar y, dependiendo de ello, se abre un espectro de soluciones (y de complicaciones).

Variables a considerar

Dada una secuencia de "n" trabajos $\{J_1, J_2, J_3, ..., J_n\}$ que es necesario programar en un conjunto de "m" máquinas $\{M_1, M_2, M_3, ..., M_n\}$ no es difícil prever la existencia de $(n!)^m$ posibles permutaciones $\{J_{P1}, J_{P2}, J_{P3}, ..., J_{Pn}\}$, donde $P_1, P_2, P_3, ..., P_n$ es el ordenamiento de la permutación. Una -o varias- de estas permutaciones, eventualmente, sería la solución óptima al problema de asignación de cargas de trabajo, sin que necesariamente se descarte la posibilidad de que algunas de esas permutaciones pudieran ser no-factibles por restricciones tecnológicas. Obsérvese cómo, si se supone que el orden de procesamiento de todos los trabajos fuese cosntante en el sistema, entonces el conjunto de permutaciones se reduciría a n!.

Para el planteamiento de cualquier problema de asignación de cargas de trabajo, es importante definir el marco de referencia que se desea atacar; con ello se delimita la cantidad de variables que es necesario considerar. Así por ejemplo, se hace necesario analizar marcos de referencia respecto a trabajos, máquinas, e inclusive, respecto a tiempos de procesamiento. Algunas ideas generales en ese sentido se dan a continuación (Abarca 1984).

Consideraciones respecto a trabajos

- ☐ Definir, en primer lugar, si el problema de asignación se ubica dentro de un escenario típico del *flow shop* o del *job shop*.
- ☐ Determinar la disponibilidad de los trabajos, por ejemplo, ¿nacen los trabajos únicamente a ciertas horas del día o el nacimiento de éstos se ubica acorde con alguna distribución de probabilidad? ¿Cuándo están disponibles los trabajos para su eventual procesamiento? ¿Cuál es el patrón de arribo de los trabajos a las máquinas si éste existe? ¿Arriban los trabajos al sistema bajo un distribución de probabilidad?
- ☐ ¿Son los trabajos estadísticamente independientes entre sí?
- ☐ ¿Es posible que los trabajos se procesen en varias máquinas simultáneamente?
- ☐ Especificar la continuidad del procesamiento. ¿Es posible detener la máquina aún si ésta no ha terminado su actividad?
- ☐ ¿Es posible que los trabajos esperen en cola el procesamiento en una determinada máquina?
- ☐ Definir el nivel de "terminado" en cada "estación de trabajo". ¿Pueden los trabajos ser procesados en la siguiente máquina aún sin terminarse la ejecución en la máquina previa? ¿O eso es imposible de darse?

Consideraciones respecto a máquinas

- ☐ Conocer la disponibilidad de las máquinas en la "línea de producción". ¿Se dispone de máquinas "todo el tiempo" o ellas están a disposición en forma intermitente?

- ☐ ¿Pueden o no las máquinas ejecutar más de una operación a la vez?
- ☐ ¿Es posible que las máquinas estén ociosas?
- ☐ Analizar las posibilidades de interrupciones de máquinas. ¿Es usual la "falla" de máquinas?
- ☐ ¿Se puede afirmar que el funcionamiento de las máquinas son estadísticamente independientes una de la otra?

Consideraciones respecto a los tiempos de procesamiento

- ☐ Determinar si los tiempos de "puesta en operación" están o no incluidos en los "tiempos de procesamiento" de cada trabajo en cada máquina.
- ☐ Estar conciente de la naturaleza de los "tiempos de procesamiento" de cada uno de los trabajos en las máquinas. ¿Son esos tiempos determinísticos? ¿Son variables aleatorias que se comportan bajo una distribución de probabilidad conocida?
- ☐ ¿Puede afirmarse que los tiempos de procesamiento son estadísticamente independientes entre sí?
- ☐ ¿Se han considerado elementos extraños en el cálculo de los tiempos de procesamiento?

En busca de soluciones

A pesar de que la Gráfica de Gantt es quizás el primero, el más conocido y el más efectivo medio de resolver gráficamente el problema de asignar cargas de trabajo a máquinas, -esta metodología data desde 1919-, desde que Garey *et al* (1976) demostraron en 1976 que dos problemas, a saber,

- El problema relacionado con la minimización de la media del tiempo de flujo, en un ambiente *flow shop*, considerando dos máquinas y más de un trabajo, y,
- El planteamiento que busca el cronograma de trabajo más corto en un escenario de "m" máquinas y en más de dos trabajos

son problemas NP-Completos, se ha justificado e incentivado la investigación en pos del diseño de heurísticos, esto es, de metodologías no necesariamente óptimas que predicen satisfactoriamente la solución. Estas soluciones heurísticas evidentemente constituyen una medida alterna para alcanzar una solución (que algunas veces resulta la solución óptima[8]).

