UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

DIVISION DE ESTUDIOS DE POSGRADO
FACULTAD DE INGENIERIA
Departamento de Sistemas

LA TECNICA DE CALENDARIZACION EN EL BALANCEO DE LINEAS DE ENSAMBLADO

Junio 1991

Yolanda Laza Tiscareña

LA TECNICA DE CALENDARIZACION EN EL BALANCEO DE LINEAS DE ENSAMBLADO

RESUMEN

Esta tesis tiene como objetivo analizar problemas de calendarización que se presentan en un Sistema Productivo, plantear soluciones y proporcionar ejemplos de aplicación. El estudio está sustentado en el compendio de literatura relacionada con el tema y en planteamientos de programación lineal y teoría de Redes; para este fin se hizo una división en tres grandes grupos en los cuales inciden gran parte de los problemas de calendarización: *La calendarización de proyectos, la secuencia de actividades y el balanceo de líneas de ensamblado.*

El trabajo proporciona una reseña de técnicas de solución y ejemplos para los dos primeros grupos; en el tercero se profundiza a más detalle en el tema, describiendo algoritmos heurísticos de solución y utilizando el Método de Balas y un programa de computación con su aplicación.

Como resultado se obtiene un panorama de orden general en cuanto a Teoría de Calendarización y a más detalle la descripción y el análisis de las técnicas que resuelven los problemas de balanceo de ensamblado de línea.

INTRODUCCION

Una importante función de la administración de Sistemas Productivos es la Coordinación y Control de las actividades; incluyendo la localización óptima de los recursos en la ejecución de estas actividades; la Investigación de Operaciones apoya con sus técnicas a la solución de problemas relacionados con el control de las organizaciones. Ubicando a la Teoría de Calendarización dentro de este contexto vemos que presta atención tanto a la secuencia en la cual las actividades son ejecutadas como a la ordenación de las tareas en las estaciones de trabajo. La Teoría de Calendarización es una de las técnicas mas rápidamente desarrolladas dentro de las ramas de la optimización discreta y de la combinatoria aplicada.

Situándose en la Historia, los procedimientos de calendarización esquemáticos fueron desarrollados por Henri L. Gannt, durante la Primera Guerra Mundial, arrojando el diagrama de barras que lleva su nombre para posteriormente a finales de la década de 1940 evolucionar hacia otro método conocido como el método de Pilares que lleva ya implícito el concepto de red. El método PERT o Técnica de Evaluación o Revisión de Programa surge en la Marina de Estados Unidos a fines de los años cincuenta como un refinamiento de las técnicas anteriores con el fin de planear y calendarizar proyectos de gran escala. El método de CPM o de ruta critica surge de manera independiente con el mismo objetivo que el método PERT y fué puesto a la práctica en 1959. Las actividades fueron representadas en una red en 1962 por Fondahl.

En el caso de secuencia de actividades, Richard W. Conway de la Universidad de Cornell y Johnson han contribuido con serias investigaciones en este renglón.

En el área de Técnicas de Balanceo de Línea Buffa y Taubert propusieron dos métodos heurísticos en 1972. Sovereign resolvió un problema clásico de balanceo en 1974. Johnson y Montgomery describen en 1974 en detalle el Modelo de Programación Entera.

Las palabras calendarización y secuencia, en el contexto de aplicaciones de la Investigación de Operaciones, se han venido asociando con un amplio rango de problemas cuantitativos. Generalmente hablando, la secuencia se refiere a la ordenación de una colección de actividades o tareas que necesitan ser ejecutadas en una empresa; mientras que la calendarización se asocia con la asignación de puntos o tiempos para cada actividad que estipulan cuando deben ser ejecutados.

La Teoria de calendarización es una disciplina, a la cual no se le ha dado la atención que merece dada su importancia en la organización de los sistemas productivos, los trabajos que se han elaborado al respecto se ha hecho de manera aislada y el material publicado es encontrado en diferentes disciplinas. La Teoria de Calendarización es comparable con la Teoria de Inventarios, aunque los temas sobre inventarios han sido más desarrollados, ambas teorías contienen implicaciones y sugerencias que son de gran valor en la práctica industrial.

Entre otras causas los hábitos de consumo dentro de las sociedades en la últimas décadas; consecuencia de la propagación de los medios de comunicación, han provocado importantes modificaciones en el área industrial, donde los empresarios intentan cubrir necesidades al menor costo posible. La Teoría de Calendarización es un refinamiento en cuanto se refiere a técnica que optimiza el uso de los recursos; razón por la cual surge de manera paralela para apoyar en la solución de problemas de ordenación y secuenciación. La importancia que tiene este tópico en los procesos de los sistemas productivos fué la motivación para el presente estudio.

El propósito de este trabajo es analizar problemas de calendarización que se presentan con mayor frecuencia en un sistema productivo, para ello se hace una división en tres grandes grupos: la calendarización de proyectos, la secuencia de actividades y el balanceo de ensamblado de línea. Se plantean soluciones y se proporcionan ejemplos para cada caso, profundizando a más detalle en el tercer bloque, en el que se proporcion varios métodos heurísticos de solución y se resuelve un caso tipo, planteado como problema de programación lineal y resuelto con un programa de computación que utiliza el método de Balas.

El presente trabajo consta de cuatro etapas: En la primera se proporcionan los conceptos básicos, enseguida se proporciona la teoría, los planteamientos matemáticos y métodos de solución de la problemática ralacionada con calendarización de proyectos y secuencia de actividades , en la tercera sección que es la mas importante dentro de la tesis, se analiza la teoría referente al balanceo de líneas de ensamblado, después se estructura matemáticamente y se describen varios métodos de solución, clásicos y heurísticos, finalmente se proporciona un aplicación concreta que se resuelve mediante un programa de computación que utiliza el método de Balas, pasando posteriormente a la validación y análisis.

CAPITULO I

CONCEPTOS BASICOS

Un problema de planeación concreto se inicia con el estudio del sistema en el cual se encuentra situado, paso importante que ayuda a conocer la relación que existe entre los diversos elementos del sistema; enseguida se procede a conocer la estructura básica del problema, sus fronteras, sus variables internas y externas y las correlaciones que puedan existir entre las mismas, detectando los puntos coyunturales y conociendo cuales están bajo control ó influyen de manera determinante en el problema.

Se sitúa el proyecto dentro del sistema global al que pertenece, porque concibiéndolo como un subsistema puro seria un subsistema truncado; resulta ilógico abstraerlo del conjunto, se deberá conocer la conexión existente entre los diversos procesos de producción y entre éstos y el micro-entorno en el cual están situados; así como del análisis del maga-entorno en general, donde se encuentran sistemas productivos similares a los estudiados. El presente capítulo intenta dar respuesta a los anteriores cuestionamientos para el caso de sistemas productivos.

El capitulo se desarrolla como sigue: En la primera sección se describen aspectos generales; situando la Teoría de Calendarización en un Sistema Productivo, En la segunda etapa se dividen los problemas de Calendarización, según corresponde a sus características de aplicación. La última sección relaciona la terminologia de la Teoría de Calendarización con la Teoria de Redes.

1.1 *Aspectos Generales*

Al hablar de sistemas productivos se piensa en algo más complejo que la simple producción física de los bienes, se alcanza la amplia perspectiva de una organización que abarca diversas actividades. La Administración de proyectos involucra: La Planeación, La Organización, El Control y La Dirección del mismo. La Teoría de Calendarización; ubicada en dicho proceso cabe tanto en la organización de los mecanismos de producción, como en la planeación y control de los proyectos.

La Teoría de calendarización ó planeación de la producción es un subsistema ligado a la administración de los Sistemas Productivos; y trata con problemas cuyo propósito es encontrar un orden ó secuencia óptima en el proceso de ciertas tareas, (también llamadas actividades, trabajos u operaciones); utilizando para ello un conjunto de máquinas que realicen las tareas, y agregando una serie de restricciones que son impuestas sobre las mismas tareas, las máquinas ó las relaciones existentes entre ambas.

En general se está frente a un problema de calendarización, cuando se tiene la necesidad de asignar; con un cierto orden o siguiendo un itinerario, de las actividades con las máquinas; de tal manera que dicha asignación sea factible y que un cierto objetivo sea alcanzado ó al menos aproximado.

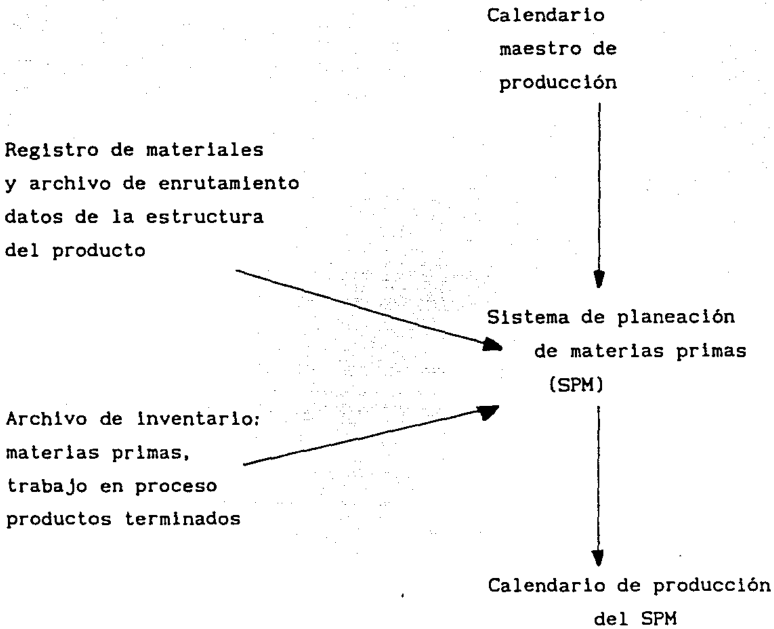
Existen usualmente un buen número de restricciones impuestas a los elementos que intervienen en un problema de calendarización y a las relaciones entre ellos. En particular las tareas pueden ser parcialmente ordenadas por restricciones tecnológicas y la maquinaria limitada por aspectos de capacidad. El objetivo es encontrar la relación en un espacio de tiempo, que optimice una medida de ejecución dada del sistema completo tareas-máquinas.

Concretamente un calendario se define como un orden asociado con las operaciones de la maquinaria de una empresa, ó con las actividades de la misma y un modelo de calendarización debe incluir relaciones de enrutamiento y otra serie de restricciones.

Al enfrentar un problema en el cual se puedan aplicar las técnicas de calendarización, es necesario plantear una serie de cuestionamientos sobre la realidad en la que están situados y que forman parte, para enseguida conceptualizarlo y después clasificarlo según la estructura de sus componentes. Un problema de calendarización en un principio se considera como tal con cierto criterio a priori, desmembrando los elementos que lo conforman, y detectando las variables de decisión que lo afectan. Esto es la base para la representación matemática en un modelo formal, (Se profundiza con más detalle en este aspecto en el capítulo tres); de los cuales existen diversos patrones elaborados y estudiados y su uso y modificaciones dependen de las características propias de cada problema.

En un Sistema de Planeación es necesario contar con una base de datos de materiales, un registro de inventario, un enrutamiento del producto y las partes manufacturadas con retraso, así como un calendario maestro. Puesto que las piezas de trabajo necesitan ser transportadas de un departamento a otro y seguido se invierte en tiempos de espera, contando con que los tiempos de producción son muy largos, consecuentemente, el control del flujo del producto se hace difícil, la utilización de recursos en el proceso del sistema es más complicada, por eso se hace necesario calendarizar la producción. La siguiente figura integra la calendarización en un sistema que tiene como insumos, materias primas.

Sistema de producción de materias primas



El análisis de un sistema comprende las siguientes etapas:

1) Determinación de las propiedades del sistema
2) Formulación del problema
3) Desarrollo de un modelo del sistema
4) Derivación de una solución

En la primera etapa se integra la mayor información posible sobre el problema que se desea resolver, se determina el medio ambiente en el cual se desenvuelve el sistema, se establecen los elementos, las leyes por medio de las cuales se rigen; en el segundo inciso se plantea con toda claridad el problema, se abstrae la información para constituir el modelo conceptual, identificando claramente las variables relevantes.

En el tercer inciso cuando el problema ha sido entendido y planteado, se toma una decisión sobre el tipo de modelo que se utilizará para el estudio, sea determinístico, estocástico, etc., conociendo de antemano que tipo de información se puede obtener de cada uno de ellos, enseguida se procede a constituir un modelo matemático, entendiendo por ello aquel en el cual el sistema está representaso por símbolos que lo describa y las reglas de decisión óptima serán derivadasdel mismo, es decir teniendo un planteamiento matemático en términos de las variables de decisión, se buscanlos valores de dichas variables que minimizan las función en cuestión, habiendo considerado un serie de restricciones para la solución.

Todas las restricciones del modelo deben ser expresadas como funciones matemáticas de las variables de decisión, que junto con la función objetivo, conforma un problema de optimización matemática que puede ser resuelto por métodos de cálculo, programación lineal, programación entera, programación dinámica, ó bién, algoritmos especiales.

Finalmente se aplica una técnica para obtener una respuesta a la inquietud que provoca el planteamiento del problema ó se reinicia la etapa de análisis y creación del modelo que de manera frecuente al estar manejado un proyecto se hace necesario volver a determinar las propiedades del sistema o reformular el mismo, que cada vez podrá ser más complejo dependiendo del número de variables de decisión que intervengan para representar de manera más científica y real y quizá menos subjetiva la problemática.

Cuando el propósito es tomar un criterio para implantar una política, el modelo matemático se puede manipular con reglas algebráicas para identificar como repercuten las variaciones de los costos en los otros elementos ó sea conocer la sensibilidad del modelo, para posteriormente aplicar las técnicas de simulación que representen el presente, confronten el pasado y pronostiquen el futuro y de esa manera manejar el comportamiento del sistema a través del tiempo. Las distintas validaciones son siempre permitidas ya que el modelo es una estructura, un simbolismo, un lenguaje metafísico.

1.2 *El problema de calendarización*

Los tópicos de calendarización estan relacionados de manera directa con los problemas de Administración de la producción y especificamente con problemas de planeación de la producción y pronóstico, además de teoria y control de inventarios. La Planeación es indispensable en los grandes proyectos, de ahí que la Teoria de calendarización funciona como una fuerte herramienta de apoyo para resolver problemas en la administración de una empresa, en la elaboración de un producto, en el ensamblado de un articulo, o en el seguimiento y control de un proyecto; utilizando para ello métodos de planeación de proyectos y técnicas de investigación de operaciones.

La planeación de grandes proyectos implica mayor esfuerzo debido a la complejidad generada por el gran volúmen de información y el desmembramiento de los elementos que intervienen, así como las relaciones que se dan entre los mismos. La Teoria de Calendarización, hace posible enmarcar el problema en un diseño que organiza lainformación, respetando las relaciones existentes entre las partes que intervienen y culminando para buscar soluciones en un modelo matemático o bién esquematizando el proyecto en una gráfica de red de actividades para posteriormente aplicar las técnicas propias a cada caso y resolver el problema de manera aproximada a la óptima.

Antes de la evolución de las técnicas sistemáticas de planeación, (que en general se clasifican como técnicas de redes), no habia garantía de que la utilización de recursos como el tiempo, fuerza de trabajo, diseño y equipo se lograran de una manera cercana a la óptima. Asimismo, la combinación de todos los subcomponentes, ó tareas del proyecto tendian a tratarse como entidades separadas en la etapa de planeación, en vez de hacerlo como un sistema total. La combinación de la Metodologia , el Enfoque de Sistemas, la computadora como herramienta, las técnicas de la diversas programaciones (Lineal, Entera, Dinámica, etc. y la Teoria de flujo en Redes, junto con algún criterio de optimalidad, permitió ciertamente instrumentar un plan mejor diseñado para alcanzar la culminación satisfactoria de un proyecto.

