

## CAPITULO I

### INTRODUCCION

Existen en la actualidad, diferentes investigaciones que se enfocan a la programación de maquinaria de un sistema productivo. Las propuestas de dichas investigaciones tienen como restricción alguna característica en especial como lo es, la fecha de entrega fija, el tamaño de lote, el tiempo de ciclo, el número de máquinas, etc. Durante el presente capítulo, se incluyen algunas de las últimas investigaciones publicadas sobre este tema. El propósito es ubicar al lector en lo que se ha escrito recientemente y en base a esto, reconocer las áreas de oportunidad para que el algoritmo propuesto en el capítulo dos, pueda tener su justificación y así complementar lo que hasta la fecha se ha presentado.

#### **1.1 Investigaciones sobre secuenciación y programación de maquinaria**

##### **MINIMIZING CICLE TIME IN A BLOCKING FLOWSHOP (1)**

En este artículo, los autores consideran un *blocking flowshop* (*flowshop* sin *buffers*) donde cada máquina procesa una pequeña parte del producto con el fin de minimizar los tiempos de ciclo o incrementar el *throughput*. Anterior a este artículo, los autores presentaron un procedimiento heurístico que resolvía problemas con 9 máquinas y 25 trabajos, con un error relativo promedio de 3%, pero en algunas ocasiones el error se incrementaba

hasta 10%. La idea de disminuir el procesamiento de operaciones, establece una conexión matemática precisa entre este problema y un *flowshop* sin esperas. Esto permite que un procedimiento heurístico para *flowshops* sin esperas, se adapte como un procedimiento heurístico para un *blocking flowshop*. Los resultados presentados en este artículo muestran errores relativos promedio menores al 2% para problemas con 20 máquinas y 250 trabajos.

#### SCHEDULING OF MULTI-CLASS SINGLE-SERVER QUEUES UNDER NONTRADITIONAL PERFORMANCE MEASURES (2)

El modelo básico que proponen los autores es un centro de maquinado donde una sola máquina, procesa productos de varios tipos. Asumen que no existe retroalimentación entre los diferentes clases y que los arribos de los diferentes tipos de trabajos son mutuamente independientes. Consideran ambientes donde la producción es "Make to Order". Comentan que la política para secuenciar trabajos SEPT (Smallest Expected Processing Time) minimiza el promedio de tiempo de *throughput* para este tipo de sistemas (Wolf 1989), así como la política FIFO (First In First Out) minimiza la varianza del *throughput* en comparación con las otras políticas existentes.

Ellos consideran un sistema de producción multiclases sin tiempos de preparación, donde varias clases de trabajo comparten un solo servidor. El parámetro tradicional para secuenciar estos sistemas es la media del tiempo de *throughput*. Sin embargo, la media del tiempo de *throughput* puede no ser el único parámetro de importancia en un sistema real. En particular, la varianza del tiempo de

*throughput* y los percentiles del tiempo de *throughput* pueden ser igualmente importantes. Presentan dos procedimientos heurísticos que están basados en un análisis de tráfico pesado que poseen un buen comportamiento con respecto a estos parámetros no convencionales.

### THE BATCH LOADING AND SCHEDULING PROBLEM (3)

En esta investigación se analiza la formación de lotes y secuenciado para ciertos tipos de estaciones de trabajo que operan por lotes. Unos ejemplos de estas estaciones son las estaciones de tratamiento térmico, particularmente en la industria del acero y en la industria de la cerámica, así como en una variedad de operaciones en la manufactura de circuitos integrados. En el problema que se presenta, hay un conjunto de trabajos esperando a ser procesados. Cada trabajo es asociado con una familia y tiene un peso (un costo por retraso) y un volumen a producir. El programador debe organizar los trabajos en lotes en los cuales cada lote contiene trabajos de una misma familia y en el cual el volumen total de trabajos en el lote no excede la capacidad de la estación. El programador debe entonces secuenciar todos los lotes. El tiempo de proceso para el lote depende sólo del tipo de familia y no de la cantidad o el volumen de trabajos en cada lote. El objetivo es minimizar la media del tiempo de flujo.

El artículo presenta una formulación de programación entera para este problema, genera un límite inferior en base a una ejecución parcial de programación lineal,

propone un algoritmo polinomial, y prueba un conjunto de heurísticos del problema general. La habilidad de organizar los trabajos en lotes es la clave para lograr soluciones eficientes y es la base para los diferentes procedimientos de solución de este artículo. Entre los heurísticos que utilizan, está el "greedy heuristic", "successive knapsack heuristic" y el "generalized assignment heuristic".

Las conclusiones del estudio realizado en computadora muestran que el "successive knapsack heuristic" y el "generalized assignment heuristic" funcionan mejor que el "greedy heuristic". El "generalized assignment heuristic" es ligeramente mejor en algunos casos que el "successive knapsack heuristic", sin embargo es más rápido y más robusto. Para problemas con pocos trabajos, el "successive knapsack heuristic" y el "generalized assignment heuristic" casi siempre proveen soluciones óptimas. Para problemas con más trabajos, los autores compararon los valores de las soluciones de estos heurísticos con el límite inferior obtenido con la programación lineal. El trabajo por medio de la computadora sugiere que los heurísticos continúan proporcionando soluciones que son óptimas o cercanas al óptimo. El estudio también muestra que el volumen de trabajo relativo a la capacidad de la estación y el número de trabajos en una familia afecta al funcionamiento de los heurísticos, mientras que el número de familias no lo hace. Finalmente proveen un análisis del peor caso para el heurístico "greedy".

