Artículo No. 03: Vanegas, C., *et al.* (2014). Aplicación del método "*branch and bound*" para la programación de producción en una empresa metalmecánica bogotana. *Semilleros*, 1 (2), 27-42.

Aplicación del método "*branch and bound*" para la programación de producción en una empresa metalmecánica bogotana

Aplication of branch and bound method in the production scheduling for a metallurgical company in Bogotá

Vanegas Torres Claudia Natalia [1], Garzón Gutiérrez Alexander Enrique [2], Muñoz Peña Fernando [3], Palacio León Oscar [4]

Resumen. La asignación de cargas de trabajo es un tema complejo para los gerentes de operaciones en general, agudizándose aún más en ausencia de un modelo metodológico para trabajo colaborativo entre los eslabones proveedor-comprador inscritos en una misma red de suministros dirigida por la demanda. Esta investigación considera el problema de programación de operaciones tipo taller en donde el poder económico está concentrado en el agente logístico que demanda el servicio de manufactura. Este proceso de integración empresarial fue desarrollado mediante el uso de una metodología ágil para proyectos industriales soportada a través del método Branch and Bound, el cual tiene por meta minimizar el tiempo total de fabricación de una familia de productos en concordancia a la velocidad que experimenta su demanda.

Palabras clave*: Método branch and bound, makespan, programación de flujo.*

Abstract. The allocation of workloads is a complex issue for operations managers, to worsen further in the deficiency of a methodological model for collaborative work between supplier and buyer links. This research considers the problem of scheduling operations type workshop where economic power is concentrated in the logistics agent that demand the manufacturing service. This process of enterprise integration was developed using an agile methodology for industrial projects supported through the Branch and Bound method, which aims at minimizing the total manufacturing time of a family of products according to the speed you experience your demand.

Key Words*: Branch and bound method, makespan, flow programming.*

[1] Ing. Industrial de la UMNG (Colombia). Universidad Militar Nueva Granada, Bogotá, Colombia. natu.vanegas@gmail.com
[2] Ing. Industrial, Especialista en Gerencia en Logística Integral de la UMNG (Colombia).Universidad Militar Nueva Granada, Bogotá, Colombia. alexgar75@gmail.com
[3] Ing. Industrial, MBA, Profesor asistente, Facultad de Ingeniería, Investigador grupo PIT Universidad Militar Nueva Granada, Bogotá, Colombia. fernando.munoz@unimilitar.edu.co
[4] Ing. Industrial, M.Sc, Doctorando en Ingeniería de la UC (Venezuela), Doctorando en Proyectos de la UNINI (México), Profesor asistente, Facultad de Ingeniería, Investigador grupo PIT Universidad Militar Nueva Granada, Bogotá, Colombia. Oscar.palacio@unimilitar.edu.co

INTRODUCCIÓN

La determinación de las secuencias específicas de trabajo a realizarse en cada línea de producto o tipo de referencia en una planta de manufactura es uno de los aspectos más relevantes a considerar en la programación de producción, ya que por su naturaleza dinámica y bajo los nuevos esquemas de competitividad se exige una respuesta rápida por parte del sistema a los requerimientos de los clientes, esto es, minimizar el tiempo de procesamiento de los diferentes pedidos pero al mismo tiempo optimizar el uso de los recursos al interior de la planta.

Respecto al desarrollo del tema, se ha evidenciado que los estudios realizados por el profesor Michael Pinedo de la Universidad de Nueva York en los últimos 25 años en el área de administración de operaciones han sido de gran reconocimiento, ya que ha sido uno de los investigadores con mayor especialización en el tema de la planeación y secuenciación de sistemas productivos y de servicios, más conocido en la literatura como Scheduling, en su trabajo se describe que en todos los problemas de programación el número de puestos de trabajo y el número de máquinas se supone que son finitos, es decir, Pinedo, ha abordado de manera más profunda los desarrollo preliminares realizados en los años 50 (French, 1982) y a través de este tipo de herramientas, se ha logrado mejorar el uso de recursos existentes en sectores como manufactura, minería y agricultura, entre otros (Pinedo, 2008). Es importante mencionar que el método branch and bound (ramificación y poda) es una técnica general de diseño algorítmico utilizada para buscar la solución óptima para situaciones de programación de producción tipo taller. La idea principal consiste en dividir el problema inicial en sub-problemas parciales de tamaño más pequeño en los que se aplican heurísticas de poda que permiten descartar el espacio de soluciones del problema que no conduce a la solución óptima (Chinneck, 2010) (Aleman, 2002). Teniendo en cuenta que todos los trabajos tienen que seguir la misma ruta de fabricación, es decir, tienen que ser procesados por primera vez en la máquina 1, luego en la máquina 2, y así sucesivamente. Por lo general, todas las colas se supone que operan bajo la disciplina first in first out (FIFO), es decir, un trabajo que no puede "pasar" a otro mientras espera en una cola (Pinedo, 2008) (Chinneck, 2010).

El problema de flujo regular ha sido uno de los temas que más intensamente se ha venido investigando en programación desde la publicación de Johnson en 1954 que describía un entorno de trabajo Fm/prmu/Cmax (permutation flow shop scheduling). Si el número de máquinas es de dos, es fácil alcanzar una secuencia óptima de fabricación para varios trabajos en un tiempo determinado usando el algoritmo de Johnson. Garey et al en 1976 (Garey M.R., 1976) demostraron que para un sistema flowshop conformado por tres máquinas es un problema NP-hard (Pascual, 2005). Entre tanto, Lomnicki (1965) (Lomnicki, 1965) e Ignall y Schrage (Ignal, Schrage, 1964) (1965) publicaron, casi simultáneamente, los primeros algoritmos de branch and bound para el problema Fm/prmu/Cmax (permutation flow shop scheduling). Desde entonces se han publicado otros algoritmos de branch and bound: Potts (1980), Companys (1999), entre otros (Pascual, 2005) (Aleman, 2002).

La complejidad del problema Fm/prmu/Cmax (permutation flow shop scheduling) ha producido el desarrollo de procedimientos heurísticos. Algunas de las heurísticas son Palmer (1965), Companys (1966), Campbell (1970), Gupta (1971), Dannenbring (1977), Nawaz (1983), etc. A

esta lista se pueden añadir gran número de aplicaciones de los procedimientos de exploración local comúnmente denominados meta heurísticas (Pascual, 2005) (Companys, 2005). Recientemente Ladhari y Haouari (Haouari, 2004) presentaron un algoritmo de branch and bound que construye las permutaciones colocando alternativamente piezas en ambos extremos. Utilizan un sub-problema de 2-máquinas como procedimiento de acotación, que proporciona una cota inferior muy ajustada, pero no se puede calcular en tiempo polinómico. El algoritmo de Ladhari utiliza la heurística NEH para obtener un valor como mejor solución inicial, o, para instancias más duras, una heurística de búsqueda basada en un original esquema de branch and bound (Haouari y Ladhari, 2003) (Pascual, 2005) (Companys, 2005).