Es necesario señalar que dentro de la administración de los sistemas productivos una función importante es la planeación y control de las actividades complejas, la teoria de calendarización presta atención a ambas, a la secuencia del proceso de manufactura y a la programación de actividades en la maquinaria.

De acuerdo a las anteriores consideraciones, si el enigma a resolver es de ordenación, control, secuencia o balanceo, los problemas de calendarización se enmarcan de manera gruesa en tres grandes bloques: a) La calendarizacion de proyectos b) La secuencia de actividades y el c) balanceo de lineas de ensamblado. Existe otra clasificación en estáticos y dinámicos dependiendo de la información disponible, en cuanto a flujo del proceso, número y comportamiento de las actividades, certeza en los tiempos de manufactura, con subclasificiación en cada caso en deterministicos y estocásticos.

Enseguida se describen de manera somera los tres grandes bloques a los que se ha hecho referencia.

Tiene como objetivo fundamental desarrollar el programa de actividades de un proyecto, entendiendo como programa un diseño tanto conceptual como gráfico, de la sècuencia de tareas que deberán realizarse y que minimice el costo total esperado; para lograrlo se esquematiza el proyecto en una gráfica respetando la precedencia de las actividades y se busca su control; se distingue de la calendarización de tareas en un almacén y de otros tipos afines de calendarización por la naturaleza no repetitiva del trabajo que involucra. Los métodos de solución conocidos son el diagrama de Barras de Gannt, el Método de Ruta Critica y el Método PERT, Métodos de solución de modelos planteados con Programación Lineal y por último los Métodos de solución de Teoria de Redes.

En general, esta clase de proyectos es compleja y de gran escala, involucra un número grande de actividades componentes; ejemplos de proyectos tipicos se encuentran en el área de construcción: edificios, autopistas, barcos, las operaciones de mantenimiento periódicas, la administración, desarrollo de programas, facilidades en la instalación de una computadora y el desarrollo e implantación de un nuevo sistema de información, son actividades que pueden también ser tratadas con los métodos de proyectos de calendarización.

Secuencia de Actividades

El almacén de calendarización de tareas ó actividades consta de un conjunto de maquinaria de propósito general que ejecuta operaciones en las órdenes de producción ó tareas. El problema consiste en ordenar las operaciones que deben ser ejecutadas en cada máquina, sujeta a las restricciones y movilidad del almacén, tal que alguna función medible sea optimizada. Frecuentemente las tareas son únicas y resultan de una orden especifica del cliente. El diseño de un modelo de secuencia de actividades se puede plantear como un modelo que resulta fundamental para un número considerable de sistemas operacionales.

Entre los métodos que dan solución a estos problemas se pueden enumerar los siguientes: construcción de calendarios, algoritmos de enumeración exhaustiva, procedimientos de ramificación y acotamiento y los procedimientos heurísticos que son los mas funcionales entre ellos se menciona el algoritmo de Johnson como uno de los de mayor uso. Un ejemplo característico para este caso se tiene en cualquier fábrica X en la cual se requiera construir una secuencia óptima de actividades, asignadas a una ó más máquinas.

El problema de balanceo de ensamblado de línea consiste en la asignación de tareas individuales a las estaciones de trabajo, de tal modo que alguna medida apropiada de *ejecución de línea*, sea optimizada Las líneas de ensamblado están caracterizadas por el movimiento de las piezas del producto de una estación a la siguiente. Las tareas individuales requeridas para completar el artículo están divididas y asignadas a las estaciones de trabajo tal que cada estación ejecuta la misma operación en cualquier unidad del producto. Si una línea está balanceada perfectamente, entonces todas las estaciones tienen una cantidad igual de trabajo a ejecutar y el flujo contínuo y suave del producto será alcanzado. La línea de ensamblado es una línea de producción donde el material se mueve contínuamente con una tasa promedio uniforme a través de una secuencia de estaciones de actividades.

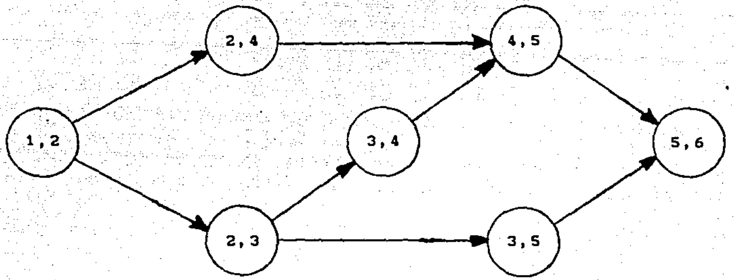
Los métodos de solución aplicables a estos casos son los característicos métodos que resuelven problemas planteados en un modelo de programación lineal entera, también los métodos que resuelven programación dinámica, algoritmos de secuencia de enumeración factible, procedimientos de búsqueda de árbol óptimo, personas que han trabajado en esta línea son Jackson (1956), Mansoor (1964), Nevins (1972) y Dar-El (1973), algoritmos de ruta más corta y por último técnicas heurísticas nombrando entre algunos los de Moodie and Young (1965), Arcus (1966), Nevins (1972) y Dar-El (1973). Ejemplos típicos de estas líneas de ensamblado son la manufactura de automóviles, el proceso de armado de las microcomputadoras, en general la manufactura de cualquier articulo que requiera las condiciones anteriores para su elaboración. Las lineas de ensamblado continuarán siendo la parte más importante de las empresas manufactureras y de las operaciones de ensamblado sin embargo esta labor puede ser reducida ó sustituida a través de la robotización, a un costo mucho mayor.

La Teoria de Calendarización y su parte de aplicación se ha situado dentro del contexto de un Sistema de producción; enseguida se hace necesario familiarizarce con el lenguaje mas frecuentemente utilizado en la programación de actividades, desmembrando su partes, de manera tal, que conforme se van describiendo los elementos se va construyendo una estructura que junto con la terminologia constituyen los medios de lenguaje que sirven para conceptualizar el problema en estudio. Por ello se presentan en el anexo A una serie de definiciones formales.

1.3 *Representación del Sistema en términos de Teoría de Redes*

Existen dos maneras básicas de asignar un proyecto a una red, ellas difieren en la representación de las actividades. En la primera llamada *actividad sobre el nodo* (AN ó red - actividad), una actividad está representada por un nodo y las restricciones de precedenciase hacen corresponde con los arcos. Por consiguiente, una red de actividad es una digráfica cuyos nodos están en correspondencia uno a uno con las actividades del proyecto y dos nodos son unidos por un arco si las actividades correspondientes tienen una relación de precedencia. La ventaja que tiene esta representación es que una digráfica puede modelar cualquier relación binaria definida sobre su conjunto de nodos, de esta manera es posible hacer el isomorfismo entre la red gráfica y el proyecto. Las propiedades del proyecto se formulan en términos de los elementos de la red. Este método gráfico no es muy ampliamente usado. la principal desventaja es que los cálculos algebráicos y matemáticos que son necesarios hacer a mano, resultan bastante dificultosos y para resolver el problema con el método de Ruta crítica, los programas de computación no lo aceptan fácilmente. La figura 1.3a muestra un ejemplo.

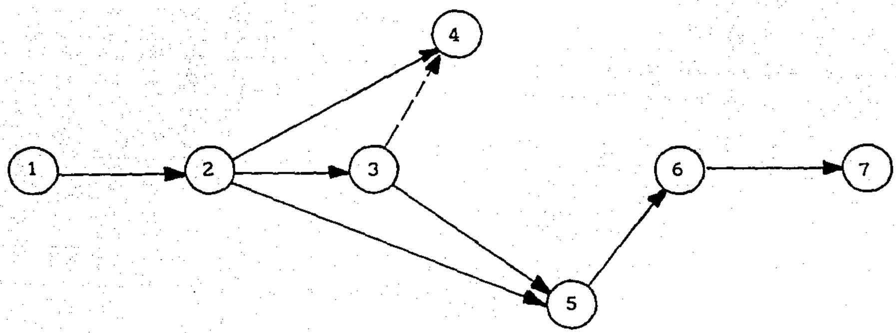
En otra representación de un proyecto, llamada *actividad sobre el arco* (AA ó red- evento); las actividades del proyecto son asignadas a los arcos y las restricciones de precedencia entre las actividades se preservan. Los nodos de tal representación son llamados eventos puesto que ellos corresponden a los momentos en el tiempo en los cuales las actividades empiezan y terminan. Aunque no es demasiado obvia la forma de construcción de una red-evento para un proyecto dado, esta representación ha sido aceptada grandemente en Análisis de Redes más que la representación de red-actividad.



Ejemplo de la representación de una red de actividades sobre el nodo

Figura 1.3a

Actividad	Duración	Predecesor Inmediato
(1,2)	4	
(2,4)	7	(1,2)
(2,3)	8	(1,2)
(2,5)	6	(1,2)
(4,6)	15	(2,4), (2,3)
(3,5)	9	(2,3)
(5,6)	12	(2,5), (3,5)
(6,7)	8	(4,6), (5,6)



Representación de una Red Actividad sobre el arco

Figura 1.3b

La flecha que forma el arco representa en la parte final en el nodo (evento), el inicio de la actividad y en la cabeza de la flecha el término en el nodo (evento). La descripción de una actividad está escrita a través del arco que representa esta actividad. Figura 1.3b

Las siguientes características deben ser tomadas en consideración cuando se construye la representación gráfica de una red de proyecto.

i) Los nodos (eventos) deberán numerarse de tal forma que el evento en donde inicia la flecha debe ser menor al evento en la cual termina.

ii) Cada nodo debe tener al menos un arco incidente y otro que sale de él, con excepción del primer nodo (origen ó fuente) y del último nodo (terminal o sumidero). Debe ser una red conexa.

iii) Cualesquiera dos eventos (nodos) deben ser conectados por al menos un arco. Sin embargo si dos actividades empiezan con el mismo evento (nodo), deberán tener su inicio en el mismo nodo, una restricción artificial ó arco raiz será usado para identificar una de las actividades.

Una *restricción artificial* es un arco sobre la red que muestra una relación de precedencia entre actividades y no requiere tiempo de consumo para satisfacer esa actividad. La inserción de restricciones artificiales prevé que dos ó mas actividades sean identificadas por el mismo conjunto de números. También las variables artificiales pueden ser usadas para eliminar una relación de dependencia que realmente no exista entre ciertasactividades. En resúmen cada actividad deberá estar representada por uno y solamente un arco en la red, además dos actividades diferentes no pueden identificarse por los mismos eventos terminal y de comienzo.

Un ejemplo de la situación anterior se muestra en la figura 1.3c, dos ó mas actividades deberán ejecutarse concurrentemente, donde las actividades A y B tienen los mismos eventos finales, se introduce la actividOd ficticia ya sea entre A y uno de los eventos finales, o entre B y uno de los eventos finales. Las representaciones modificadas, después de introducir la actividad ficticia D se muestra en la fig.1.3d

Como un resultado de usar D, las actividades A y B pueden ahora identificarse por eventos finales únicos. Nótese que una actividad ficticia no consume tiempo o recursos.

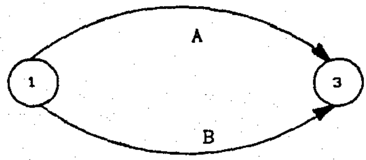


Figura 1.3c

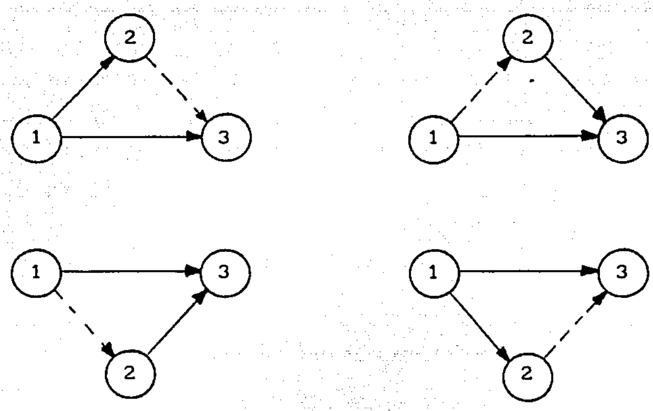


Figura 1.3d

v) A fin de asegurar la relación de precedencia correcta en el diagrama de flechas, las siguientes preguntas deben responderse cuando se agrega cada actividad a la red.

a) Qué actividades deben terminarse inmediatamente antes de que esta actividad pueda comenzar?

b) Qué actividades deben seguir a esta actividad?

c) Qué actividades deben efectuarse concurrentemente con esta actividad?

Errores más comunes en la construcción de redes

Se presentará en esta sección algunos ejemplos de los errores más comunes en la construcción de una red, así como el uso incorrecto de las varibles artificiales.

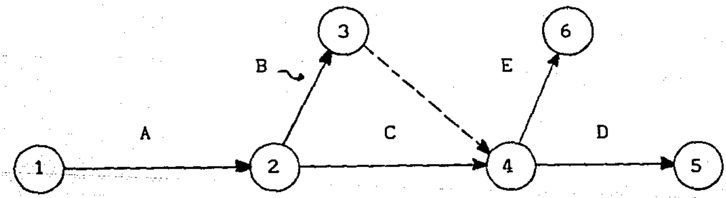
Ejemplo 1

Trazar un proyecto de red con las siguientes relaciones de precedencia

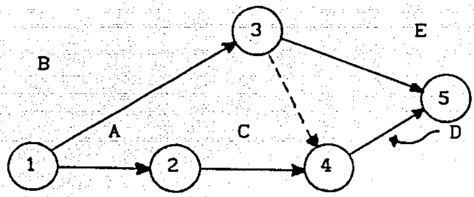
$$A < C \qquad B < D,E \qquad C < D$$

Solución:

Las figuras 1.3e y 1.3f representan una forma incorrecta y una correcta de la representación de la red.



Representación incorrecta del ejemplo 1.
A no precedería a B; C no precedería a E.
Figura 1.3e



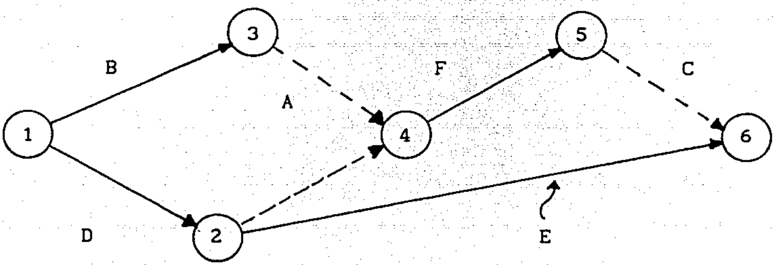
Presentación correcta del ejemplo 1

Figura 1.3f

Ejemplo 2

Trace un proyecto de red con las siguientes relaciones de precedencia

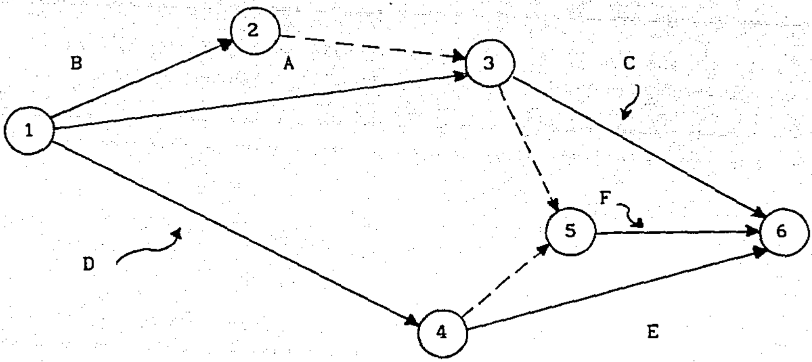
A < C,F   B < C,F   D < E,F



Representación incorrecta del ejemplo 2

D no precedería a C

Figura 1.3g



Correcta representación del ejemplo 2

Figura 1.3h

Comprobación de la consistencia de las relaciones de precedencia.