Desde otro frente, se ha intentado resolver el problema mediante algoritmos óptimos basados en las metodologías de "ramificaciones y cotas" (*branch and bound*) que dan soluciones óptimas, pero tienen el inconveniente de que en última instancia resultan algoritmos del tipo exponencial lo que, en esencia, significa que computacionalmente son ineficientes.

[8] Mención aparte ha de hacerse al "algoritmo de Johnson", aquel elegante procedimiento que optimiza el *makespan* de un ambiente *permutation flow shop*, para n trabajos en dos máquinas. Véase Johnson (1954).

Usando simulación, Byung Park (1899) ha evaluado soluciones generadas por heurísticos muy citados en la literatura, y, por ende más utilizados en la asignación de cargas de trabajo orientados hacia ambientes *flow shop*; utilizando SLAM II, y concentrándose en planteamientos que buscan una minimización del tiempo de entrega, concluye que, en efecto, los heurísticos son una vía muy promisoria para solucionar problemas de asignación.

Investigaciones como esta última abundan en la literatura y no concluyen nada novedoso. Lo que se desea resaltar al citar el trabajo de Byung Park, es el hecho de que aún no existe una solución óptima global y computacionalmente eficiente al problema de asignación de cargas de trabajo, de forma tal que los heurísticos constituyen entonces la *única* vía de solución para tiempos de ejecución razonables.

Análisis del peor caso

Pero no basta con desarrollar heurísticos a diestra y siniestra. Se requieren heurísticos eficientes que predigan al usuario que tan cerca o que tan lejos está la solución alcanzada respecto al óptimo cuyo valor se desconoce. Es decir, asociar al valor de la solución, un nivel de garantía del peor caso esperado del heurístico, labor que, como es de suponer, es bastante complicada.

Concretamente, dado un contexto de un problema y un heurístico que genera una solución, junto a esta solución sería muy deseable conocer cual es el probabilidad del peor caso que genera el algoritmo en el contexto dado.

Conclusión

Dada la imposibilidad actual de encontrar soluciones óptimas al problema de asignación de cargas de trabajo a puestos de trabajo, -merced a su etiqueta de problema NP-Completo-, se hace necesario retomar el diseño de heurísticos eficientes y capaces de predecir "buenas" soluciones. Y si a la solución se le asocia un nivel de garantía del peor caso esperado del heurístico, mucho mejor.

Referencias

1. Abarca, A., "An Efficient Heuristic that Determines a Schedule. Minimizing Sum of Completion Times", disertación doctoral, Lehigh University, Pensilvania, 1984, sin publicar.

2. Byung Park, Y., "An Evaluation of Static Flowshop Scheduling Heuristics in Dynamic Flowshop Models via a Computer Simulation", Computers and Industrial Engineering, 14(2):103-112, 1988.

3. Elmaghraby, S.E., "The Machine Sequencing Problem. Review and Extensions"; Naval Research Logistic Quarterly, 15(2):205-232, junio de 1968.

4. Garey, M.R., Johnson, D.S., Sethi, R., "Complexity of Flow Shop and Job Shop Scheduling", Mathematics of Operations Research, 1(2):117-119, mayo de 1976.

5. Horowith, E., Sahni, S., "Fundamentals of Computer Algorithms", I edición, Computer Science Press, Maryland, 1978, capítulos 11 y 12.

6. Johnson, S.M., "Optimal Two and Three Stage Production Schedules with Setups Times Included", Naval Research Logistic Quarterly, 1(1):61-68, enero 1954.

7. Johnson, L.A., Montgomery, D.C., "Operations Research in Production Planning, Scheduling, and Inventory Control", I edición, John Wiley & Sons, Nueva York, 1974.

8. Panwalker, S.S., Iskander, "A Survey of Scheduling Rules", Operations Research, 25(1):45, enero 1977.

ANEXO

Se presenta a continuación una mánera eficiente para calcular tiempos contenidos en una gráfica de Gantt. Supóngase la siguientes tabla de tiempos de procesamiento.

Trabajos	M1	M2	M3
A	6	2	9
B	7	1	8
C	12	11	13
D	15	16	4
E	5	17	18
F	14	10	3

Si se tuviera que calcular los tiempos que generaría una Gráfica Gantt para la secuencia {A-B-C-D-E-F} se procede de la siguiente forma

Secuencia	En (M1)	En (M1+M2)	En (M1+M2+M3)
A	6	6+2=8	6+2+9=17
B	6+7=13	Max(13,8)+1=14	Max(14,17)+8=25
C	6+7+12=25	Max(25,14)+11=36	Max(36,25)+13=49
D	6+7+12+15=30	Max(30,36)+16=52	Max(52,49)+4=57
E	6+7+12+15+5=35	Max(35,52)+17=69	Max(69,57)+18=87
F	6+7+12+15+5+14=4	Max(48,69)+10=79	Max(79,87)+3=90

De donde se deduce que, entre otros, la secuencia {A-B-C-D-E-F} tendrá un *makespan* de 90 y que la suma de tiempos de terminación de todos los trabajos es de (17+25+49+57+87+90).