La principal contribución de este artículo se fundamenta en el desarrollo de un modelo metodológico para la integración empresarial entre los eslabones proveedor-comprador que este inversa en los principios de las escuelas de Taniguchi (Taniguchi, 2001) y Yin (Yin, 2008). Que permita la programación de operaciones tipo taller de manufactura flowshop para ambientes mypime que refleje el poder económico que posee el agente logístico que demanda el servicio. Garantizando el equilibrio entre el makespan y el Takt Time a través del método branch and bound.

El documento se organiza como sigue. En la Sección 1, Materiales y Métodos, se presentará: i) Programación secuencial; ii) Fundamentos del algoritmo de Lomnicki; iii) Descripción del algoritmo de Branch And Bound o de Ignall-Schrage y Lomnicki (N/3/F/FMAX); y iv) Modelo metodológico propuesto. En la Sección 2, Resultados y análisis, se analizarán y discutirán los principales resultados, los cuales arrojan que: i) Pasos básicos para la ejecución de un modelo con Branch And Bound; y ii) Aplicación de metodología para la industria metalmecánica (Empresa Darwis Colombia Limitada). En la Sección 3, Conclusión, se exponen los beneficios que brinda la propuesta metodológica y se hace algunas conclusiones relevantes en torno a la revisión bibliográfica. Finalmente, en la Sección 5, Investigaciones futuras, se hace un acercamiento de las líneas de investigación que pueden abordarse, a partir del presente estudio.

MATERIALES Y MÉTODOS

Programación secuencial

Los modelos secuenciales tienen aplicaciones principalmente en un taller de tareas, donde un conjunto de máquinas, de propósito general, ejecutan una serie de operaciones sobre las órdenes de trabajos o tareas de producción. (Little, 1966). Las tareas son a menudo únicas y ordenadas por un determinado cliente. El proceso en el taller de tareas es un modelo fundamental para un considerable número de sistemas operacionales, tales como las actividades de mantenimiento o la programación de “n” tareas en “m” máquinas en un taller de fabricación intermitente, etc. (Pinedo, 2008) (Companys, 2005).

La programación en un taller de tareas consiste en determinar el orden o la secuencia de las tareas en las máquinas que permitan optimizar alguna medida de ejecución. (Little, 1966)

Existen cuatro factores que describen y clasifican un problema específico de programación en un taller de tareas de acuerdo con:

- El patrón de llegada de los trabajos. Si “n” tareas llegan simultáneamente al taller y quedan disponibles para iniciar su proceso se

tendrá un problema de programación estática, si las tareas llegan intermitentemente, posiblemente de acuerdo a un proceso estocástico, el problema de programación es dinámico. (Bautista, 2008) (Sipper, 1998)

- El número de máquinas que integran el taller. Existe un problema de secuenciación cuando "*n*" trabajos son programados en "*m*" máquinas. (Little, 1966) (Bautista, 2008)
- El flujo de producción. El flujo de proceso de las tareas a través de las máquinas debe ser especificado, si todas las tareas siguen la misma ruta el flujo de producción es continuo o en serie. El extremo opuesto, donde no existe una ruta preconcebida de procesos se tiene un taller cuyo flujo de producción es aleatorio. Los trabajos pueden ser independientes unos de otros, o bien interdependientes. Cuando se mezclan los diferentes tipos de flujos de producción, los de serie con los aleatorios, existen rutas generales de proceso. (Little, 1966) (Bautista, 2008)
- El objetivo que se desea optimizar. La medida de desempeño que frecuentemente se utiliza y evalúa es la optimización del tiempo total de proceso de todas las tareas o trabajos en todas las máquinas. Otras medidas de ejecución son: el tiempo promedio de permanencia de los trabajos en el taller, impuntualidad, el tiempo promedio de espera, etc. (Little, 1966) (Bautista, 2008)

Para identificar los diferentes problemas de secuenciación Conway, Maxwell y Miller proponen una notación de cuatro parámetros escritos así: **A/B/C/D.** (Sipper, 1998)

Dónde:

A: Describe las llegadas de las tareas al proceso. Para problemas dinámicos **A** identifica la distribución de probabilidad de los tiempos entre llegadas, si se trata de un problema estático **A** denota el número de trabajos, estas tareas llegan simultáneamente. (Sipper, 1998)

B: Identifica el número de máquinas en el taller. (Sipper, 1998)

C: Define el tipo de flujo de producción en el taller. Puede ser en serie **(F)**, aleatorio **(R)**, y general **(G)**.Cuando se tiene un taller con una sola maquina no hay ningún tipo de flujo y el tercer parámetro es ignorado. (Sipper, 1998)

D: Describe el criterio de optimización. Por ejemplo (**n/3/F/Fmax**) indica programar "**n"** tareas en serie en tres máquinas tal que se minimice el tiempo de flujo máximo. Este ejemplo define el algoritmo de branch and bound. (Sipper, 1998)

Para fijar parámetros de modelos secuenciales se consideran los siguientes Supuestos:

- Cada máquina elabora un proceso a la vez.
- Se tiene "***n*"** trabajos independientes, cada uno con una serie de procesos que requieren una relación de precedencia.
- La descripción de cada proceso se conoce con anticipación.
- Los tiempos de ajuste de las maquinas son independientes de la secuencia de los procesos y se incluyen en los tiempos de proceso.
- Una vez iniciado el proceso, este solo se interrumpe hasta su conclusión.

Sea[k] = q esto indica el K-esimo trabajo se encuentra en el q-esimo lugar para ser procesado.

Ej. (Chinneck, 2010) =1 significa que el trabajo 3 se hará de primero. (Boyd, 2003) (Corominas, 2001)

Fundamentos del Algoritmo de Lomnicki

El Procedimiento branch and bound fue propuesto simultáneamente por Lomnicki (1965) y por Ignall y Schrage (1965).