Después de construir la red del proyecto, es necesario verificar la consistencia de las relaciones de precedencia através de la red. La manera de hacer esto es desarrollando una *matriz de adyacencia*, la cual es una matriz $n \times n$, donde n es el número de nodos en la red. La matriz tiene como entradas el elemento +1, en el lugar ij, si existe una flecha del nodo i al j 0 en caso contrario. (Recuérdese que los nodos han sido numerados, tal que las flechas van del nodo numerado con el número más pequeño a más grande).

Las relaciones de precedencia del proyecto de red puede ser representada por grafos dirigidos ó por matrices boolianas. La forma implícita de inconsistencia ó contraindicaciones en la forma gráfica es un circuito cerrado o flecha que va del final al principio. La matriz de un conjunto de objetos en un conjunto en un circuito cerrado, contendrá un elemento en cualquier renglón y en cualquier columna; esto es no tendrá renglones ó columnas ceros. Puesto que la matriz de un subconjunto del conjunto original de objetos, será una submatriz principal de la matriz adyacente original, se concluye que un conjunto de relaciones de precedencia es consistente, sí y solo sí cualquier submatriz tiene al menos un renglón ó columna cero. El anterior método para corroborar la consistencia fué sugerido por Marimont. Enseguida se sintetizan los pasos para verificar la consistencia de una red.

i) Borrar cualquier renglón y columnas de ceros y la columna ó renglón que tengan el mismo número del renglón o columna que se borró; esto es, si el renglón (columna) i ha sido borrada porque todos sus elementos son ceros, es necesario borrar también la columna (renglón) i.

ii) Repetir el paso *i* hasta que se cumpla cualquiera de los dos siguientes incisos.

a) Cualquier linea que ha sido borrada en la matriz, indicaba una relación de precedencia consistente. ó

b) Una submatriz que no tenga lineas de ceros, indica un conjunto inconsistente

Enseguida se muestra un ejemplo que exhibe el uso de una matriz adyacente para verificar la consistencia de un proyecto de red.

Ejemplo 3

Construyase un proyecto de red con las siguientes relaciones de precedencia y verifique la consistencia de la red.

A < F	B < C,D,E
D < I	F < I
E < J	C < G,H
J < K	H < K
G < J	

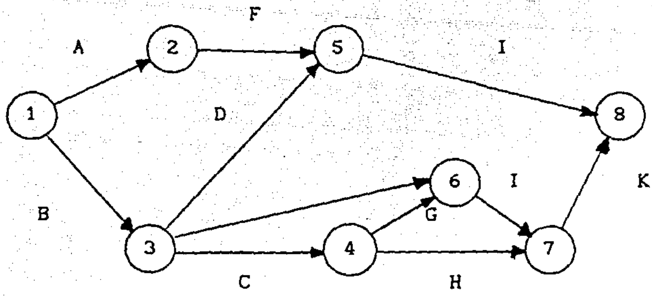


Figura 1.3d. Red para el ejemplo 3

Solución:

La figura 3.1d representa el proyecto de la red como se describió anteriormente. La matriz de adyacencia de la red es la siguiente.

	1*	2	3	4	5	6	7	8
1		1	1					
2					1			
3				1	1	1		
4						1	1	
5								1
6						1		
7								1
8*								

Borrar las columnas y renglones 1^* y 8^*.

$$
\begin{array}{c|cccccc|}
 & 2^* & 3^* & 4 & 5 & 6 & 7 \\
\hline
2 & & & & 1 & & \\
3 & & & 1 & 1 & 1 & \\
4 & & & & & 1 & 1 \\
5^* & & & & & & \\
6 & & & & & & 1 \\
7^* & & & & & & \\
\hline
\end{array}
$$

Borrar las columnas y renglones 2^*, 3^*, 5^* y 7^*.

$$
\begin{array}{c|cc|}
 & 4^* & 6 \\
\hline
4 & & 1 \\
6 & & \\
\hline
\end{array}
$$

Ahora cualquier línea en la matriz ha sido borrado, indicando la consistencia de la red.

ANEXO A

Definiciones de Términos

Proyecto

Sistema a gran escala que comprende un conjunto actividades interrelacionadas que se combinan y coordinan y que deben ejecutarse en un cierto orden para lograr el éxito del objetivo.

Tarea ó actividad

Es uno de los diversos trabajos autónomos que constituye el proyecto, es el esfuerzo en consumo de tiempo requerido para completar un segmento dentro de la red del proyecto. Su función es coordinar y programar el proyecto. Es necesario que todas las actividades inicien y terminen con un evento.

Recurso

Cualquier elemento necesario para propiciar el logro exitoso del proyecto. Los recursos típicos son el personal, equipo, dinero y tiempo entre otros. Si los recursos disponibles son ilimitados, en la satisfacción del proyecto no influyen los problemas de programación de recursos. En la programación y coordinación del proyecto se toma en consideración el caso de recursos limitados.

Evento

Un punto en el tiempo señalando el principio o completez de una o más actividades. Los puntos inicial y final de una actividad, por consiguiente, están descritos por dos eventos usualmente conocidos como evento de *comienzo* y evento *terminal*.

Flecha ó Arco

Representación gráfica de una tarea. Habrá tantas flechas como tareas haya en la representación gráfica de un proyecto. La punta indica el sentido de avance del proyecto. El arco deberá ser dirigido, pero la longitud no necesita ser proporcional a la duración de la actividad ni es necesario que se dibuje como una línea recta.

Nodo

Los puntos del tiempo entre los cuales se pueden programar las actividades. Los nodos se representan gráficamente como círculos, y se utilizan como conectores de flechas en la representación gráfica del proyecto general.

Red (Gráfica)

Es la representación gráfica del proyecto total, donde el proyecto de la red es un gráfica dirigida, la cual consiste de una colección finita N de n elementos i, j, ... llamados nodos, junto con un subconjunto A de parejas ordenadas (i,j) de nodos llamados arcos. Para trazar la red, se selecciona un punto que corresponda a cada nodo i de N y se dirige una flecha de i a j si la pareja ordenada está en A.

Actividad crítica

Se dice que una actividad es crítica si un retraso en su comienzo causa una demora en la fecha de terminación del proyecto completo.

Actividad no crítica

Es tal que el tiempo entre su comienzo más temprano y su terminación mas tardía (como se permite en el proyecto), es mas grande que su duración real.

Holgura

En términos generales la *holgura* de un evento, es el posible retraso que ese evento podría expeimentar, sin causar retraso alguno a la duración total del proyecto. Se definirán sobre una actividad otros tipos de holgura.

Holgura total. Presupone que el evento N_i se realiza lo más rápidamente posible, mientras que el N_j se retrasa lo más posible.

$$HT_{ij} = TT_j - IR_i - t_{ij}$$

Holgura se seguridad. Presupone que los eventos N_i y N_j se retrasan lo más posible.

$$HS_{ij} = TT_j - TT_i - t_{ij}$$

Holgura de libertad. Presupone que los eventos N_i y N_j se empiezan lo más rápidamente posible.

$$HL_{ij} = IR_j - IR_i - t_{ij}$$

Holgura de independencia. Mide la libertad absoluta en retrasar una actividad sin afectar a ninguna otra actividad.

$$HI_{ij} = \text{Máx}\ (0,\ IR_j - TT_i - t_{ij}\ )$$

Precedencia

Un término que describe la relación entre dos ó mas actividades en la red. Si una actividad A precede a otra actividad B, la actividad A debe estar finalizada antes de que la actividad B sea iniciada. Esta relación de precedencia es usualmente escrita como A < B. Esta definición formal es muy estricta, ya que sobre la marcha de un proyecto, se pueden trabajar actividades de manera simultánea, hasta donde sea posible.

Arco = Actividad

Evento $i \longrightarrow j$ Evento

La *precedencia* es una relación binaria que debe cumplir:

i) No reflexiva.

La actividad A no puede precederse a si misma

ii) No simétrica

Si la actividad A precede a la actividad B, entonces B no puede preceder a la actividad A

iii) Transitividad de la precedencia

Prácticamente en todos los proyectos del mundo real, las relaciones de precedencia corresponden a una relación transitiva; esto es, si $a<b$ y $b<c$, entonces $a<c$. Por simplicidad para este estudio se postula este resultado.

Proyecto consistente

Es cuando todos los tiempos de duración de las actividades son positivos, y no se tiene una secuencia de actividades cuyas restricciones de precedencia formen un ciclo.

Estructura del flujo

Para la elaboración de un producto alguna veces se hace necesario un trabajo simultáneo, entonces se tiene el caso paralelo; cuando las actividades se dan cronológicamente se tienen el caso de flujo en serie. El caso multicanal es una combinación de los dos anteriores.

CAPITULO-II

MODELOS DE CALENDARIZACION

El rápido cambio tecnológico, la disminución del margen de utilidades, un ritmo de vida más acelerado, se combinan para hacer la administración más difícil y exigente, demandando técnicas y métodos de administración y control más completos . Paralelamente en el área de la ciencia, se han dado grandes avances que permiten soslayar esos problemas. En el presente capítulo se describen algunas técnicas que sirven de apoyo para la solución de los anteriores problemas.

El capítulo se desarrolla como sigue: en la primera sección se analiza *la calendarización de proyectos*, con los métodos de solución de Diagramas de Gannt y Ruta Critica, en el segundo inciso se revisa la Teoría de la êTécnica de Evaluación y Revisión del Programa, en la última etapa se describe la *secuencia de actividades*, proporcionado planteamientos matemáticos para diversos casos, según varíe el número de máquinas o de tareas. Se proporcionan ejemplos tipo para cada caso.

2.1 *Calendarización de proyectos*

Un proyecto es una combinación de actividades interrelacionadas que ejecutadas en un orden particular alcanzan un objetivo general. Una actividad en un proyecto, se entiende como un trabajo específico que requiere tiempo y recursos para su cumplimiento, incluyendo las variables que se ha dado por llamar "milestone", que son etapas especiales dentro del proceso, son pasos sencillos de ejecución, pero muy importantes porque el proyecto no puede fluir sin tomarlas en cuenta. Las complejidades crecientes de los proyectos actuales han demandado técnicas de planeación efectivas y sistemáticas con el objeto de optimizar la eficiencia en la realización del mismo. La eficiencia aquí implica efectuar la mayor reducción en el tiempo requerido para terminar el proyecto, tomando en cuenta la factibilidad económica de la utilización de recursos disponibles.

Una de las debilidades más serias de los métodos de planeación y control tradicionales es su falta de uniformidad, en el sentido de que varían considerablemente en los diversos niveles del organigrama de una empresa y aún en el mismo nivel difieren de ejecutivo a ejecutivo. Cada persona planea sus actividades de acuerdo con su experiencia, su habilidad de organización, la intuición y sentido común que posee. A través de los años se presenta la tendencia a crear sus propios métodos. Es frecuente que al estar trabajando en un proyecto haya necesidad de acelerarlo ya sea porque sus actividades se retrasaron o porque es urgente terminarlo; la inclinación general cuando se usan métodos tradicionales es adelantar todas las actividades lo máximo posible hasta lograr que el proyecto logre su fin en una fecha establecida.

Existen tres categorías generales de elementos en un proyecto *operacionales* o sea acciones, *recursos* o elementos con los que se cuenta y *condiciones o limitaciones* bajo las cuales se va a trabajar. Estas partes estan fuera de control del tomador de decisiones y el objetivo es coordinarlas, los métodos de calendarización son una fuerte herramienta en este sentido.

El principal interés de administrar un proyecto es saber como calendarizar las actividades componentes para poder alcanzar un cierto objetivo, que puede ser: Terminar el proyecto en una fecha determinada, minimizar el costo de la fecha fijada ó minimizar el tiempo total del proyecto.

La calendarización ó administración de proyectos es un ejemplo particular de sistema intermitente, realizado en un solo período de tiempo.En general esta clase de proyectos es compleja y de gran escala, porque involucra un gran número de actividades componentes, que deben ser acomodados en la linea del tiempo, según manifiesten las relaciones de precedencia en cada caso.

La clase de técnicas usadas para analizar, planear y calendarizar proyectos de gran escala tienen una faceta importante en común: todas ellas están basadas en la representación del proyecto como una red de actividades. Por lo que frecuentemente se hace referencia en su solución al análisis de redes, planeación de redes y planeación y calendarización de redes. Las técnicas más frecuentemente utilizadas para la solución de problemas de este tipo son: Los Diagramas de Gannt, Método de Ruta critica (CPM) y la Técnica de Evaluación y Revisión del Programa (PERT). Herramientas ampliamente aceptadas en el área de construcción y en la Industria.

La carta de navegación o carta tipo gráfica, llamada Gannt fué la primera técnica científica para planeación de proyectos y calendarización. El método de las Barras de Gannt que se describe a continuación tiene la intención de disponer de información organizada, oportuna y pertinente, además de emprender camino para las técnicas que buscan la optimización de los recursos.

Desde el punto de vista histórico, el diagrama de Gannt, ó de Barras,desarrollado por Henri L. Gannt, junto con los procedimientos de análisis de redes se iniciaron durante la Primera Guerra Mundial. El diagrama de Gannt aún se usa y es muy útil para analizar la factibilidad de un programa de ruta crítica a costo óptimo. En la figura 2.1a se presenta un ejemplo típico de Diagrama de Gannt. El interés radica en la representación gráfica de las tareas que se deben efectuar en el diagrama, donde la longitud de una barra, como la A-1 en la figura, es análoga al tiempo que se requiere para efectuar dicha actividad. Los beneficios característicos del diagrama de Gannt son:

i) Todas las tareas se expresan gráficamente en un diagrama de fácil comprensión.

ii) Si la persona que tiene a su cargo el proyecto sombrea la barra del diagrama en una parte correspondiente al porcentaje del trabajo que se completa en cada periodo básico, se podrá comprobar el progreso general del sistema de tareas en cualquier punto en el tiempo. Por ejemplo, en la figura 2.1b se aprecia el avance hasta el 7 de mayo; ya se completaron las tareas A-1, B-1, B-2 y están atrasadas las tareas A-2 y C-1.

Desafortunadamente, el diagrama convencional de Gannt no indica causa posible del problema existente en las tareas que no se han cumplido . Esto se corrige aplicando la técnica de líneas de equilibrio.

iii) Cuando los recursos requeridos están limitados, el diagrama de Gannt permite una evaluación inicial del uso planeado de estos recursos.

La carta de Gantt es un excelente artificio de exposición del problema de calendarización de proyectos, sin embargo adolece de una técnica de calendarización porque no provee de una aproximación estructurada, el analista debe utilizar la intuición.

La evolución del diagrama de Gannt hacia una red surge a finales de la década de 1940 cuando aparece el *método de pilares* de la Marina de Estados Unidos, que es un diagrama de Gannt modificado en el que se indican periodos clave, llamados pilares, señalando la duración de las tareas. Las interrelaciones entre pilares se presentan conectando los pilares afectados con líneas sólidas. Los pilares obvios para cualquier tarea incluyen los tiempos de inicio de la tarea y el punto en que es necesario terminar. En los puntos significativos dentro de una tarea, como la terminación de una subpartida de partes que luego permitirían el inicio de otra tarea, conviene colocar otros pilares. El que se encuentra en la terminación de la subpartida y el pilar de inicio de la segunda tarea marcan un ejemplo de interrelación entre pilares y señalan la precedencia de una tarea respecto a otra. En la figura 2.1c se muestra un ejemplo del diagrama de Gannt agregándole pilares.