Cabe señalar que este artículo basa sus estudios y análisis en un sistema que consiste en una sola máquina.

#### SCHEDULING WITH FIXED DELIVERY DATES (4)

En la mayoría de los modelos clásicos de programación de una estación, (scheduling), se asume que un trabajo se entrega al cliente inmediatamente después de que este termina de ser procesado. Sin embargo, en muchos casos prácticos, un conjunto de fechas de entrega pueden ser establecidas antes de que los trabajos se procesen. Esto es particularmente relevante en los casos en que la entrega es una operación costosa o complicada. Una situación similar se presenta cuando los clientes requieren que las entregas se realicen en un intervalo limitado de tiempo que se repite periódicamente. Una tercera posibilidad es que una función comercial periódica, por ejemplo el ciclo de pagos, definitivamente defina un tiempo de entrega, e incluya todos los trabajos que han sido completados desde el último ciclo de pago. Estas situaciones suelen no ser representadas adecuadamente por los modelos clásicos de programación de maquinaria. Los autores toman en cuenta una variedad de problemas en los cuales un trabajo es entregado al cliente en la fecha de entrega más próxima que no es antes que el tiempo que se requiere de proceso para completar el trabajo. En este artículo se estudian los problemas donde el número de entregas es constante durante un periodo (por ejemplo 40 entregas al mes), y también aquellos donde el tiempo de entrega es considerado dentro de los datos de entrada. Se presentan comparaciones entre la solución de estos problemas con tiempos de entrega establecidos y los correspondientes problemas clásicos de programación de maquinaria.

EFFECTIVE SEQUENCING RULES FOR CLOSED MANUFACTURING NETWORKS (5)

El objetivo al controlar una red cerrada de maquinaria es minimizar el tiempo ocioso de las máquinas. En un sistema donde los recursos son compartidos por varios productos, las políticas de secuenciado pueden tener un impacto significativo en el funcionamiento de la red de proceso. El autor en este artículo usa el modelo aproximado de Brownian, que le ayuda a desarrollar políticas que disminuyen el promedio de tiempo ocioso en los recursos. A pesar del hecho de que el modelo Brownian se basa en el tráfico pesado (altos niveles de utilización de recursos), el procedimiento de programación de maquinaria propuesto proporciona mejoras para sistemas con niveles más moderados de utilización de la capacidad. Chevalier y Wein (1993) desarrollaron también una política estática basada en el análisis del modelo aproximado de Brownian, y demostraron los beneficios de su política sobre muchos modelos tradicionales de programación. En este caso, el autor argumenta que inclusive la política Chevalier-Wein puede ser mejorada, ya que en algunos casos esta provoca tiempo ocioso en el servidor. Se muestra vía simulación, que la política propuesta reduce el tiempo de *throughput* por más del 20% en comparación con la política de Chevalier-Wein, y que este beneficio se incrementa a medida que aumenta la utilización en los servidores. La magnitud del beneficio permanece casi igual cuando se mide el beneficio en términos de reducción del trabajo en proceso.

SCHEDULING WITH INSERTED IDLE TIME: PROBLEM TAXONOMY AND  
LITERATURE REVIEW (6)

La gran mayoría de las investigaciones en programación de maquinaria se ha concentrado en la construcción de programas sin retrasos. Sin embargo, los autores de este artículo se interesan en una clase más general de programación llamada "inserted idle time schedule (IIT)". Una programación sin retrasos fue propuesta por Baker (1974). Esta es una programación factible en donde ninguna máquina se queda ociosa al mismo tiempo que puede comenzar a procesar una operación. Los autores definen "inserted idle time schedule (IIT)" como una programación factible en donde una máquina puede quedar ociosa al mismo tiempo que puede comenzar a procesar una operación. Conway et al. (1967) ha mostrado que para ciertos problemas no es necesario considerar el tiempo ocioso.

La programación IIT es particularmente relevante en situaciones industriales donde se consideran más de una máquina, donde los costos y las llegadas continuas de trabajos con tiempos de entrega establecidos, juegan un papel importante.

Los autores presentan en este artículo la taxonomía de los ambientes en donde la programación IIT es relevante, también la extensa literatura que se ha publicado acerca de este tipo de programación, y por último, identifica áreas de oportunidad para futuras investigaciones.

Como se puede observar al revisar los artículos aquí presentados, existe un gran interés por resolver los

problemas que se presentan en la industria con respecto a la programación de la maquinaria en el piso productivo o el secuenciado de los trabajos antes de entrar al sistema.

Las investigaciones publicadas abarcan desde una sola máquina (aplicable a un centro de maquinado) hasta un conjunto de varias estaciones, involucran *buffers*, lotes, y tiempos de entrega. Sin embargo, después de una búsqueda en las diferentes revistas especializadas que existen en el medio, no se encontró un algoritmo que abarcara diferentes aspectos al mismo tiempo. De ahí la necesidad de realizar la propuesta de un algoritmo que sí los abarcara.

Debido a la complejidad del problema que se pretende resolver, se excluye la solución por medios matemáticos puros. Esta restricción es tal vez la principal, por la que la mayoría de los artículos sólo abarquen un enfoque más cerrado en sus modelos, desechando circunstancias que en la realidad, están íntimamente relacionadas unas con otras. Estas relaciones hacen al sistema productivo un ente viviente que cambia constantemente conforme pasa el tiempo, cuando cada una de las circunstancias se va presentando durante el proceso de los trabajos.