La exploración inteligente del conjunto de permutaciones de "***n***" piezas; se distribuyen las permutaciones en paquetes, cada uno definido por las piezas concretas que ocupan

determinadas posiciones (en la versión original, las primeras posiciones).
Para cada paquete se determina una cota inferior del valor de **Fmax** de las permutaciones incluidas en el mismo.
La cota es más ajustada cuantas menos permutaciones contiene el paquete; es decir cuando contiene una permutación, coincide con el valor **Fmax** de la misma.
Al ser las cotas de diferentes paquetes, la exploración de algunos es más prometedora. (Aleman, 2002) (Companys, 2005) (Färber, 2006)
En caso de disponer de una solución del problema heurística o del procedimiento ciertos paquetes pueden eliminarse de cualquier exploración posterior si:

- Cota>/Fmax (solución disponible)

Un paquete se explora al dividirlo en sub-paquetes y calculando sus cotas.

Algoritmo:

1) Definición del contenido de cada paquete
2) Procedimiento de acotación de un paquete
3) Procedimiento de división de un paquete en sub-paquetes
4) Estrategia de exploración

- El conjunto de paquetes se estructura en forma de árbol.
- Los vértices del árbol son los paquetes (se utiliza preferentemente "vértice").
- La raíz del árbol es un paquete que contiene todas las n! permutaciones (TODOS).
- Los rcos del árbol corresponden a la relación de inclusión (teoría de conjuntos).
- El árbol se va construyendo en el transcurso del algoritmo.
- Cuándo un paquete se subdivide, el vértice "padre" queda sustituido por vértices "hijos" (que contienen todas sus permutaciones), con cota no inferior a la del padre.

Durante la aplicación del algoritmo solo es preciso conservar vértices pendientes (todavía no ramificados o subdivididos en sus "hijos" y cuya cota sea inferior a Fmax de la mejor solución conocida). (Pinedo, 2008) (Aleman, 2002) (Goic, 2006)

1) Definición del contenido de cada paquete. Cada vértice "***v***" queda definido por las piezas situadas en primeras posiciones de la permutación y por su orden. Sea I= conjunto de los índices de las piezas fijadas, r = número de elementos de I (r = [I], π[l]= permutación de las mismas y E=conjunto de piezas no fijadas: Pueden existir varios vértices con igual I pero diferentes permutaciones (hasta r!). El vértice contiene todas las (n-r)! permutaciones comenzadas por π (I).

2) Procedimiento de acotación de un paquete. Dado un vértice "*v*", se determina una cota de Fmax de las permutaciones que contiene (cotas vistas). Sea fj(v) el instante en que queda liberada la máquina *j* realizando las operaciones de las piezas de I en el orden definido por *π (I)*.Para valores *1<=k<=*, se puede determinar las cotas: Si j,k(E) es una cota de Fmax del sub-problema *n-r/k-j+1/p/Fmax* deducido del original considerando únicamente *k- j* + 1 máquinas comprendidas entre j y k , y n-r piezas del conjunto *E*. La cota definitiva será la más ajustada: *K(v) = max {Kj,k(v)}*.
Si el vértice es terminal, su cota corresponde a *Fmax* de la permutación que contiene. Si dicho valor es mejor que la mejor solución conocida, se guardan la permutación y el valor como nueva mejor solución conocida. En todos los casos, se cierra o elimina el vértice terminal.

3) Procedimiento de división de un paquete en sub-paquetes. Para un vértice "*v*" no terminal, se definen *n-r* vértices hijos situando sucesivamente una de las piezas del

conjunto *E* detrás de la permutación *π (I)*, formando una nueva permutación con *r +1* piezas. Generados y acotados los *n-r* hijos se cierra el padre.

Estrategia de exploración. Elección del vértice a explorar o ramificar. Propuesta del algoritmo original: dar prioridad al vértice pendiente con menor cota, y en caso de empate, con mayor número de piezas fijadas. Cuando n adopta valores elevados (>8), para conservar un numero de vértices dentro de unos límites conviene utilizar permutar los criterios (elegir el vértice con mayor número de piezas, y en caso de empate, dar prioridad a la menor cota). (Aleman, 2002) (Companys, 2005) (Goic, 2006)

Descripción del Algoritmo de Branch and Bound o de Ignall-Schrage y Lomnicki (N/3/F/Fmax)

Este tipo de metodología describe la elaboración de un árbol de ramificaciones donde cada nodo o ramal representa una solución parcial. El primer nodo en la estructura del árbol corresponde al estado inicial sin programar las tareas existentes. De este nodo, hay ***n*** ramificaciones correspondientes a ***n*** posibles trabajos que pueden ser colocados, primeros en la secuencia. De estos nodos hay ***n-1*** ramificaciones correspondientes a los trabajos restantes para ser colocados segundos en la secuencia, como existe **n!** posibles secuencia habrá *1 + n + n (n-1) + --- + n!* nodos en el árbol; y cada nodo representará una secuencia parcial que contienen de 1 a **n** trabajos (Boyd, 2003) (Corominas, 2001).

Considérese un nodo P, con una secuencia J_R, es decir, J_R es un subconjunto de tamaño $r\left(1 \le r \le n\right)$ de los n trabajos. Sean T1 (J_R), T2 (J_R) y T3 (J_R) los tiempos en los que se completan los trabajos de la secuencia J_R en las máquina 1, 2 y 3 respectivamente. Entonces una cuota inferior del tiempo total de procesamiento de la secuencia J_R en el nodo P, denotada CI (P), se calcula de (Romero, 2012):

$$C1\left(P\right) = CI\left(J_R\right) = \max = \begin{cases} T1\left(J_R\right) + \sum_{i \in J_R}^{n} t_{i1} + \min_{i \in J_R}\left(t_{i2} + t_{i3}\right) \\ T2\left(J_R\right) + \sum_{i \in J_R}^{n} t_{i2} + \min_{i \in J_R}\left(t_{i3}\right) \\ T3\left(J_R\right) + \sum_{i \in J_R}^{n} t_{i3} \end{cases} \quad (1)$$

METODOLOGÍA

Una metodología es un conjunto de acciones que se elaboran a partir de un constructo epistemológico emanado por uno o varios modelos, los cuales a su vez requieren de procedimientos, técnicas, herramientas y soporte documental que indican el derrotero a seguir para el desarrollo de soluciones a problemas específicos (Charvat, 2003), la cual deberá poseer características y criterios de cientificidad (Palacio, 2012). El diseño de la metodología propuesta en el presente trabajo está inmersa en los protocolos del estudio de caso (Yin, 2008) universalmente aceptados, que basa su visión decisional en los principios rectores

de la escuela de Taniguchi (Taniguchi, 2001), que sugieren un procedimiento sistemático para diagnosticar y proponer soluciones inteligentes a las problemáticas que coexisten en las redes de suministro dirigidas por la demanda (Mendes, 2011) locales, que consta de los siguientes pilares fundamentales:

Parte de, **analizar el problema** con respecto a los datos de la producción en firme (Sipper, 1998) (Silver, 1985; Elsayed, 1994), que mitiguen el efecto látigo del inventario de producción entre los eslabones proveedor-comprador (Mejía, 2013).