	Fecha programada Mayo												
Trabajo		3	4	5	6	7	10	11	12	13	14	17	18
A-1		•	•	•									
A-2					•	•	•	•					
B-1		•	•										
B-2				•	•	•							
C-1				•	•	•	•	•	•				
C-2										•	•	•	
D-1									•	•	•	•	•

Diagrama de barras típica

Figura 2.1a

	Fecha programada Mayo												
Trabajo		3	4	5	6	7	10	11	12	13	14	17	18
A-1		•	•	•									
A-2					•								
B-1		•	•										
B-2				•	•	•							
C-1				•	•	•							
C-2										•	•	•	
D-1									•	•	•	•	•

Diagrama de barras con reporte de avance

Figura 2.1b

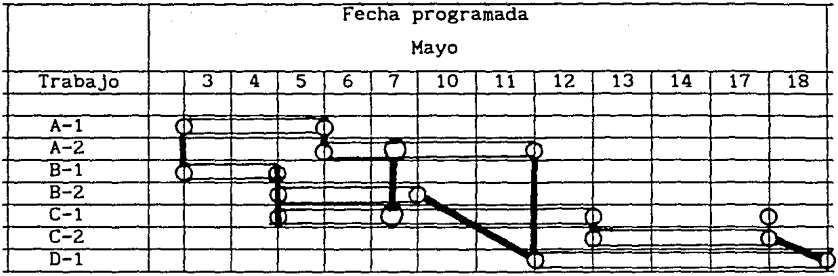


Diagrama de barras con pilares que muestra las precedencias

Figura 2.1c

En este ejemplo elemental se ve rápidamente que la responsable del deslizamiento en las actividad A-2 y C-1 es la actividad A-2, ya que gran parte de C-1, no puede iniciar hasta haber alcanzado un pilar en la actividad A-2, lo cual no ha sucedido. Al igual que con el diagrama de Gannt, el procedimiento de pilares es loable respecto a sus propósitos, pero tiene solo un interés pasajero excepto por su posición en la evolución del proceso que lleva a la red de planeación.

Método de la Ruta Crítica

El método de la Ruta Crítica fué el esfuerzo conjunto de la Compañía de duPont de Nemours y de la División Remington Rand Univac, originado por la búsqueda de un mejor control y un mantenimiento mas completo de las plantas químicas. El proyecto se inició en 1957 y el método de CPM fué puesto a la práctica en 1959 por Kelley y Walker. Desde entonces, su uso se ha extendido considerablemente. Existe un sin número de artículos en intercambio y revistas profesionales reportando aplicaciones y desarrollos metodológicos, y muchos programas de computación para ejecutar los cálculos que se hacen necesarios.

El método de la Ruta Crítica tiene como *objetivo fundamental desarrollar un programa del proyecto que minimice el costo total esperado*, o bién crear un programa o serie de programas, que queden dentro del margen admisible del costo del proyecto, asignando tiempos de inicio y término a las actividades.

El método de la Ruta Crítica inicia con la construcción de una red ó diagrama de flechas que representa las relaciones de precedencia entre las actividades del proyecto. La construcción de la red tiene como función la planeación del proyecto y es el aspecto más difícil de la técnica. Esto involucra análisis y razonamiento cuidadosos, especialmente para los proyectos de gran escala. Los beneficios de este esfuerzo son gratificantes, porque reditúan en un claro entendimiento de lo que puede ser realizado. El planteamiento lógico de la red es fundamental cuando se utiliza el método de ruta crítica, enseguida para buscar su solución deberá aplicarse un método de optimización al problema, que pueden ser las técnicas propias de Teoría de Redes, métodos de programación lineal ó algoritmos heurísticos.

El planteamiento del problema se inicia utilizando la Teoría de Gráficas, específicamente creando una red de actividades en la cual se esquematiza la información; enseguida se define la estructura matemática. Se conviene que N_i sea el nodo que representa al evento i donde i=1,...m de un proyecto que contiene m eventos. Dados dos eventos N_j y N_i, donde N_i precede a N_j entonces A_{ij} representa a la actividad que se origina en el evento N_i y termina en el evento N_j. Se conviene también que T_{ij} y C_{ij} sean respectivamente el costo y la duración de la actividad A_{ij}. La duración puede ser un valor determinístico o bien aleatorio. Por el momento se considera como valor determinístico. Sea A el conjunto de todas las actividades y N el conjunto de todos los m eventos, es decir. Sea Π_k la cadena que conduce del evento inicial N_1 al evento N_j. Sea $t(\Pi_k)$ la duración total de la cadena Π_k, específicamente:

$$t(\Pi_k) = \sum_{A_{ij} \varepsilon \pi_k} t_{ij},$$

Se denota al *tiempo de inicio mas rápido* de un evento N_j, como IR_j y se le define como:

$$IR_j = \text{Max } t_k(\Pi_k), \qquad \ldots 2.2a$$

donde Π_K son todas las posibles cadenas que conectan al nodo inicial N_1 con N_j. El *inicio más rápido del evento inicial* N_1 es

$$IR_1 = 0$$

La ecuación 2.2a indica que se requiere el cálculo de la ruta más larga del nodo N_1 al nodo N_j, en una red acíclica direccional. Si se denota a B(j) como el conjunto de nodos que conectan con N_j, la ecuación 2.2a puede escribirse en forma recursiva de la siguiente manera.

$$IR_j = \underset{i \varepsilon B(j)}{\text{Max}} [\ IR_i + t_{ij}\ ]; \qquad j = 2,3,\ldots m \qquad \ldots 2.2b$$

$$IR_1 = 0$$

El cálculo de todos los inicios más rápidos de todos los eventos debe llevarse a cabo primero con todos los eventos que conectan directamente con el evento inicial; después con todos los que conectan directamente con estos eventos, etc., hasta alcanzar el evento final del proyecto.

De una manera análoga, se pueden realizar otra serie de cálculos empezando en el evento final N_m y concluyendo en el evento inicial. Se denota a la *terminación mas tardía* de un evento N_j como TT_j y se le define como:

$$TT_j = \text{Min}\ [TT_m - t\ (\ \Pi_k\ )\ ] \qquad \ldots 2.2c$$

$$TT_m = IR_m$$

donde (Π_k) es la k-ésima cadena que va del evento N_j al evento final N_m y $t(\Pi_k)$ es la duración total de esta cadena. Se nota en 2.2c que la duración asociada al inicio más rápido del evento final N_m (IR_m), coincide con la duración asociada a la terminación más tardía del mismo evento (TT_m). En forma recursiva se puede escribir a 2.2c como:

$$TT_i = \underset{i \in B(j)}{\mathrm{Min}} [\, TT_j - t_{ij} \,]; \qquad i = m,\ m-1,\ m-2, \ldots 1$$

$$TT_m = IR_m, \qquad \ldots 2.2d$$

donde B(j) denota al conjunto de eventos N_i que preceden al evento N_j.

. Desde el punto de vista de la planeación este método tiene aspectos deseables, incluyendo los siguientes:

i) El desarrollo de la red necesaria permite que un proyecto en gran escala se vea como un sistema completo. La interacción de actividades que comprende el proyecto se puede ver de una sola mirada y se evalúa en función de su efecto sobre otras tareas, sin llevar a cabo ningún análisis de optimización; se pueden preveer y eliminar los cuellos de botella potenciales antes de realmente poner en marcha el proyecto. Muchos usuarios de la ruta crítica afirman que este desarrollo real de la red que permite observar gráficamente las interacciones del sistema, es el aspecto mas favorable del método .

ii) Se puede determinar iterativamente un margen de costo para un proyecto dependiente de un margen factible de tiempos de programa. A la vez, el margen de costo se aprovecha para determinar la factibilidad económica del proyecto en particular. Una conveniencia lateral se deriva del hecho de tener que desarrollar estimados certeros de tiempo y costo para cada tarea, de modo que el margen de costo, y en última instancia, el programa de costo mínimo se pueden determinar.

iii) Mediante el uso de la combinación del esquema de una red con el método de ruta crítica y el procedimiento de diagrama de Gannt, se pueden determinar los requerimientos de recursos para un programa ó una serie de programas en particular. En cambio el efecto de los recursos escasos se puede evaluar en función del margen del costo.

Ejemplo 1

Encuentre la Ruta Crítica de la siguiente red:

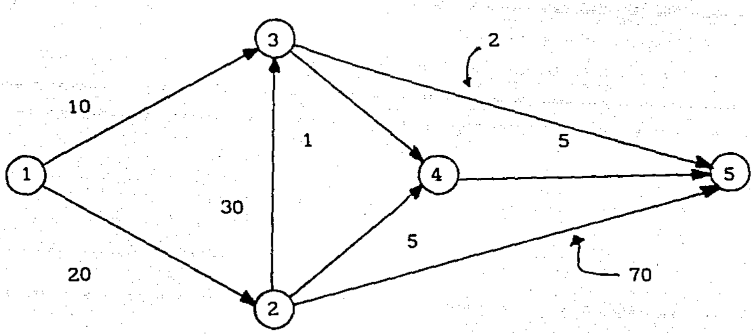


Figura 2.1a

Cálculo de los inicios más rápidos de cada evento.

IR_1 = 0 días,
IR_2 = (0+20) = 20 días
IR_3 = Max (0+10, 20+30) = max (10,50) = 50 días,
IR_4 = Max (20+5, 50+1) = Max (25,51) = 51 días,
IR_5 = Max (20+70, 50+2, 51+5) = Max (90,52,56) = 90 días.

Cálculo de la terminación más tardía de cada evento.

TT_5 = IR_5 = 90 días,
TT_4 = 90 - 5 = 85 días,
TT_3 = Min (90-2,85-1) = Min (88,84) = 84,
TT_2 = Min (90-70, 85-5, 84-30) = min (20,80,54) = 20,
TT_1 = Min (20-20, 84-10) = Min (0,74) = 0

Se define como holgura total de un evento N_i al posible retraso que ese evento podría experimentar, sin causar retraso alguno a la duración total del proyecto. La holgura del evento N_i, se denota por H_i y se define como

$$H_i = TT_i - IR_i, \text{ para toda } i \in N$$

Así en el ejemplo tendríamos:

H_1 = 0-0 = 0 días,
H_2 = 20-20 = 0 días,
H_3 = 84-50 = 34 días,
H_4 = 85-51 = 34 días,
H_5 = 90-90 = 0 días.

Los eventos N_3 y N_4 tienen una holgura de 34 días, mientras que los eventos restantes tienen una holgura nula. Esto quiere decir que si el evento N_1 empieza el día 1, el N_2 debe de empezar exactamente 20 días después, si no se quiere retrasar la terminación del proyecto. Si el evento N_2 se empieza 25 días de N_2, entonces, el proyecto terminaría por lo menos 5 días después de lo programado. En cambio, el evento N_3, que tiene 34 días de holgura, podría empezar no antes del décimo día de empezado el proyecto, pero no después del cuadragésimo cuarto día, si no se quiere retrasar la obra. Al grupo de toma de decisiones, le interesa saber que para no retrasar el proyecto, las actividades A_{12} y A_{25} deben empezar una inmediatamente después de la otra (ya que la holgura es de cero), mientras que las actividades A_{13}, A_{23}, A_{34} y A_{35}, tienen cierta holgura ó rango de retraso, que no afectan la duración de todo el proyecto (mientras se respeten esas holguras).

En resumen, las actividades A_{12} y A_{25} *son criticas* para el proyecto,pues el retraso de una unidad de tiempo en su inicio, ejecución o terminación retrasarían la duración total del proyecto. Recordando, que se entiende como *ruta critica* aquella que se forma del evento inicial y el evento final con eventos cuya holgura es nula.

Se pueden definir otros tipos de holgura, cuya aplicación se hará después. Esas holguras se definirán sobre una actividad y no sobre un evento, como se hizo hasta ahora.

Ejemplo 2

Se construye primero el diagrama de flechas que comprende las actividades. (figura 2.4a)

1.- 1 precede a 2
2.- 2 precede a 4
3.- 3 precede a 5
4.- 5 precede a 6

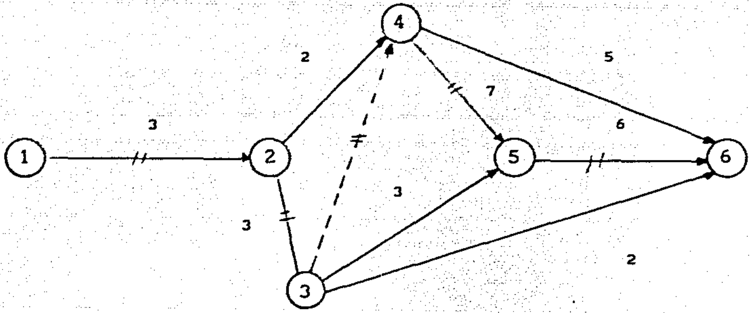


Figura 2.4a

El tiempo requerido para ejecutar cada actividad se indica sobre las flechas. Los cálculos de ruta critica incluyen dos fases. La primera, llamada *paso hacia adelante*, donde los cálculos comienzan desde el nodo de inicio y se mueven hacia el nodo de terminación. En cada nodo se calcula un número que representa el tiempo de ocurrencia más temprano del evento correspondiente. En la figura se muestran dentro de los cuadrados. En la segunda fase, llamado *paso hacia atrás*, comienzan los cálculos desde el nodo de terminación y se mueven hacia el nodo de inicio. El número calculado en cada nodo representa el tiempo de ocurrencia más tardío del evento correspondiente. El paso hacia adelante se presenta a continuación.

Sea $ES_j = \max_i \{ES_i + D_{ij}\}$; (i, j) actividades definidas donde $ES_1 = 0$. Por consiguiente, a fin de calcular ES_j para el evento j, ES_i para los eventos de comienzo de todas las actividades (i, j) que entran, deben calcularse primero. Los cálculos del paso hacia adelante aplicados a la figura proporcionan $ES_1 = 0$ como se muestra en el cuadrado sobre el evento 1. Ya que existe solamente una actividad que entra (1,2), al evento 2 con $D_{12} = 3$,

$$ES_2 = ES_1 + D_{12} = 0 + 3 = 3$$

esto se anota en el cuadrado asociado al evento 2. El siguiente evento que se va a considerar es el 3. (note que el evento 4 no puede considerarse en este punto, ya que ES_3 (evento 3) todavía no se conoce). Por lo tanto:

$$ES_3 = ES_2 + D_{23} = 3 + 3 = 6$$

El valor de ES_4 se puede obtener ahora. Ya que existen dos actividades que entran (2,4) y (3,4)

$$ES_4 = \max_{i=2,3} \{ES_i + D_{i4}\} = \max\{3 + 2,\ 6 + 0\} = 6$$

Esto se escribe en el cuadrado asociado con el evento 4.

El procedimiento continúa de la misma manera hasta que ES_j se calcula para toda j. Por consiguiente

$$ES_5 = \max_{i=3,4} \{ES_i + D_{i5}\} = \max\{6 + 3,\ 6 + 7\} = 13$$

$$ES_6 = \max_{i=3,4,5} \{ES_i + D_{i6}\} = \max\{6 + 2,\ 6 + 5,\ 13 + 6\} = 19$$

Estos cálculos terminan el paso hacia adelante.

El paso hacia atrás comienza desde el evento de 'terminación'. El objetivo de esta fase es calcular LC_i, el tiempo de terminación más tardío para todas las actividades que están en el evento *i*. Por consiguiente, si i=n es el evento de terminación, $LC_n = ES_n$ inicia el paso hacia atrás. En general, para cualquier nodo i,

$LC_i = \min \{LC_j - D_{ij}\}$, para todas las actividades (i, j) definidas

Los valores de LC se determinan de la manera siguiente:

$$
\begin{aligned}
LC_6 &= ES_6 = 19 \\
LC_5 &= LC_6 - D_{56} = 19\text{-}6 = 13 \\
LC_4 &= \min \{LC_j - D_{4j}\} = \min \{13\text{-}7, 19\text{-}5\} = 6 \\
LC_3 &= \min \{LC_j - D_{3j}\} = \min \{6\text{-}0,\ 13\text{-}3, 19\text{-}2\} = 6 \\
LC_2 &= \min \{LC_j - D_{2j}\} = \min \{6\text{-}3, 6\text{-}2\} = 3 \\
LC_1 &= LC_2 - D_{12} = 3\text{-}3 = 0
\end{aligned}
$$

Esto completa los cálculos del paso hacia atrás.