En segundo lugar, **determinar un método de análisis**, que permita establecer la estrategia, política y el sistema de producción (Sipper, 1998) (Elsayed, 1994), con el propósito de realizar lo planificado en el paso anterior.

Luego, **generar la base de datos**, fundamental para cuantificar el tamaño de lote de producción multi-producto (Silver, 1985) (Tersine, 1994) acorde al diseño espacial del área de operaciones (Hernández, 1986) orientado hacia la programación de operaciones flow shop (Pinedo, 2008) mediante el uso del método Branch and Bound (Aleman, 2002) (Lomnicki, 1965), en función del agente que posee el poder en la red de suministro de manufactura (Palacio, 2012).

En cuarto lugar, **se busca fijar las alternativas** que puedan contribuir a solucionar el problema de programación de operaciones planteado (French, 1982), (Sipper, 1998) (Elsayed, 1994), en términos del equilibrado entre los tiempos de ciclo y del takt time para los eslabones proveedor-comprador del servicio de manufactura de tipo taller para ambientes mypime (Aleman, 2002) (Palmer, 1965) (Boyd, 2003).

Luego, se recurre al **planteamiento de modelos** que combine la investigación de operaciones, la estadística y la administración de proyectos ágiles (Palacio, 2012; BLÉ, 2010) que ayuden a seleccionar el programa de producción que brinde el mejor makespan para el taller de manufactura flow shop (Pinedo, 2008) (Elsayed, 1994). Los resultados obtenidos a partir del modelado son posteriormente sometidos a una **evaluación**, en la que se contrastan con el tiempo de respuesta pactados con el área de operaciones (Sipper, 1998) (Elsayed, 1994)(Tersine, 1994), y en la que se involucren no sólo los aspectos cuantitativos (Díaz - Pérez, 2012), sino también cualitativos (Geilfus, 2002). El resultado de esta evaluación, robustecido a través de análisis de sensibilidad (Palacio, 2012), permitirá al tomador de decisiones **negociar** el programa de producción que posea el mejor makespan (Sipper, 1998) (Elsayed, 1994).

Seguidamente, se procedería a la **ejecución** del programa de producción pactado entre los eslabones proveedor-comprador (Palacio, 2012), con el compromiso de efectuar una **revisión** periódica a los resultados obtenidos a través del método Branch and Bound (Aleman, 2002) (Lomnicki, 1965), con el objeto de tomar acciones correctivas y preventivas frente a futuros acuerdos empresariales como tal (Palacio, 2012), en caso de que estos no fueran los deseados por parte de los agentes intervinientes (Vollmann, 2005). Estos dos últimos pasos, ejecución y revisión de resultados, quedarán fuera del ámbito de este trabajo, que sólo busca recomendar soluciones viables desde el punto de vista técnico.

El estudio de caso como herramienta de investigación empresarial (Yin, 2008) permea este modelo metodológico, en lo referente a: i) Análisis de la capacidad

disponible del taller (Vollmann, 2005); ii) Procesos de optimización colaborativa (Palacio, 2012); iii) Determinación del tamaño de lote de producción (Elsayed, 1994)(Tersine, 1994); y iv) Fabricación colaborativa (Palacio, 2012), que dé respuesta a una pregunta del tipo **por qué o cómo** tratan el concepto de causalidad etnográfica donde se privilegien las limitaciones de tiempo y espacio en los que se enmarca los estudios de cuantificación del makespan eficiente (Pinedo, 2008) (Elsayed, 1994), que permite gestionar el tamaño del lote de producción (Silver, 1985) (Tersine, 1994) abordado desde la casuística que en forma agregada impacte los procesos de negociación entre los eslabones proveedor-comprador (Palacio, 2012), y que a su vez, permite la construcción del modelo de gestión del conocimiento (Gómez, 2009) vía modelos metodológicos de trabajo colaborativo para proyectos (Charvat, 2003). Esta situación motivo el desarrollo de este trabajo buscando mejorar la visión decisional de las actuales metodologías de trabajo para la secuenciación de operaciones de manufactura (French, 1982), (Pinedo, 2008) (Elsayed, 1994) implementado la integración de modelos matemáticos preexistentes en el medio empresarial relacionados con el método Branch and Bound para ambientes flow shop (Dannenbring, 1977) (Nawaz, 1983) (Companys, 2005) y gestionando el proceso de negociación entre los eslabones proveedor-comprador administrada como un proyecto ágil (BLÉ, 2010).

ANALISIS DE LOS RESULTADOS

La presente propuesta metodológica ha sido comprobada al interior de una empresa dedicada a la Metalmecánica, dicha compañía está ubicada en la ciudad de Bogotá, y el objetivo general pretendido fue el de determinar el óptimo para cinco (5) tareas en tres (3) máquinas identificando a través de técnicas de poda la estrategia de ramificación optimizada, es decir, a través del algoritmo branch and bound se pretende generar un enfoque eficaz para la elección de subconjuntos con características óptimas (Narayiyakul, 2014).

Pasos básicos para la ejecución de un modelo con branch and bound

- El procedimiento consiste primero, en generar nodos en un árbol y calcular los límites inferiores asociados a la secuencia Sr.
- Luego, se ramifica desde el nodo que tenga el mínimo límite inferior.
- Posteriormente, se calcula el límite inferior con la ecuación (1) y tan pronto un nodo tenga en su secuencia los **n** trabajos, se ha llegado finalmente a la solución óptima. (Bautista, 2008) (Iñigo, 2011)

Aplicación de Metodología para la Industria Metalmecánica (Empresa Darwis Colombia Limitada)

Se toma como ejemplo para el cálculo los datos consignados en la tabla número 2 estos que pueden ser modificados dependiendo de las necesidades de cada empresa, tanto de sus tareas o labores como del número de máquinas existentes en el taller, esto responde a la generación de la base de datos indicada en la metodología. Para este caso son las siguientes actividades a realizar:

Tabla No.1. *Actividades a realizar por máquina*

TAREA	Máquina1	Máquina2	Máquina3
1. Limado	1	3	2
2. Tornado	2	5	3
3.Fresado	3	1	1
4.Cromado	2	7	6
5.Zincado	3	4	1

Una vez se poseen los datos de las operaciones y máquinas que intervienen se procede a generar la sumatoria de tiempos por tarea en las tres máquinas, como se muestra en la tabla N° 2. Esto con el fin de determinar aquellas alternativas que puedan contribuir a la resolución del problema de programación planteado.