Las actividades de ruta crítica pueden ahora identificarse usando los resultados de los pasos hacia adelante y hacia atrás. Una actividad está en la ruta crítica si satisface las tres condiciones:

i) $ES_i = LC_i$
ii) $ES_j = LC_j$
iii) $ES_j - ES_i = LC_j - LC_i = D_{ij}$

Estas condiciones realmente indican que no existe tiempo de holgura entre el inicio más próximo (terminación) y el inicio más tardío (terminación) de la actividad. Por consiguiente, esta actividad debe ser crítica. En el diagrama de flechas estas actividades están caracterizadas por los números en cuadrados y triángulos siendo los mismos en cada uno de los eventos terminales y de comienzo y que la diferencia entre los números en cuadrado y en triángulo en el evento terminal y el número en cuadrado o en triángulo en el evento de comienzo es igual a la duración de la actividad.

Las actividades (1,2), (2,3), (3,4), (4,5) y (5,6) definen la ruta crítica en la figura 2.4a. Este es realmente el tiempo más corto posible para terminar el proyecto. Note que las actividades (2,4), (3,5), (3,6) y (4,6) satisfacen las condiciones *i)* y *ii)* para las actividades críticas, pero no la condición *iii)*. Por lo tanto, no son críticas. Observe también que la ruta crítica debe formar una cadena de actividades conectadas, la cual abarca la red desde el inicio hasta la terminación.

Determinación de las holguras

Una actividad crítica debe tener holgura cero, de hecho es la diferencia entre el máximo tiempo disponible para realizar la actividad ($LC_j - ES_i$) y su duración (D_{ij}).

Holgura libre. Supone que todas las actividades comienzan tan temprano como sea posible. En este caso, FF_{ij} para la actividad (i,j) es el exceso de tiempo disponible ($ES_j - ES_i$) sobre su duración (D_{ij}); esto es: $FF_{ij} = ES_j - ES_i - D_{ij}$

Los cálculos se muestran en la tabla 2.4a del ejemplo Note que una y solo una actividad crítica, debe tener una holgura total cero. La holgura libre debe también ser cero cuando la holgura total es cero. La inversa no es cierta, sin embargo, en el sentido de que una actividad no crítica pueda tener una holgura libre cero. En el ejemplo se muestra que la holgura total es la misma que la holgura libre. Esto es accidental ya que todos los eventos del proyecto coinciden en estar en la ruta crítica.

Otra técnica de solución es en términos de *programación lineal*, en donde el problema continúa visualizándose como una red, la diferencia estriba en que el planteamiento y la búsqueda de soluciones se hace mediante ecuaciones lineales tratando de optimizar una función objetivo sujeta a una serie de restricciones; un flujo entra en el origen *1* y sale en el nodo terminal *n*. La duración de cada actividad, t_{ij}, es el tiempo de transportación del nodo *1* al nodo *n*. Determinar la trayectoria de flujo máximo del nodo *1* a *n* es equivalente a determinar la trayectoria crítica (la más larga trayectoria en la red).

Sea $x_{ij} > 0$ la cantidad de flujo pasando por el arco (i, j) en la dirección $i \rightarrow j$. Entonces el problema de programación lineal primal está dado por:

$$\mathrm{Max} \sum_{\forall ij} t_{ij}\, x_{ij}$$

sujeto a:

$$\sum_{j} X_{1j} = 1 \qquad \forall\ 1j \qquad \ldots 2.2a$$

$$-\sum_{i} X_{ik} + \sum_{j} X_{kj} = 0 \qquad \forall\ k \neq 1, n \qquad \ldots 2.2b$$

$$-\sum_{i} X_{in} = -1 \qquad \forall\, i, n \qquad \ldots 2.2c$$

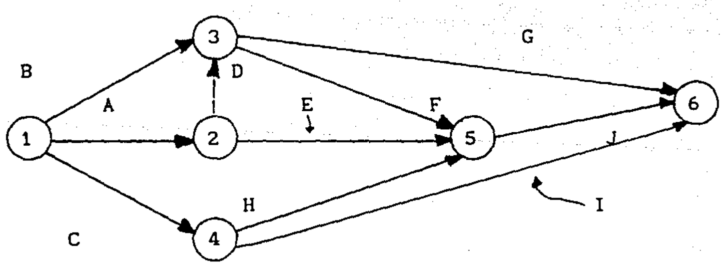
$$X_{ij} > 0 \qquad \ldots 2.2d$$

Donde el nodo 1 es la fuente y el nodo n es el nodo terminal.

La restricción 2.2a presiona a que el flujo corra solo sobre uno de los arcos abandonando el nodo fuente. La restricción 2.2b representa la conservación del flujo en cualquier nodo (cantidad de flujo que entra es igual a la que sale). Mientras que la restricción 2.2c afirma que solo una unidad de flujo llega al nodo terminal. La solución al problema muestra los valores de las X_{ij} para cada arco. La ruta crítica consiste de los arcos cuyo valor es $X_{ij} = 1$.

Ejemplo 1

Use la programación lineal para encontrar la Ruta Crítica de la siguiente red:



Actividad	Duración
A	5
B	4
C	8
E	6
F	4
G	5
H	2
I	3
J	4

Solución:

La función objetivo está dada por:

$$\text{Max } 4x_{12} + 5x_{13} + 8x_{14} + 0x_{23} + 5x_{25} + 6x_{35} + 4x_{36} + 2x_{45} + 3x_{46} + 4x_{56} \qquad \ldots 2.1a$$

sujeto a:

$$x_{12} + x_{13} + x_{14} = 1 \qquad \ldots 2.1b$$

$$-x_{12} \quad x_{23} + x_{25} = 0 \qquad \ldots 2.1c$$

$$-x_{13} - x_{23} + x_{35} + x_{36} = 0 \qquad \ldots 2.1d$$

$$-x_{14} + x_{45} + x_{46} = 0 \qquad \ldots 2.1e$$

$$-x_{25} - x_{35} - x_{45} + x_{56} = 0 \qquad \ldots 2.1f$$

$$-x_{36} - x_{46} - x_{56} = -1 \qquad \ldots 2.1g$$

$$y \ x_{ij} \geq 0.$$

El dual de este problema de programación lineal puede ser presentado como sigue. Sean y_i las variables duales correspondiendo a la i-ésima ecuación del problema primal. Por consiguiente el problema dual se presenta de la siguente manera.

$$\text{Minimice } y_1 - y_6$$

sujeto a:

$$
\begin{array}{lcr}
y_1 - y_2 \quad \geq 4 & & 2.1h \\
y_1 - y_3 \quad \geq 5 & & 2.1i \\
y_1 - y_4 \quad \geq 8 & & 2.1j \\
y_2 - y_3 \quad \geq 0 & & 2.1k \\
y_2 - y_5 \quad \geq 5 & & 2.1l \\
y_3 - y_5 \quad \geq 6 & & 2.1m \\
y_3 - y_6 \quad \geq 4 & & 2.1n \\
y_4 - y_5 \quad \geq 2 & & 2.1o \\
y_4 - y_6 \quad \geq 3 & & 2.1p \\
y_5 - y_6 \quad \geq 4 & & 2.1q
\end{array}
$$

Puesto que cada desigualdad en el problema dual contiene dos y solo dos variables, el problema puede ser resuelto por simple inspección. Una solución de este problema dual es

$$y_1 = 15, \quad y_2 = 10, \quad y_3 = 10, \quad y_4 = 6, \quad y_5 = 4, \quad y_6 = 0$$

La solución dual óptima es $y_1 - y_6 = 15$ (la cual representa la longitud de la trayectoria crítica). Puesto que las desigualdades (2.1i), (2.1k), (2.1m)y 2.1q, se satisfacen como igualdades, las variables correspondientes a estas desigualdades en el problema primal son mayores que cero (relaciones primal-dual); esto es, x_{13}, x_{23}, x_{35} y $x_{56} \geq 0$. Que sustituyendo estos valores en las ecuaciones desde 2.1a hasta 2.1f iguales a cero, se obtiene:

$$
\begin{aligned}
x_{13} \qquad\qquad\qquad\qquad &= 1 \\
x_{23} \qquad\qquad\quad &= 0 \\
-x_{13} - x_{23} + x_{35} \qquad\quad &= 0 \\
- x_{35} + x_{56} &= 0 \\
- x_{56} &= -1
\end{aligned}
$$

donde:

$$x_{13} = x_{35} = x_{56} = 1; \qquad x_{23} = 0$$

Donde la trayectoria crítica es: $1 \longrightarrow 3 \longrightarrow 5 \longrightarrow 6$, con una duración total de 15 días.

Los métodos PERT y CPM fueron desarrollados independientemente y aplicados por primera vez durante los últimos años de la década de los 50. Inicialmente, el método PERT fué diseñado como una técnica de reporte para valorar y controlar el progreso fase a fase de los diversos proyectos del programa de proyectiles dirigidos Polaris. El método CPM, por otra parte, fué concebido originalmente como una técnica de planeación, diseñada para controlar proyectos de construcción, de ingeniería y de mantenimiento de plantas, actualmente está orientado a usarse con el apoyo de computadoras. Las características de una técnica han sido incorporadas a la otra y viceversa. Una diferencia que se ha marcado, es que PERT resulta más adecuado para proyectos de investigación y desarrollo en los cuales aparece más incertidumbre, en cambio en CPM se supone que las duraciones de las actividades son determinísticas, con ese criterio se describirá en esta sección el método PERT.

El diagrama de flechas o red es común a ambos métodos. *El punto importante es desarrollar un modelo de trabajo de un proyecto, creando un diseño a partir del cual pueda prepararse un modelo realista.* Ambas técnicas son igualmente aplicables a la planeación, dirección y control y son adaptables a cualquier tipo de proyecto.

Descripción .- Se supone que el tiempo t_{ij}, asociado a la actividad A_{ij} que une al evento N_j y al evento que lo precede N_i, no es un valor determinístico, sino que puede tomar diferentes valores con diferentes probabilidades.

Se considera que una actividad puede tener tres clases de duración:

i) *Optimista,* que se denota por a_{ij}; es una estimación que se asocia a una actividad si la ejecución va extremadamente bién.

ii) *Pesimista,* se denota por b_{ij}; es una estimación que se asocia a una actividad en condiciones completamente desventajosas.

iii) *Realista,* que se denota por m_{ij}; es una estimación que se asocia a una actividad en condiciones normales.

Se procede a calcular el valor esperado de la duración t_{ij}, en función de las tres estimaciones de duración asociadas a la actividad A_{ij}. Este valor se calcula con la fórmula de la distribución beta que es la más apropiada para estos casos:

$$t_{ij} = 1/6\ (a_{ij} + 4m_{ij} + b_{ij}); \quad \text{para toda } A_{ij}.$$

Con ese valor esperado se aplica la metodología de ruta crítica que se explicó anteriormente. La ruta crítica que se encuentre tendrá una duración esperada de

$$T(\Pi_c) = \sum_{A_{ij} \varepsilon \Pi_c} t_{ij}$$

y una varianza

$$\sigma^2 \pi_c = \sum_{A_{ij} \in \Pi_c} \sigma^2_{ij},$$

donde la varianza σ^2_{ij} de la actividad A_{ij} en la ruta crítica se calcula con la fórmula

$$\sigma^2_{ij} = 1/36\ (b_{ij} - a_{ij})^2$$

Utilizando de la misma manera la Distribución Beta.

Ejemplo 1

Determínese la ruta critica por el método PERT, en la Red de actividades asociadas desde la creación de un nuevo programa de estudios de posgrado hasta que éste se pone en operación recurrente. La duración optimista, realista y pesimista de cada actividad está dada en semanas. En la tabla 2.3a se muestran las equivalencia de los arcos con las actividades y sus cotas.

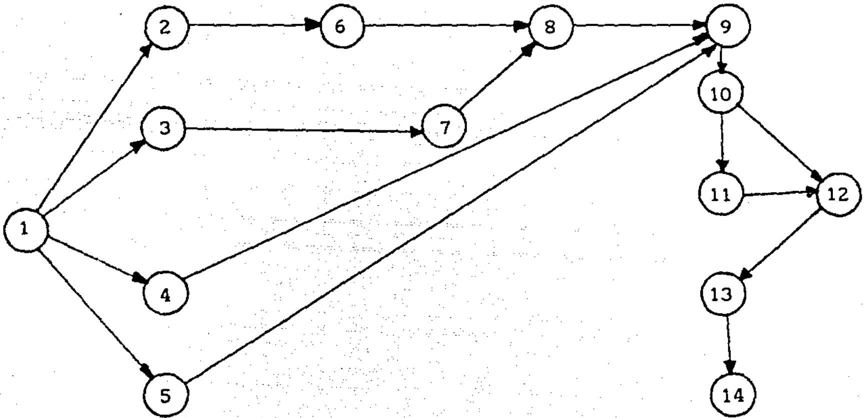


Figura 2.3a

EQUIVALENCIAS DE LOS ARCOS CON LAS ACTIVIDADES

ARCOS	COTAS	ACTIVIDAD
1 ⟶ 2	(10,15,20)	Consultar con el sector industrial
2 ⟶ 6	(20,25,40)	Objetivos y diseño de un programa industrial
6 ⟶ 8	(20,30,35)	Integración admva. al programa
8 ⟶ 9	(0, 0, 0)	Ficticia
9 ⟶ 10	(30,40,50)	Aprobación por autoridades de Ed. Superior
10 ⟶ 11	(2, 3, 5)	Campaña de publicidad
10 ⟶ 12	(4, 8,10)	Contratación de profesores
12 ⟶ 13	(26,35,45)	Primer año de prueba
13 ⟶ 14	(3, 5,10)	Eval. y ajuste del programa a la realidad
1 ⟶ 3	(3, 5,10)	Cons. con sector admvo.,público y privado
3 ⟶ 7	(50,60,80)	Objetivos y diseño de un programa de admcion.
7 ⟶ 8	(25,30,35)	Integración industrial al programa admvo.
1 ⟶ 4	(6,10,20)	Consultar con el sector académico
4 ⟶ 9	(0, 0, 0)	Ficticia
1 ⟶ 5	(2, 3,10)	Consultar con el sector gobierno
5 ⟶ 9	(0, 0, 0)	Ficticia

tabla 2.3a

En las tablas 2.3b y 2.3c se resumen los cálculos del valor esperado de duración y varianza asociada con cada una de las actividades de la red.

En base al valor esperado t_{ij}, se pueden calcular los inicios más rápidos, las terminaciones más tardadas y las holguras de toda la red, tal como se muestra en la tabla 2.3d.