Tabla No. 2. *Ramificación de nodos*

TIME M1 (1) =	1	TIME M1 (2) =	2	TIME M1 (3) =	3	TIME M1 (4) =	2	TIME M1 (5) =	3
TIME M2 (1) =	4	TIME M2 (2) =	7	TIME M2 (3) =	4	TIME M2 (4) =	9	TIME M2 (5) =	7
TIME M3 (1) =	6	TIME M3 (2) =	10	TIME M3 (3) =	5	TIME M3 (4) =	15	TIME M3 (5) =	8

-Se determinan los limites inferiores de los primeros 5 nodos.

Tabla No.3. *Determinación límites inferiores de nodos.*

	2	9	2	=	13		TAREA	Máquina1	Máquina2	Máquina3					
LI(1)=MAX	9	13	1	=	23	23	1	1	3	2	5				
	15	7		=	22		2	2	5	3	8				
							3	3	1	1	2				
							4	2	7	6	13				
							5	3	4	1	5				
	3	8	2	=	13		TAREA	Máquina1	Máquina2	Máquina3					
LI(1)=MAX	7	16	1	=	24	24	1	1	3	2	5				
	8	12		=	20		2	2	5	3	8				
							3	3	1	1	2				
							4	2	7	6	13				
							5	3	4	1	5				
	1	10	2	=	13		TAREA	Máquina1	Máquina2	Máquina3					
LI(1)=MAX	4	17	1	=	22	22	1	1	3	2	5				
	6	11		=	17		2	2	5	3	8				
							3	3	1	1	2				
							4	2	7	6	13				
							5	3	4	1	5				
	2	9	2	=	13		TAREA	Máquina1	Máquina2	Máquina3					
LI(1)=MAX	7	15	1	=	23	23	1	1	3	2	5				
	10	10		=	20		2	2	5	3	8				
							3	3	1	1	2				
							4	2	7	6	13				
							5	3	4	1	5				
	3	8	5	=	16		TAREA	Máquina1	Máquina2	Máquina3					
LI(1)=MAX	4	19	1	=	24	24	1	1	3	2	5				
	5	12		=	17		2	2	5	3	8				
							3	3	1	1	2				
							4	2	7	6	13				
							5	3	4	1	5				

-El nodo 1 tiene el límite inferior por esto el proceso se divide en (12), (13), (14),(15) y para hallar su límite inferior se necesita calcular TIME (12), TIME (13), TIME (14), TIME (15).

Tabla No.4. *Cálculo de límites inferiores con time 12, 13, 14 y 15*

TIME 1(12) = TIME M1 (1) + P21 =					3	TAREA	Máquina1	Máquina2	Máquina3		
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]					9	1	1	3	2		5
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]					12	2	2	5	3		8
						3	3	1	1		2
						4	2	7	6		13
	3	8	2	=	13	5	3	4	1		5
L1(12)= MAX	9	12	1	=	22	22					
	12	8		=	20		TIME M1(1)=	1			
							TIME M2(1)=	4			
							TIME M3(1)=	6			

TIME 1(13) = TIME M1 (1) + P31 =					4	TAREA	Máquina1	Máquina2	Máquina3		
TIME 2(13) = max[TIME 1(13)+ P32 ----o------TIME M2(1)+P32]					5	1	1	3	2		5
TIME 3(13) = max[TIME 2(13)+ P33 ----o------TIME M3(1)+P33]					7	2	2	5	3		8
						3	3	1	1		2
						4	2	7	6		13
	4	7	5	=	16	5	3	4	1		5
L1(12)= MAX	5	16	1	=	22	22					
	7	10		=	17		TIME M1(1)=	1			
							TIME M2(1)=	4			
							TIME M3(1)=	6			

-Los cuatro nodos tienen el limite igual por esto el proceso se divide primero en {(123), (124), (125)}.

Tabla No.5. División entre 123, 124 y 125

TIME 1(123) = TIME M1 (12) + P31 =					6	TIME 1(12) = TIME M1 (1) + P21 =					3
TIME 2(123) = max[TIME 1(123)+ P32 ----o------TIME M2(12)+P32]					10	TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22					9
TIME 3(123) = max[TIME 2(123)+ P33 ----o------TIME M3(12)+P33]					13	TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23					12
	6	5	5	=	16						
L1(123)= MAX	10	11	1	=	22	22	TAREA	Máquina1	Máquina2	Máquina3	
	13	7		=	20		1	1	3	2	5
							2	2	5	3	8
							3	3	1	1	2
							4	2	7	6	13
							5	3	4	1	5

TIME 1(124) = TIME M1 (12) + P41 =					5	TIME 1(12) = TIME M1 (1) + P21 =					3
TIME 2(124) = max[TIME 1(124)+ P42 ----o------TIME M2(12)+P42					16	TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]					9
TIME 3(124) = max[TIME 2(124)+ P43 ----o------TIME M3(12)+P43					22	TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]					12
	5	6	2	=	13						
L1(124)= MAX	16	5	1	=	22	24	TAREA	Máquina1	Máquina2	Máquina3	
	22	2		=	24		1	1	3	2	5
							2	2	5	3	8
							3	3	1	1	2
							4	2	7	6	13
							5	3	4	1	5

TIME 1(125) = TIME M1 (12) + P51 =					6	TIME 1(12) = TIME M1 (1) + P21 =					3
TIME 2(125) = max[TIME 1(125)+ P52 ----o------TIME M2(12)+P52]					13	TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]					9
TIME 3(125) = max[TIME 2(125)+ P53 ----o------TIME M3(12)+P53]					14	TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]					12
	6	5	2	=	13						
L1(125)= MAX	13	8	1	=	22	22	TAREA	Máquina1	Máquina2	Máquina3	
	14	7		=	21		1	1	3	2	5
							2	2	5	3	8
							3	3	1	1	2
							4	2	7	6	13
							5	3	4	1	5

-El proceso se divide ahora en {(132), (134), (135)}.