La ruta crítica esperada está formada por los eventos N_1, N_3, N_7, N_8, N_9, N_{10}, N_{12}, N_{13}, N_{14}. La duración esperada del proyecto es de 185.6 semanas con una varianza de:

$$\sigma^2 = 1.4 + 25 + 2.9 + 0 + 10.9 + 1 + 10.2 + 1.4 = 52.8 \text{ semanas,}$$

es decir una desviación estandard de:

$$\sigma = \sqrt{52.8} = 7.27 \text{ semanas}$$

Actividad	Duración optimista	Duración realista	Duración pesimista
A_{ij}	a_{ij}	m_{ij}	b_{ij}
A_{12}	10	15	20
A_{13}	3	5	10
A_{14}	6	10	20
A_{15}	2	3	10
A_{26}	20	25	40
A_{37}	50	60	80
A_{49}	0	0	0
A_{59}	0	0	0
A_{68}	20	30	35
A_{78}	25	30	35
A_{89}	0	0	0
$A_{9,10}$	30	40	50
$A_{10,11}$	2	3	5
$A_{10,12}$	4	8	10
$A_{11,12}$	0	0	0
$A_{12,13}$	26	35	45
$A_{13,14}$	3	5	10

Tabla 2.3b

Actividad A_{ij}	Valor esperado $t_{ij} = 1/6(a_{ij}+4m_{ij}+b_{ij})$	Variancia $\sigma^2=1/36(b_{ij}-a_{ij})^2$	Desviación estandar σ_{ij}
A_{12}	15.0	2.9	1.4
A_{13}	5.5	1.4	1.2
A_{14}	11.0	5.3	2.3
A_{15}	4.0	1.7	1.3
A_{26}	26.7	10.9	3.3
A_{37}	61.7	25.0	5.0
A_{49}	0.0	0.0	0.0
A_{59}	0.0	0.0	0.0
A_{68}	29.2	6.3	2.5
A_{78}	30.0	2.9	1.7
A_{89}	0.0	0.0	0.0
$A_{9,10}$	40.0	10.9	3.3
$A_{10,11}$	3.2	0.3	0.5
$A_{10,12}$	7.7	1.0	1.0
$A_{11,12}$	0.0	0.0	0.0
$A_{12,13}$	35.2	10.2	3.2
$A_{13,14}$	5.5	1.4	1.2

Tabla 2.3c

Evento N_i	Iniciación rápida IR_i	Terminación tardada TT_i	Valor esperado de la holgura H_i	Ruta Crítica Esperada
1	0	0	0	*
2	15	41.3	26.3	
3	5.5	5.5	0	*
4	11	97.2	86.2	
5	4	97.2	93.2	
6	41.7	68	26.3	
7	67.2	67.2	0	*
8	97.2	97.2	0	*
9	97.2	97.2	0	*
10	137.2	137.2	0	*
11	140.4	144.9	4.5	
12	144.9	144.9	0	*
13	180.1	180.1	0	*
14	185.6	185.6	0	*

Tabla 2.3d

Limitaciones de PERT y/o CPM

1) PERT está restringido a proyectos de gran escala. No se usa en operaciones tipo producción.

2) La ejecución actual de una actividad puede alterar el tiempo de término de la red completa.

3) Los tiempos son estimados y por consiguiente sujetos a errores de parte de los procesos.

4) En PERT, la suposición de que los tiempos para terminar las actividades no están correlacionados puede no ser creíble en algunas situaciones reales.

5) Todas las actividades dirigidas a un evento deben ser completadas antes de que el evento sea realizado; en otras palabras la realización parcial no es permitida.

2.3 Secuencia de actividades

La secuencia de actividades está considerado como uno de los más interesantes problemas en el análisis de producción y ha recibido una atención especial por parte de diversos investigadores. *El tópico de secuencia de actividades, requiere determinar el orden para procesar un conjunto de actividades, utilizando varias máquinas de tal manera que resulte óptimo.* El problema es bastante complejo y está lejos de ser completamente resuelto. Se han encontrado soluciones cuando se trabaja con un número pequeño de máquinas, sin embargo, soluciones óptimas para problemas con un gran número de máquinas no existe, dado que es imposible comprobar su optimalidad.

La secuencia de actividades se define como el conjunto de máquinas que están asociadas con un conjunto de trabajos y el objetivo es ordenar las operaciones que van a ser ejecutadas en cada máquina; sujeto el problema a una serie de restricciones y tal que alguna función medible sea optimizada.

Cuatro factores sirven para clasificar un problema de calendarización de tareas en un almacén. *primero*, el patrón de llegada al trabajo. Si n actividades se tienen que realizar simultáneamente en un almacén que está ocioso y disponible inmediatamente para trabajar, se tiene en este caso un problema estático. Si los trabajos llegan intermitentemente, quizá de acuerdo a procesos estocásticos, el problema se dice dinámico. *Segundo*, es necesario especificar el número de máquinas m y el número de trabajos t que componen al almacén. *Tercero*, el proceso de flujo de actividades a través de las máquinas debe ser detallado. Si todos los trabajos siguieran la misma rutina, entonces el almacén será llamado almacén con flujo, en caso contrario es un almacén de trabajo según las facilidades de la manufactrura, ó bién un almacén aleatoriamente enrutado, en el cual las actividades no siguen una secuencia común. *Cuarto*, el criterio para evaluar las alternativas en el almacén juega un rol crítico en el proceso de calendarización.

Una lista típica de criterios de ejecución a optimizar es:

1) Tiempo promedio de flujo
2) Tiempo ocioso de máquinas
3) Tardanza promedio de actividades (Es definida como la diferencia entre el tiempo actual de término y la fecha fijada).
4) Promedio mas temprano de actividades (si una actividad es completada antes de su fecha fijada, entonces el valor de la tardanza es negativa y es referida en ese caso a la tardanza).
5) Tiempo promedio de colas
6) Número promedio de actividades en el sistema.
7) Porcentaje de trabajos retrasados.

La medida de ejecución más utilizada es el lapso ó cantidad total de tiempo requerido para procesar todas las actividades. El objetivo entonces será desarrollar un procedimiento de calendarización que minimice ese lapso.

Los problemas de secuencia de actividades fueron resueltos en un principio con la técnica gráfica de Henry Laurence Gannt que utiliza en un sistema cartesiano la absisa como escala del tiempo y la ordenada para denotar las máquinas. cada operación está representada por un segmento cuyo principio indica el tiempo de inicio planificado y la longitud es porporcional al tiempo de proceso.

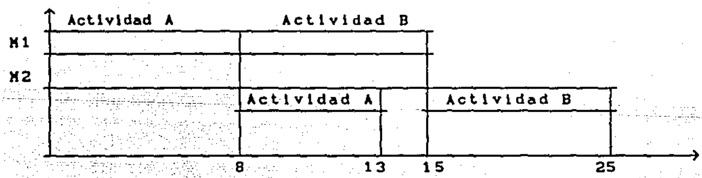
Enseguida se muestra un *ejemplo* que amplia lo antes expuesto. Dos actividades A y B se requiere que sean calendarizadas en dos máquinas M1 y M2. Cada tarea es procesada primero en la M1 y después en la M2. Los tiempos de procesamiento se dán en la tabla siguiente:

	Tiempos de procesamiento	
Tareas	M1	M2
A	8	5
B	7	10

Encuentre la secuencia de estas actividades que minimice el lapso.

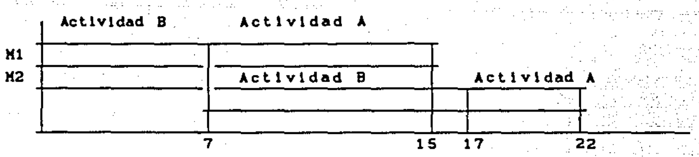
Solución:

Existen dos posibles secuencias: Iniciar con la actividad A ó con B. Suponiendo que iniciamos con A, la cantidad de tiempo total requerido para completar las actividades A ó B son 25 unidades de tiempo, como se muestra en la figura 2.4 b. Si se inicia con B, el tiempo total para realizar las actividades son 22 unidades, como se muestra en la Figura 2.4c.



La actividad A calendariza primero

Figura 2.3a



La actividad B calendariza primero

Figura 2.3b

Cada trabajo es procesado en la máquina M1 primero y después pasa a la máquina M2 (patrón de la secuencia de actividades). Las actividades están permitidas al inicio del periodo de calendarización (patrón de arrivo de trabajo estático). El criterio es minimizar el calendario del tiempo de las actividades (o sea la cantidad total de tiempo requerido para completar el proceso total de todas las actividades).

Enseguida se sintetizan algunas técnicas usadas para la optimización de problemas de secuenciación. Si se incrementa el número de actividades o bién el número de máquinas, el problema de calendarización se torna cada vez más complicado. No existen soluciones óptimas para los problemas con n y m muy grandes, los estudios se han encaminado a crear una serie de algoritmos heurísticos como herramientas de búsqueda y de solución. Un mayor nivel de complicación se tiene al manejar un patrón de llegada de las actividades con carácter dinámico.

A continuación se hará la descripción de diversos casos de calendarización de almacenes en los cuales varía el número de máquinas y/ó el número de actividades, especificamente el caso de N actividades y una máquina, hasta N actividades y M máquinas.

N TRABAJOS, UNA MAQUINA

Este es el problema de secuencias más simple. Existen n trabajos que se requiere que sean procesados por una máquina; Cada actividad experimenta dos tipos de tiempos: tiempo de espera W_i y tiempo de proceso t_i. Se supone que la optimalidad es obtenida si el flujo de tiempo promedio (MFT) de las actividades es minimizado. Dicho tiempo es calculado como:

$$MFT = \frac{\sum_{i=1}^{n} C_i}{n}$$

Donde:

MFT = Tiempo medio de flujo (tiempo promedio en el almacén)
C_i = Tiempo de acabado del trabajo i ($C_i = W_i + t_i$)
n = Número de actividades a ser procesadas.

Ejemplo de *N actividades, una máquina*

Se presentan en la tabla siguiente los tiempos de proceso de cuatro trabajos y una maquina x. Encuentre la secuencia óptima tal que el tiempo medio de flujo sea minimizado.

Actividad i	j1	j2	j3	j4
Tiempos de proceso	7	6	8	5

Solución:

Seleccione la secuencia A tal: J1 ⟶ J2 ⟶ J3 ⟶ J4.

Actividad	W_i	t_i	C_i
J1	0	7	7
J2	7	6	13
J3	13	8	21
J4	21	5	26
	41	+ 26	= 67

La secuencia del tiempo medio de flujo (MFT) es 67\4

Seleccione la secuencia B tal: J2 ⟶ J3 ⟶ J1 ⟶ J4

Actividad	W_i	t_i	C_i
J2	0	6	6
J3	6	8	14
J1	14	7	21
J4	21	5	26
	41	+ 26	= 67

La secuencia de tiempo medio de flujo es 67\4

Enseguida se selecciona otra secuencia C tal que la actividad con *tiempo de proceso mas corto* (SPT) se calendariza primero. La secuencia es: J4 ⟶ J2 ⟶ J1 ⟶ J3.

Actividad	W_i	t_i	C_i
J4	0	5	5
J2	5	6	11
J1	11	7	18
J3	18	8	26
	34	+ 26 =	60

La secuencia del tiempo medio de flujo es: 60/4 = 15

Se concluye que si existen n actividades para procesarse en una máquina, la secuencia óptima que minimiza el tiempo medio de flujo está basado en la regla de tiempo de proceso mas corto (SPT), donde la actividad con el tiempo de proceso mas corto se calendariza primero, seguido por la actividad mas corta y asi sucesivamente, empleando dicha regla, la actividad de tardanza media también se minimiza, donde la tardanza L_i de la actividad i es definida como $L_i = C_i - d_i$.

Este proceso es llamado la secuencia de tiempo de procesamiento más corto (abreviado SPT).

donde:

C_i = tiempo completo de la actividad i

d_i = fecha vencida de la actividad i

Se busca determinar una secuencia óptima para un conjunto de n actividades a ser procesados por una máquina singular. Sean $d_1, d_2 \ldots d_n$ los tiempos de procesamiento de las actividades, los cuales se supone que son conocidos con certeza. Una calendarización será justamente alguna permutación de los enteros 1,2,...n (claramente existen *n* factorial). Por paréntesis rectangulares se denota posición de secuencia en la escala. Esto es, [1] es la actividad que está en la primera posición en la secuencia y $d_{[1]}$ será el tiempo de proceso de la actividad. Una manera de minimizar el flujo de tiempo medio es secuenciando las actividades en orden de tiempo de procesamiento no decreciente, tal que,

$$d_{[1]} \leq d_{[2]} \leq \ldots \leq d_{[n]}$$

La tardanza media (L) está definida como

$$L = \frac{1}{n} \sum_{i=1}^{n} L_i = \frac{1}{n} \sum_{i=1}^{n} (C_i - d_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} C_i - \frac{1}{n} \sum_{i=1}^{n} d_i = MFT - \bar{d}$$

donde $\bar{d}$ es el promedio de las fechas vencidas de los n actividades y es constante. Consecuentemente, la minimización de L es alcanzada usando la regla de SPT, la cual minimiza MFT.

Si el criterio de secuenciación es minimizar la máxima tardanza de n actividades, la secuencia óptima es procesar las actividades en orden de fechas vencidas no decrecientes o

$$d_1 \leq d_2 \leq d_3 \ldots \leq d_n$$

En algunas situaciones, se puede asignar cierto peso g_i para la actividad i para indicar la importancia (prioridad) de la actividad (el más grande g_i, el criterio de secuenciación en este caso es la secuencia de actividades en orden para minimizar el *tiempo medio de flujo ponderado* (MWFT), y el calendario óptimo es obtenido si las actividades son calendarizadas de la siguiente manera:

$$\frac{t_1}{g_1} \leq \frac{t_2}{g_2} \leq \ldots \leq \frac{t_n}{g_n}$$

Ejemplo:

Suponga que los pesos asignados a los trabajos 1,2,3 y 4 en el ejemplo anterior son 7,5,10 y 3 respectivamente, en una escala del 0 al 10, donde 10 representa la mas alta prioridad y 0 representa la mas baja. Encuentre la óptima secuencia que minimiza el tiempo del flujo ponderado promedio de los trabajos.

Solución:

Trabajos	t_i	g_i	t_i/g_i
J1	7	7	1.0
J2	6	5	1.2
J3	8	10	0.8
J4	5	3	1.66

La óptima secuencia es:

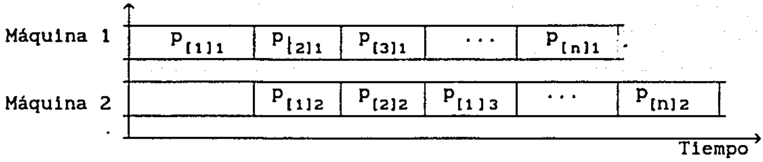
$$J3 \longrightarrow J1 \longrightarrow J2 \longrightarrow J4$$

La calendarización que se hace a un conjunto de n actividades, las cuales deben pasar por dos máquinas en el mismo orden tecnológico. Es decir se tiene un problema de flujo de almacén de dos máquinas y se desea determinar un calendario que minimice el lapso de tiempo que requieren estas n actividades. El procedimiento usado para construir el calendario fué desarrollado por Johnson.

Suponga que todas las actividades son procesadas primero en la máquina 1 y después en la máquina 2 y que la secuencia a la máquina 1 es primero que la máquina 2. El procedimiento de Johnson es como sigue: Encuentra la actividad con el mínimo p_{ij}. (tiempo de procesamiento de la máquina j-ésima en la posición i-ésima). Si este valor mínimo ocurre en la máquina 1 (j=1), asigna esta actividad al primer lugar disponible en la secuencia. Si el valor mínimo ocurre en la máquina 2 (j=2), asigna esta actividad al último lugar disponible en la secuencia. Entonces remueva esta actividad de la lista y repita el proceso. El problema se resuelve escogiendo aleatoriamente cualquier posición en la secuencia por medio de la asignación. El proceso continúa hasta agotar las actividades. Enseguida se dá una noción intuitiva de como trabaja el algoritmo.

Un calendario es mostrado en la figura 2.2a, donde siendo $p_{[i]1}$ el tiempo de procesamiento en la m1 de la actividad en la i-ésima posición, etc. Un calendario es mostrado en la figura 2.2a. Es claro que el último trabajo no puede ser completado mas pronto que el tiempo requerido para procesar todas las actividades en la m1 más el tiempo requerido para procesar el último trabajo en la m2. Entonces

$$F \geq \sum_{i=1}^{n} p_{[i]1} + p_{[n]2} \qquad \ldots\ 2.2a$$



Carta de Gannt de un calendario de dos máquinas.

Figura 2.3c

Similarmente, el último trabajo no puede ser completado en menos tiempo que el requerido para procesar todos los n trabajos sobre la máquina 2, más la demora antes de que la m2 pueda empezar o

$$F \geq \sum_{i=1}^{n} P_{[i]2} + P_{[1]1} \qquad \text{... 2.2b}$$

Las sumatorias en las ecuaciones 2.2a y 2.2b no son afectadas por la frecuencia y solo pueden ser modificadas estas cotas por la selección de $P_{[n]2}$ y $p_{[1]1}$. Por consiguiente se selecciona la actividad con el más pequeño p_{ij}. Si j=1, se pondrá esta actividad en el primer lugar en la secuencia, tal que minimice $p_{[1]1}$. Si j=2 se acomodará esta actividad en el último lugar de la secuencia para minimizar Ahora con la primer actividad secuenciada, se puede repetir el mismo argumento para el conjunto de n-1 actividades restantes.

Ejemplo de N actividades, dos máquinas

Las máquinas del ejemplo taladran y remachan seis actividades cuyos tiempos se muestran en la tabla de abajo. Para cada actividad un agujero es taladrado seguido del remachado. Encuentre la secuencia óptima que minimiza el lapso de todas las actividades.

Actividad	J1	J2	J3	J4	J5	J6
Taladrado	4	7	3	12	11	9
remachado	11	7	10	8	10	13

El taladrado y remachado esta referido a las máquinas M1 y M2, respectivamente. Constrúyase tabla de secuencias eligiendo el menor tiempo y la actividad correpondiente se pondrá en ese lugar, así sucesivamente eliminando cada vez la actividad correpondiente de la tabla principal y

Tabla de secuencias

J3 J1 – – – –

entonces J3 y J1 deberán ser borrados de la lista

Actividad	J2	J4	J5	J6
M1	7	12	11	9
M2	7	8	10	13

J1 será borrado de la lista de actividades y la tabla de secuencias queda de la siguiente manera:

Actividades	J2	J4	J5	J6
M1	7	12	11	9
M2	7	8	10	13

Se continúa de esa manera hasta completar la tabla de secuencias, en este caso se tienen dos por los tiempos iguales de la actividad J2.

Tabla de secuencias					
J3	J1	J6	J5	J4	J2

Tabla de secuencias					
J3	J1	J2	J6	J5	J4

Cualquiera de las dos secuencias dadas minimiza el lapso para todas las actividades.

EXTENSION AL CASO DE N ACTIVIDADES Y DOS MAQUINAS

El prototipo que se analizará será cuando se tengan n actividades y tres máquinas. Para este caso se ha dado considerable atención a los problemas de calendarización El algoritmo de Johnson aplicado al ejemplo de dos máquinas puede ser extendido para tres máquinas bajo ciertas circunstancias. Si cualquiera de las siguientes dos condiciones son verdad, entonces el método es aplicable:

$$\min \{p_{i1}\} \geq \max \{p_{i2}\}$$

ó

$$\min \{p_{i3}\} \geq \max \{p_{i2}\}$$

Esto es, el método de Johnson es aplicable si la máquina 2 está completamente dominada por la primera ó por la tercera máquina. El trabajo que se desarrolla dentro del proceso involucra definir dos máquinas artificiales, sean 1' y 2', con tiempos de proceso:

$$p_{i1'} = p_{i1} + p_{i2}$$

$$p_{i2'} = p_{i2} + p_{i3}$$

El algoritmo de Johnson se aplica a este problema con dos nuevas máquinas.

Ejemplo *de n actividades, tres máquinas*

Encuentre la secuencia óptima para las siguientes seis actividades en M1 M2 M3.

Tiempos de proceso

Actividad	M1	M2	M3
1	5	3	9
2	7	2	5
3	4	3	7
4	8	4	3
5	6	2	2
6	7	0	8

Se comprueban las condiciones:

$$\min t_{i1} \leq \max t_{i2} \; ; \quad i = 1, 2, \ldots ,6$$
$$\min t_{i3} \leq \max t_{i2} \; ; \quad i = 1, 2, \ldots ,6$$

$$\min t_{i1} = \min \{ t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16} \} = 4$$

$$\max t_{i2} = \max \{ t_{12}, t_{22}, t_{32}, t_{42}, t_{52}, t_{62} \} = 4$$

Como la condición se satisface, es posible aplicar el algoritmo de Johnson

Actividad	M_1'	M_2'
1	8	12
2	9	7
3	7	10
4	12	7
5	8	4
6	7	8

Aplicando el algoritmo de Johnson se encuentra que las secuencias óptimas son las siguientes:

$3 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5;\quad 6 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$

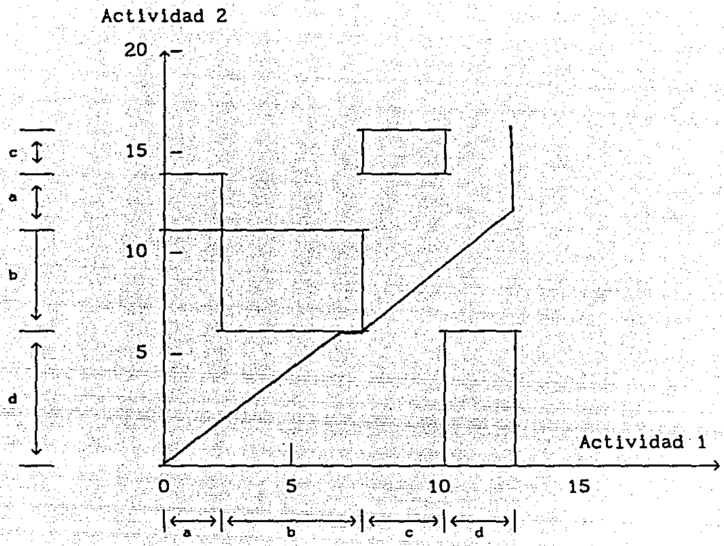
$3 \rightarrow 6 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 5;\quad 6 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 5$

DOS ACTIVIDADES, M MAQUINAS

Existe una solución gráfica simple para el caso de dos actividades y *m* máquinas. El procedimiento se ilustrará con un ejemplo. Suponga que dos actividades van a ser procesadas por cuatro máquinas. Sean las máquinas designadas por las letras *a*, *b*, *c*, *d*. El orden tecnológico para la actividad 2 es *d*, *b*, *a*, *c*, los tiempos en los procesos se enmarcan en la tabla 2.2b. Una representación gráfica del problema es mostrado en la figura 2.2b Los ejes representan la cantidad de trabajo necesario para terminar las actividades, con los tiempos de máquina en el orden tecnológico propio (desde la restricción para dos actividades).

	Mäquina			
Actividad	a	b	c	d
1	2	5	3	2
2	3	5	2	6

Tabla 2.2b Tiempos de procesamiento



Solución gráfica al problema de calendarización de dos actividades y cuatro máquinas.

Figura 2.3d

Las coordenadas representan un estado posible de completez para las dos actividades. Los tiempos de procesamiento en producto cartesiano de las actividades 1 y 2 es mostrada para cada máquina como una región sombreada. Claramente estas regiones no son factibles y representan procesos simultáneos de ambas actividades en la misma máquina.

Una solución para este problema es cualquier línea del punto (0,0) al punto ($\sum_{i=1} p_{i1}$, $\sum p_{i2}$), el cual no pasa través de una región sombreada.

La línea puede estar compuesta de segmentos horizontales (trabajo sobre la actividad 1 solamente) y verticales (trabajo sobre la actividad 2 solamente) y el segmento de 45 grados (trabajo simultáneo de ambas actividades).

Un calendario de lapso de tiempo mínimo es un línea que maximiza la longitud de segmentos verticales y horizontales, esto es, un calendario que maximiza la cantidad de proceso simultáneo. Puede ser determinado por ensayo y error. Usualmente, sólo unas pocas líneas deben ser trazadas antes de encontrar la solución óptima.

MINIMIZACION DE LOS COSTOS POR TIEMPOS DE PREPARACION

En los casos presentados anteriormente, no se han considerado por separado los costos proveniente del tiempo de preparación de las máquinas y los tiempos de uso real, se tomó solo un valor t_{ij}, sin embargo, existen muchas situaciones prácticas donde el tiempo de preparación para una actividad sobre una máquina es dependiente de la actividad precedente. Por ejemplo una compañía que mezcla pinturas y usa la misma máquina para mezclar diferentes colores. La preparación de la máquina para cada color depende del color que fué procesado previamente. Puesto que la suma de los tiempos de proceso son constantes para todas las actividades, en cambio los tiempos de preparación dependen de la secuencia, y una secuencia óptima es aquella que minimiza la suma de los tiempos de preparación.

Este problema es similar al bién conocido problema del agente viajero, en el cual un viajero, inicia en una ciudad y desea visitar cada una y solo una vez n-1 restantes y regresar a la primera. Se está buscando un orden óptimo que minimice la distancia total. La distancia entre ciudades en el caso del agente viajero correponde al tiempo de preparación en el problema de secuencias. Con la técnica de asignación y acotamiento se ha resuelto este problema.

BALANCEO DE LINEAS DE ENSAMBLADO

Cuando un producto ó una familia de productos tecnológicamente similares, se presentan en gran volúmen y con demanda estable por grandes períodos de tiempo, se justifica económicamente la necesidad de plantear un diseño que involucre máquinas y trabajadores, éstos ejecutan operaciones manuales en las máquinas y físicamente son acomodados en una secuencia, de acuerdo al orden tecnológico de las etapas de manufactura, el propósito de realizar tal estudio es minimizar costos.

El desarrollo de las empresas y el avance tecnológico exigen un refinamiento en cuanto a optimización de la capacidad productiva; esto implica involucrar en los diseños mayor número de variables y conocer la organización y funcionamiento de los sistemas de manera más profunda, los resultados obtenidos son gratificantes, aunque el proceso se torne más complejo. El tercer gran bloque perteneciente a la Teoría de Calendarización que pretende atacar este tipo de problemas se conoce con el nombre de *balanceo de líneas de ensamblado.*

El capítulo se desarrolla como sigue: En la primera sección seplantea el objetivo que persigue el estudio del balanceo de líneas de ensamblado y la necesidad de utilización de diseños para resolver los problemas; en el segundo inciso se recopilan ciertos conocimientos que son necesarios como lenguaje para entender y estructurar los modelos; en la tercera sección se relaciona el problema de balanceo de ensamblado de líneas con la Teoría de Flujo en Redes; en la cuarta sección se presenta la formulación matemática del modelo y en el último inciso se proporcionan métodos de solución y ejemplos.

3.1 *Descripción del problema*

El diseño de modelos de balanceo de ensamblado de línea tiene como objetivo fundamental lograr que alguna medida de ejecución de línea sea optimizada, asignando tareas individuales a las estaciones de trabajo. El sistema es continuo y está caracterizado por el movimiento de una pieza de trabajo de una estación a la siguiente;de tal manera que las tareas individuales requeridas para completar el producto estén divididas y asignadas a las estaciones de manufactura. Se dice que una línea está balanceada perfectamente si todas las estaciones generan una cantidad igual de trabajo y consecuentemente tienen un movimiento continuo y suave arrojando un producto terminado en el menor tiempo posible.

En el control de los sistemas productivos el análisis mas detallado que ha logrado tenerse es el que se refiere a la secuencia de actividades y al balanceo de línea de ensamblado, en el primero logrando que en el proceso que se dá entre máquinas y trabajos, alguna medida de ejecución sea optimizada y en el caso de balanceo de línea de ensamblado, la operaciones de ensamblado son asignadas a varias estaciones de trabajo a lo largo de la línea de ensamblado con el fín de alcanzar un balance entre las estaciones y el crecimiento de una completa eficiencia de la línea de ensamblado.

El balanceo de línea de ensamblado no había sido analizado, ni clasificado como tal. Es muy posible que como un reflejo de las necesidades de la época, en función del desarrollo industrial y el tamaño de las empresas, se buscara una optimización más gruesa,no a tanto detalle en cuanto a la minimización de costos en todas los niveles y etapas de un proceso productivo.

Aún en la década anterior los escritos sobre Teoría y Métodos de Solución de Secuenciación y Calendarización mencionados en los libros de Investigación de Operaciones manejaban las ideas descritas anteriormente, sin incluir, ni mencionar explícitamente el concepto de *balanceo de líneas de ensamblado,* los dos tipos de problemas básicamente tratados eran el que se refería a la selección de una disciplina para una cola ó una línea de espera y el segundo tipo de problema tratado correponde a proyectos o trabajos que consisten de actividades que se deben realizar en una secuencia especifica.

Estos problemas implican determinar cuánto esfuerzo se debe hacer para ejecutar cada actividad y cuándo programarla de modo que se optimice una parte de ejecución o de manera integral el proyecto. A estos proyectos se les denominó primeramente problemas de coordinación.

Las palabras calendarización y secuenciación en el contexto de las aplicaciones, se han asociado con una amplia gama de problemas cuantitativos. Hablando de manera genérica la secuenciación se refiere a la ordenación de una colección de trabajos o tareas que se desean ejecutar, mientras que la ordenación se relaciona con la asignación de puntos sobre el tiempo que indican cuando debe ser ejecutada cada actividad. En un ambiente determinístico los dos conceptos se pueden intercambiar fácilmente porque cuando las características de los problemas se conocen *a priori* y las actividades se inician tan rápido como es posible entonces una secuencia determina un calendario y viceversa.

Un *calendario* se define formalmente como un *orden* asociado conlas operaciones de la maquinaria de una empresa. Un *modelo de calendarización* incluye las relaciones maquinaria - actividad y contiene restricciones de enrutamiento entre otras.

Ejemplos típicos de estas líneas de ensamblado son el armado de automóviles, lavadoras y secadoras eléctricas, ensamblado de computadoras, fabricación de juguetes y en general la manufactura de cualquier herramienta electrónica. Las líneas de ensamblado son una parte muy importante de las operaciones de ensamblado y especificamente de cualquier Compañia Manufacturera; no obstante esta actividad puede llegar a reducirse notablemente a través de la robotización, para la cual obviamente se hace necesario en un principio un buen diseño de calendarización..

Se muestra enseguida en la figura 3.1a un diagrama típico de ensamblado de línea. La disposición de trabajo a través de la línea de ensamblado variará de acuerdo: al tamaño del producto que se hace necesario ensamblar, los requerimientos de precedencia, el espacio disponible, los elementos de trabajo y la naturaleza misma del trabajo que va a ser ejecutado.

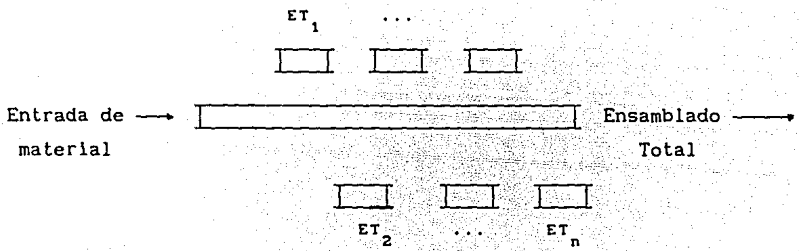


Figura 3.1a Ensamblado típico de Línea

3.2 *Componentes conceptuales*

Los modelos y diseños que utilizan las técnicas *del balanceo de una linea de producción* intentan optimizar la producción no solo minimizando costos relacionados con la calendarización de actividades -máquinas ó con la secuenciación de actividades; sino que *el balanceo de lineas de ensamblado* podria pensarse como un refinamiento en cuanto a optimización en los sistemas productivos se refiere, porque es una etapa posterior a la identificación de un conjunto de máquinas con un conjunto dado de actividades o de operaciones que se resuelve con las técnicas de *actividades en un almacén*; es decir es una segunda etapa ó paso posterior de optimización, que busca mejorar los sistemas productivos minimizando costos al planear las actividades de una empresa y equilibrando los periodos de tiempos que en cada etapa del proceso de producción son invertidos.

El resultado de la fabricación de un articulo es llamado una linea de fabricación si el proceso de producción inicia con materias primas hasta lograr un objeto determinado ó llamado linea de ensamblado, si el producto es el resultado de la unión de varias partes elaboradas. En este sentido no se hará diferencia alguna en el presente estudio al analizar y plantear los problemas.