Tabla No.6. *División entre 132, 134 y 135*

TIME 1(132) = TIME M1 (13) + P21 =		6
TIME 2(132) = max[TIME 1(132)+ P22 ----o------TIME M2(13)+P22		11
TIME 3(132) = max[TIME 2(132)+ P23 ----o------TIME M3(13)+P23		14

	6	5	5	=	16
L1(132)= MAX	11	11	1	=	23
	14	7		=	21

23

TIME 1(13) = TIME M1 (1) + P31 =		4
TIME 2(13) = max[TIME 1(13)+ P32 ----o------TIME M2(1)+P32		5
TIME 3(13) = max[TIME 2(13)+ P33 ----o------TIME M3(1)+P33		7

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

TIME 1(134) = TIME M1 (13) + P41 =		6
TIME 2(134) = max[TIME 1(134)+ P42 ----o------TIME M2(13)+P42		13
TIME 3(134) = max[TIME 2(134)+ P43 ----o------TIME M3(13)+P43		19

	6	5	5	=	16
L1(134)= MAX	13	9	1	=	23
	19	4		=	23

23

TIME 1(12) = TIME M1 (1) + P21 =		4
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22		5
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23		7

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

TIME 1(135) = TIME M1 (13) + P51 =		7
TIME 2(135) = max[TIME 1(135)+ P52 ----o------TIME M2(13)+P52		11
TIME 3(135) = max[TIME 2(135)+ P53 ----o------TIME M3(13)+P53		12

	7	4	8	=	19
L1(125)= MAX	11	12	3	=	26
	12	9		=	21

26

TIME 1(12) = TIME M1 (1) + P21 =		4
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22		5
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23		7

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

-El proceso se divide después en {(142), (143), (145)}

Tabla No.7. *División entre 142, 143 y 145*

TIME 1(142) = TIME M1 (14) + P21 =		5
TIME 2(142) = max[TIME 1(142)+ P22 ----o------TIME M2(14)+P22		16
TIME 3(142) = max[TIME 2(142)+ P23 ----o------TIME M3(14)+P23		20

	5	6	2	=	13
L1(142)= MAX	16	5	1	=	22
	20	2		=	22

22

TIME 1(14) = TIME M1 (1) + P41 =		3
TIME 2(14) = max[TIME 1(14)+ P42 ----o------TIME M2(1)+P4		11
TIME 3(14) = max[TIME 2(14)+ P43 ----o------TIME M3(1)+P4		17

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

TIME 1(143) = TIME M1 (14) + P31 =		6
TIME 2(143) = max[TIME 1(143)+ P32 ----o------TIME M2(14)+P32]		12
TIME 3(143) = max[TIME 2(143)+ P33 ----o------TIME M3(14)+P33]		18

	6	5	5	=	16
L1(143)= MAX	12	9	1	=	22
	18	4		=	22

22

TIME 1(12) = TIME M1 (1) + P21 =		3
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]		11
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]		17

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

TIME 1(145) = TIME M1 (14) + P51 =		6
TIME 2(145) = max[TIME 1(145)+ P52 ----o------TIME M2(14)+P52]		15
TIME 3(145) = max[TIME 2(145)+ P53 ----o------TIME M3(14)+P53]		18

	6	5	2	=	13
L1(145)= MAX	15	6	1	=	22
	18	4		=	22

22

TIME 1(12) = TIME M1 (1) + P21 =		3
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]		11
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]		17

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

-El proceso se divide finalmente en {(152), (153), (154)}.

Tabla No.8. *División entre 152, 153 y 154*

TIME 1(152) = TIME M1 (14) + P21 =		6
TIME 2(152) = max[TIME 1(142)+ P22 ----o------TIME M2(14)+P22]		13
TIME 3(152) = max[TIME 2(142)+ P23 ----o------TIME M3(14)+P23]		16

	6	5	2	=	13
L1(142)= MAX	13	8	1	=	22
	16	7		=	23

23

TIME 1(15) = TIME M1 (1) + P51 =		4
TIME 2(15) = max[TIME 1(15)+ P52 ----o------TIME M2(1)+P52]		8
TIME 3(15) = max[TIME 2(15)+ P53 ----o------TIME M3(1)+P53]		9

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

TIME 1(153) = TIME M1 (14) + P31 =		7
TIME 2(153) = max[TIME 1(143)+ P32 ----o------TIME M2(14)+P32]		9
TIME 3(153) = max[TIME 2(143)+ P33 ----o------TIME M3(14)+P33]		10

	7	4	8	=	19
L1(143)= MAX	9	12	3	=	24
	10	9		=	19

24

TIME 1(12) = TIME M1 (1) + P21 =		4
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]		8
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]		9

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

TIME 1(154) = TIME M1 (14) + P51 =		6
TIME 2(154) = max[TIME 1(145)+ P52 ----o------TIME M2(14)+P52]		15
TIME 3(154) = max[TIME 2(145)+ P53 ----o------TIME M3(14)+P53]		21

	6	5	2	=	13
L1(145)= MAX	15	6	1	=	22
	21	4		=	25

25

TIME 1(12) = TIME M1 (1) + P21 =		4
TIME 2(12) = max[TIME 1(12)+ P22 ----o------TIME M2(1)+P22]		8
TIME 3(12) = max[TIME 2(12)+ P23 ----o------TIME M3(1)+P23]		9

TAREA	Máquina1	Máquina2	Máquina3	
1	1	3	2	5
2	2	5	3	8
3	3	1	1	2
4	2	7	6	13
5	3	4	1	5

-Los nodos {(123) , (125)} , {(142) , (143) , (145)} tienen el límite inferior por esto el proceso se divide en { (1234), (1235), (1253), (1254) } {(1423) , (1425) , (1432) , (1435), (1452) ,(1453)} y para hallar su límite inferior se procede a manera de ejemplo de la siguiente manera:

Tabla No.9. *Determinación de límites inferiores*

TIME 1(1234) = TIME M1 (123) + P41 =						8				TIME 1(123) = TIME M1 (12) + P31 =						6			
TIME 2(1234) = max[TIME 1(1234)+ P42 ----o------TIME M2(123)+P						17				TIME 2(123) = max[TIME 1(123)+ P32 ----o------TIME M2(12)+P						10			
TIME 3(1234) = max[TIME 2(1234)+ P43 ----o------TIME M3(123)+P						23				TIME 3(123) = max[TIME 2(123)+ P33 ----o------TIME M3(12)+P						13			
	8	3	5	=	16					TAREA	Máquina1	Máquina2	Máquina3						
L1(1234)= MAX	17	4	1	=	22	24				1	1	3	2			5			
	23	1		=	24					2	2	5	3			8			
										3	3	1	1			2			
										4	2	7	6			13			
										5	3	4	1			5			
TIME 1(1235) = TIME M1 (123) + P51 =						9				TIME 1(123) = TIME M1 (12) + P21 =						6			
TIME 2(1235) = max[TIME 1(1235)+ P52 ----o------TIME M2(123)+P						14				TIME 2(123) = max[TIME 1(123)+ P22 ----o------TIME M2(12)+P						10			
TIME 3(1235) = max[TIME 2(1235)+ P53 ----o------TIME M3(123)+P						15				TIME 3(123) = max[TIME 2(123)+ P23 ----o------TIME M3(12)+P						13			
	9	2	13	=	24					TAREA	Máquina1	Máquina2	Máquina3						
L1(1235)= MAX	14	7	6	=	27	27				1	1	3	2			5			
	15	6		=	21					2	2	5	3			8			
										3	3	1	1			2			
										4	2	7	6			13			
										5	3	4	1			5			

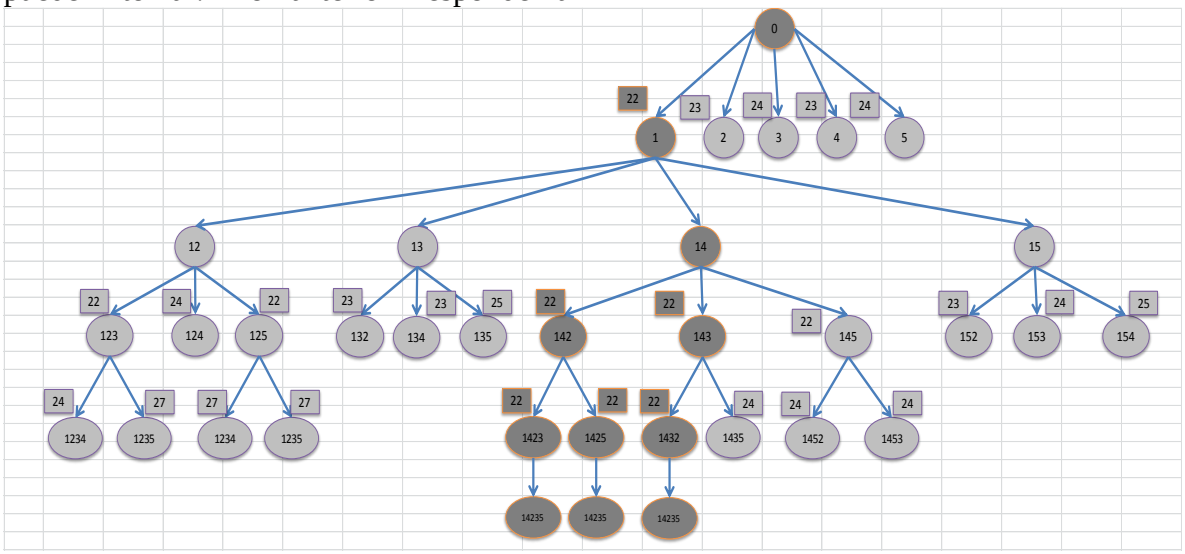
Tabla No.10. *Determinación de límites inferiores*

TIME 1(1425) = TIME M1 (142) + P51 =						8				TIME 1(142) = TIME M1 (14) + P21 =						5			
TIME 2(1425) = max[TIME 1(1425)+ P52 ----o------TIME M2(142)+P						20				TIME 2(142) = max[TIME 1(142)+ P22 ----o------TIME M2(14)+P						16			
TIME 3(1425) = max[TIME 2(1425)+ P53 ----o------TIME M3(142)+P						21				TIME 3(142) = max[TIME 2(142)+ P23 ----o------TIME M3(14)+P						20			
	8	3	2	=	13					TAREA	Máquina1	Máquina2	Máquina3						
L1(1425)= MAX	20	1	1	=	22	22				1	1	3	2			5			
	21	1		=	22					2	2	5	3			8			
										3	3	1	1			2			
										4	2	7	6			13			
										5	3	4	1			5			
TIME 1(1432) = TIME M1 (143) + P21 =						8				TIME 1(143) = TIME M1 (14) + P31 =						6			
TIME 2(1432) = max[TIME 1(1432)+ P22 ----o------TIME M2(143)+P						17				TIME 2(143) = max[TIME 1(143)+ P32 ----o------TIME M2(14)+P						12			
TIME 3(1432) = max[TIME 2(1432)+ P23 ----o------TIME M3(143)+P						21				TIME 3(143) = max[TIME 2(143)+ P33 ----o------TIME M3(14)+P						18			
	8	3	5	=	16					TAREA	Máquina1	Máquina2	Máquina3						
L1(1432)= MAX	17	4	1	=	22	22				1	1	3	2			5			
	21	1		=	22					2	2	5	3			8			
										3	3	1	1			2			
										4	2	7	6			13			
										5	3	4	1			5			

En este punto se puede observar que el mínimo límite inferior de todos los nodos es igual en {(1423), (1425), (1432)}.

L1(14235)= 22
L1(14253)= 22
L1(14325)= 22

El diagrama de árbol mostrado a continuación para cinco tareas y tres máquinas evidencia las rutas críticas que se pueden tomar. Lo anterior responde al planteamiento de aquella alternativa que genera el menor makespan dentro de las diferentes opciones analizadas.



Gráfica No.1. *Diagrama de árbol para cinco tareas y tres máquinas*

La gráfica anterior proporciona las rutas de posibles de secuencia, sin embargo aquella ruta demarcada como 0 – 1 – 14 – 142 o 143 – 1423 o 1425 o 1432 es la que representa la mejor solución, es decir, la solución optimizada de secuencia.

CONCLUSIONES

El método de Branch and Bound (en español Ramificación y Acotamiento) abordó la resolución de un modelo de programación entera a través de la propuesta de una secuencia de modelos de programación lineal que constituyeron los nodos o sub-problemas del caso presentado.

Es factible presentar alternativas de solución de problemas que se observan en el área de producción en las pequeñas y medianas empresas empleando técnicas y procedimientos de análisis que han sido desarrollados en las últimas dos décadas. Sin embargo y a pesar de la complejidad de los problemas de programación, debido a su explosión combinatoria y debido no es fácil llegar rápidamente a una solución los análisis y aplicación de modelos se tornan interesantes, por tal motivo se debe considerar adecuadamente un criterio de elección entre la simulación, los procedimientos heurísticos o la búsqueda de una solución óptima.

En los problemas de secuenciación son muchos los objetivos que se pueden alcanzar, por ejemplo minimizar el tiempo total transcurrido, maximizar el nivel de producción, minimizar el retraso total, minimizar el retraso máximo, minimizar el costo de producción, entre otros, no obstante desde el principio de cualquier tipo de investigación se debe definir con claridad el objetivo principal de este tipo

de problemas, pese a lo anterior, un objetivo a este nivel de programación por lo general depende de la disponibilidad de información y al interés por parte de los interesados para encontrar maneras de optimizar el desempeño al interior de la planta.

El método Branch and Bound posee diversos esquemas de desarrollo de investigación a nivel de toma de decisiones, en áreas como la economía, los diferentes subsistemas logísticos en la cadena de abastecimiento, administración, selección de tecnologías, entre otros, es decir, el método busca determinar la mejor decisión dentro de un esquema de posibles soluciones.

Referencias Bibliográficas

French, S., Sequiencing and scheduling. 1982, John Wiley.

Pinedo, M., (2008). Single Machine Models (Deterministic) En: Scheduling Theory, Algorithms, and Systems. Third edition, pp. 36-61.

Chinneck, J., (2010). Chapter 12: Integer/Discrete Programming via Branch and Bound En: Practical Optimization: a Gentle Introduction, pp. 1-10.

Aleman, A., (2002). Estudio y aplicación del algoritmo Lomnicki pendular al problema Fm /block/Fmax. Laboratorio de Organización Industrial, pp. 15-85.

Garey M.R., Johnson D.S., Sethi Ravi. The complexity of flowshop and jobshop scheduling. 1976. Mathematics of operations research.

Pascual, R., y Doll, M., (2005). Un algoritmo branch and bound doble para resolver el problema flowshop con bloqueos. IX congreso de Ingeniería de Organización. Gijón. Escuela Politécnica Superior de Ingeniería Industrial de Barcelona.

Lomnicki, Z.A. A "Branch and Bound" algorithm for the exact solution of the three machine-scheduling problem. 1965.

Ignal Edward. Schrage, Linus. Application of the Branch and Bound technique to some flow-shop scheduling problems. 1964.

Potts, C.N. An adaptive branching rule for the permutation flow shop problem. 1980. Elsevier.

Palmer, DS. Sequencing jobs through a multistage process in the minimum total time: a quick method of obtaining a near optimum. 1965.

Gupta, JND. A functional heuristic for the flow shop-scheduling problem. 1971.

Dannenbring, DG. An evaluation of flow shop sequence heuristic. 1977.

Nawaz, M. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. 1983.

Companys, R., Mateo, M., yAleman, A., (2005). Aplicación del Algoritmo Lompen a los Problemas fm/prmu/cmax y fm/block/cmax. Barcelona: Laboratorio de Organización Industrial, pp. 5-20.

Haouari, M. Ladhari, T. A branch and bound based local search method for the flow shop problem. 2004. Journal of operational research society.

Taniguchi, Eiichi, et al. City Logistics: Network modelling and intelligent transport systems. Amsterdam: Pergamon, 2001. 245p.

Yin, R.K. Case Study Research: Design and Methods. 4th Edition. California: SAGE Publications, 2008. 312p.

Little, J., (1966). Branch and Bound Methods for Combinatorial Problems. Paper prepared for presentation at the TAPPI Operations Research Symposium, Philadelphia, pp. 2-18.

Bautista, J., y Pereira, J., (2008). Transparencias Dirección de Operaciones. Cataluña: Universidad Politécnica de Cataluña, Departamento de Organización de Empresas.

Sipper, D., Bulfin, R., (1998) Planeación y Control de la Producción. México, pp. 13-35, 62-91.

Boyd, S., Ghosh, A., y Magnani, A., (2003). Branch and Bound Methods. Notes for EE392o, Standford University.

Corominas, A., y Companys, R., (2001). Procedimiento Generalizado de Branch and Bound.
Färber, G., (2006). Sequencing in Mixed Model Non-Permutation Flowshop Production Lines Using Constrained Buffers. Tesis Doctoral. Cataluña. Universidad Politécnica de Cataluña.
Goic, M., (2006). Análisis Post Optimal y Algorítmico de Ramificación y Acotamiento.
Romero Arenas Luis. Principios básicos de los sistemas de producción P. 222. 2012. Universidad Militar.
Charvat, J.J. (2003). Project management methodologies: Selecting, Implementing, and Supporting Methodologies and Processes for Projects. New York: Jhon Wiley & Sons, Inc, 264p.
Palacio-León O. (2012). "Propuesta metodológica para el diseño y operación de instalaciones de almacenamiento modulares ecoeficientes para productos no perecederos". Director: Wilson Adarme J. Universidad Nacional de Colombia-Sede Bogotá, Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial.
Mendes Jr., P. (2011). Demand driven supply chain: A structured and practical roadmap to increase profitability. Berlín: Springer. 185p.
Silver, E.A., and Peterson, R., (1985). Decision systems for inventory management and production planning. 2th edition, John Wiley and Sons, 722 p.
Elsayed A. E., and Thomas, O. Boucher. (1994). Analysis and control of production systems. 2nd Ediction. New York: Prentice Hall. 450p.
Mejía V., J.C, Palacio L., O, y Adarme J., W. (2013). "Efecto látigo en la planeación de la cadena de abastecimiento, medición y control". Bogotá D.C (Colombia). Revista Ciencia e Ingeniería Neogranadina, Volumen 23, Diciembre. Numero No 2. ISSN 0124-8170.
Tersine, Richard J. (1994). Principles of Inventory and Materials Management. New York: Prentice-Hall International. 591p.
Hernández P., G., y Woithe, G. (1986). "Fundamentos de la proyección de fábricas de construcción de maquinarias". Ciudad de la Habana (Cuba). Editorial Pueblo y Educación.
BLÉ J., C., et al. (2010). "Diseño Ágil con TDD". Islas Canarias (España). Publicado por www.iExpertos.com
Díaz-Batista, J. and Pérez-Armador, D. (2012). "Optimización de los niveles de inventario en una cadena de suministro," Ingeniería Industrial, vol. 33, (2), pp. 126–132.
Geilfus, F. (2002). "80 herramientas para el desarrollo participativo: Diagnóstico, planificación, monitoreo, evaluación". San José de Costa Rica (Costa Rica). Publicado por Instituto Interamericano de Cooperación para la Agricultura (IICA).
Vollmann, T.E., et al. (2005). "Planeación y control de la producción: Administración de la cadena de suministros". 5ª Edición. Ciudad de México D.F. (México). Editorial McGraw-Hill. 739p.
Gómez H., M. (2009). Memoria de la Tesis Doctoral: "Desarrollo de un modelo de evaluación de la gestión del conocimiento en empresas de manufactura". Madrid (España). Publicado por la UPM.
Narayiyakul, Songyot. (2014). A comparative study of suboptimal branch and bound algorithms. Elsevier. Pp. 1-2
Iñigo, B., Sánchez, H., y Urrutia, A. 2011, Ejercicios Resueltos de Investigación Operativa. Exámenes propuestos en la facultad de ciencias económicas y empresariales. Universidad del País vasco, pp 81-10.