Existen dos tipos principales de problemas de ensamblado de linea: El Balanceo de estaciones de trabajo y la conservación del ensamblado de linea en producción continua.

El problema de lineas de ensamblado se puede iniciar de la siguiente manera: Dado un producto que requiere N elementos de trabajo para ser completado, encontrar el perfecto balance de la linea de ensamblado. Cuando el perfecto balance no puede ser alcanzado, se mide la efectividad del balance por:

1. *Linea de efectividad* (LE): Es el radio del tiempo total de la estación entre el ciclo del tiempo multiplicado por el número de estaciones de trabajo. Es expresado como

$$LE = \frac{\sum_{i=1}^{k} ST}{(K)(CT)} \cdot 100\ \%$$

Donde:

ST = Es el tiempo de estación en la estación i
K = Número total de estaciones de trabajo
CT = El ciclo del tiempo

2. *Suavidad del indice* (SI): Es un indice que indica la suavidad relativa de un balance de ensamblado de linea dado. Una suavidad de indice con valor igual a 0 indica un perfecto balance.

$$SI = \sum_{i=1}^{k} (ST_{max} - ST_i)^2$$

Donde:

ST_{max} = Máximo tiempo de estación
ST_i = Tiempo de estación de la estación *i*
K = Número total de estaciones de trabajo

Diseñando un ensamblado de linea, las siguientes restricciones deben ser impuestas en el grupo de los elementos de trabajo:

1. Relaciones de precedencia

2. El número de estaciones de trabajo no puede ser mayor que el número de elementos de trabajo (operaciones). El número mínimo de estaciones de trabajo es $1 \leq K \leq N$.

3. El ciclo del tiempo es mayor que o igual al tiempo máximo de cualquier tiempo de estación y del tiempo de cualquier elemento de trabajo T_i. El tiempo de la estación no excederá el ciclo del tiempo, esto es,

$$T_i \leq ST_i \leq CT$$

Una linea de producción consiste de una secuencia con *n estaciones de trabajo*. El producto se procesa de la estación 1 a la estación N. En cada estación de trabajo una *operación* tiene lugar. N unidades de producto están simultáneamente en proceso en diferentes etapas hasta su término. La asignación de las tareas a las estaciones de trabajo es llamado el *balance*. El agrupamiento de las tareas observa varias restricciones. Las más numerosas son las relaciones de precedencia, las cuales limitan el orden en el cual las tareas deben ser ejecutadas.

Sea c el ciclo del tiempo de la linea (entendiéndolo como la cantidad de tiempo transcurrido entre dos unidades sucesivas iniciando o completando el proceso en una estación de trabajo particular) y p_i, $i = 1,2,...N$, la duración de la i-ésima operación. La duración p_i consiste de los intervalos de tiempo enteros requeridos a la estación i para recibir, posicionarse, realizar el trabajo, abandonar la maquinaria y transportar la unidad a la siguiente estación. Si se agrupan las tareas de tal manera que todos los tiempos de duración sean iguales ($p_i = p$, $\forall i$); la linea está perfectamente balanceada; entonces $c = p$. Una linea puede estar perfectamente balanceada cuando $p_i = k_i c$, con k_i entero y positivo. Obviamente, máquinas idénticas k_i ó trabajadores han ejecutado operaciones i simultáneamente sobre k_i unidades diferentes y sucesivas. Esta es una linea con estaciones de trabajo múltiple.

Como el producto se mueve de una estación a la siguiente, implica que el promedio del ciclo del tiempo debe ser el mismo en cualquier estación de trabajo e igual a c, en ese caso se tiene un perfecto balance. Considere ahora una linea desbalanceada teniendo alguna operación i fuera de balance con $p_i < c$. Se puede correr la estación i continuamente, lo cual significa que tan pronto arriva una unidad, la estación se ocupa con una duración p_i, después de lo cual es ocioso para el intervalo $(c - p_i)$ hasta que la siguiente unidad llega.

El producto debe fluir suavemente sin obstáculos, a lo largo de la linea. Sin embargo, se pueden seleccionar varios intervalos ociosos $(c - p)$ en periodos de tiempo más grandes corriendo una parte del tiempo la operación i y almacenando el producto alternativamente antes y después de la estación. El flujo del producto se convierte en intermitente y trabaja acumulando inventarios. Tomando lo anterior en cuenta, se encontrará, que el tiempo promedio transcurrido entre dos productos sucesivos sigue siendo c.

En el caso más general se puede obtener una línea perfectamente balanceada. La estación i se asocia con un tiempo ocioso $(c-p_i)$ en cualquier ciclo c. Si se acumulan estos tiempos ociosos para la línea completa, una cantidad d llamada *balance de demora* resulta.

$$d = Nc - \sum_{i=1}^{n} p_i$$

Se puede expresar la demora del balance en términos relativos:

$$d = \frac{Nc - \sum_{i=1}^{n} p_i}{Nc} \cdot 100$$

Puesto que el tiempo ocioso es caro en términos de labor y equipo, el objetivo es equivalente a minimizar el producto Nc.

En el contexto de altos volúmenes de producción, la calendarización implícitamente se alcanza una vez que la línea de producción ha sido satisfecha. Agrupando las tareas en operaciones y colocándolas en las estaciones de trabajo, se decide sobre el flujo del producto, la secuencia en la cual será ejecutadas, y los tiempos de inicio y término. La calendarización de operaciones se convierte en un problema que establece la línea de producción.

Para ilustrarlo considérese un proceso de producción consistiendo de J tareas. Una tarea es denotada por T_j y tiene un tiempo de ejecución de t_j. Se muestra un ejemplo en la figura 3.2a con J = 12 tareas y duración total de tareas de 55. Como se dijo el objetivo es agrupar las tareas en N estaciones de trabajo en una línea de producción.

Evidentemente:

$$\sum_{i=1}^{n} p_i = \sum_{j=1}^{J} t_j = 55.$$

La tabla de la figura 3.3a contiene los datos básicos, las relaciones de precedencia y los tiempos de ejecución de las tareas. El diagrama de precedencia, cuyos nodos representan las tareas, sintetiza con el uso de flechas las restricciones de secuencia tecnológica.

En la construcción del diagrama de precedencia se sigue un procedimiento dado por Jackson

Paso 1) En la columna 1 en el lado izquierdo se representan todas las tareas, las cuales no tienen predecesoras.

Paso m) (iniciar con m = 2; repita este paso hasta que todas las tareas hayan aparecido en el diagrama, con cada repetición incrementando m en 1)

Tarea Tj	Predecesores inmediatos	Duración
T1	-	6
T2	-	9
T3	T1	4
T4	T1	5
T5	T2	4
T6	T3	2
T7	T3, T4	3
T8	T6	7
T9	T7	3
T10	T5, T9	1
T11	T8, T10	10
T12	T11	1
Duraciön total = 55		

Tabla 3.2a

(a) En la columna m, a la derecha de la columna m-1, se representan todas las tareas no antes incluídas en el diagrama y las cuales no necesitan ser seguidas por cualquier tarea la cual no este aún representada.

(b) Trace todas las flechas de las tareas en columna m-1 a las tareas en columna m, las cuales deben seguirla. Repita este proceso reemplazando la columna m - 1 por m - 2, ... , 1 sucesivamente.

Se supone que las duraciones de ejecución son siempre enteras. Se puede lograr usando una unidad apropiada de tiempo. En la figura 3.2b se muestra un ejemplo en el cual el balance de la demora depende del número de estaciones y del ciclo de tiempo.

La figura 3.2b contiene una familia de curvas de la forma $N_c = (\sum_{j=1}^{n} t_j) / (1-d_x/100)$, teniendo d_x como parámetro. Se limita para el caso en que $c \geq \max p_i$. Puesto que por definición, las tareas no pueden estar divididas, se sigue que $\max p_i \geq \max t_j$. De donde $c \geq \max t_j$. Por consiguiente no se han considerado los ciclos de tiempo mas pequeños que 10. Es claro que cuando $N = 1$ todas las tareas son puestas en la misma estación y $c = 55$ y no es necesario considerar un ciclo de tiempo mayor que

$$\sum_{j=1}^{J} t_j = 55.$$

Para una c dada, el orden para alcanzar el mas pequeño valor del producto Nc, el número mínimo requerido de estaciones N_{min}, está dado por

$$N_{min} = \min\left(N \geq \frac{\sum_{j=1}^{n} t_j}{c} \right);\ N \text{ entero}$$

Por ejemplo, si se selecciona $c = 22$ se necesitan al menos tres estaciones, en tal caso $d_{\%} = 16.67$ (ver figura 3.2a). La curva superior es la localización de las combinaciones (N_{min}, c) las cuales garantizan la minimización del balance de la demora. La curva es transpuesta en la gráfica inferior que es conocida como la función del balance de la demora. Sea el punto $c = 11$, $N = 5$, teniendo $d_{\%} = 0$.

Esto implicará un perfecto balance, pero desafortunadamente, no se pueden agrupar las tareas del ejemplo en 5 estaciones de trabajo con un ciclo de tiempo de 11.

Note que, una vez que se han supuesto enteros los tiempos de ejecución, solo se tienen que considerar valores enteros para el ciclo de tiempo. Esto explica porque la curva no toca al cero y retrasa la hipérbola a todos los valores de N.

Para establecer una línea con balance de demora mínima se tienen dos caminos:

a) Fijar el ciclo de tiempo y minimizar el númeo de estaciones

b) Fijar el número de estaciones y encontrar el balance el cual minimiza el ciclo de tiempo.

En general, en el contexto del proceso del sistema, las tasas de producción constituyen decisiones de planeación agregada, tal que el círculo de tiempo tiene que ser fijado para asegurar el alcance de la tasa de producción preestablecida. Sin embargo existe alguna flexibilidad seleccionando el valor del ciclo de tiempo. Si una tasa de producción puede ser conseguida operando en un cambio de línea con ciclo *c*, la misma salida puede resultar de una línea mas lenta (i.e., con un c mas largo) trabajando horas extra ó en dos cambios, ó corriendo dos líneas paralelas idénticas. O si una línea es mas rápida en operación (i.e. teniendo un c mas pequeño) tendrá que correr solo una parte del tiempo. Obviamente, estas alternativas afectan los costos agregados relacionados con horas extras, tiempo ocioso, inventario. Para el ejemplo, la figura muestra que los mejores valores teóricos para c son 11, 14, 19 y 28.

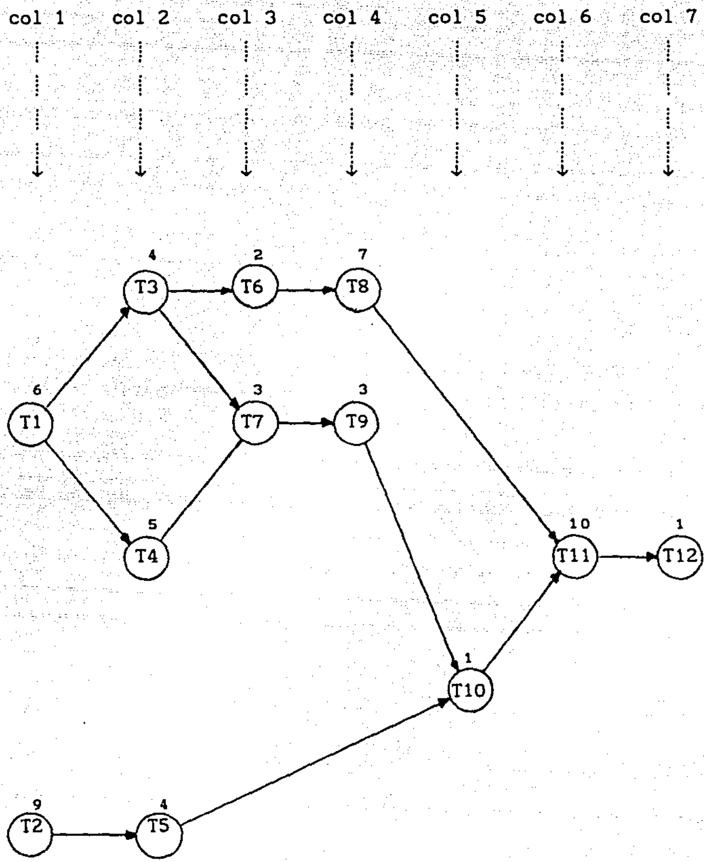
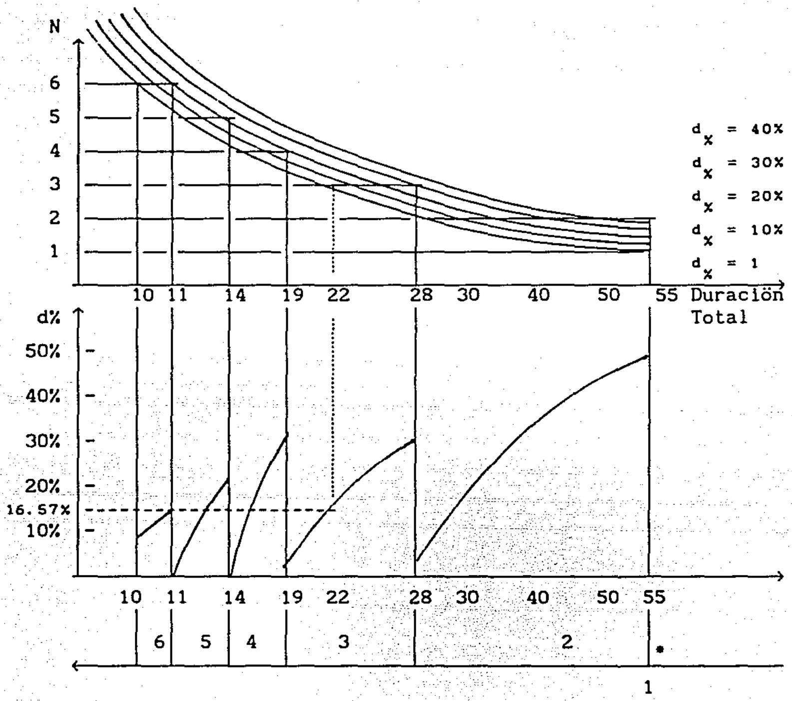


Diagrama de precedencia para un proceso de producción

Figura 3.2a



Número de estaciones, ciclo del tiempo y balance de la demora

* representa $c = 22$, $N = 3$, $d_{\%} = 16.67\%$

figura 3.2b

3.3 *Planteamiento en Términos de Teoría de Redes*

Un problema de balanceo de ensamblado de línea puede plantearse como un problema de Teoría de Redes, básicamente su desarrollo y expansión en los años recientes, se debe a esta manera de conceptualizarlo. El ensamblado de un producto que tenga una manufactura complicada en su elaboración conviene describirlo en términos de una gráfica de precedencia. Cada nodo x, de la gráfica representa una operación de ensamblado, requiriendo un tiempo de procesamiento, los arcos representarán los costos de transporte, de operación o del tiempo cuantificado en costo. El ensamblado final de un producto se logra moviendo los productos en una línea de ensamblado, recorriendo durante el proceso varios estaciones de subensamble, (varios nodos).

Cada una de estas estaciones de subensamble ejecuta una o más operaciones obedeciendo a las leyes de precedencia; es decir una operación se ejecuta solo si las actividades precedentes ya han sido satisfechas, que en forma gráfica sería, encontrar una trayectoria, o sea una ruta dirigida que va del nodo fuente al nodo sumidero, se detecta una ruta a través de las estaciones de subensambles, formando una red conexa, que es aquella que está unida en todos sus nodos, es decir, pasando solo una vez por cada uno de los subensambles, teniendo como nodo fuente la primera etapa dentro del proceso y subsecuentemente ir siguiendo la ruta hasta alcanzar un producto terminado.

Matemáticamente, el problema consiste en particionar el conjunto de los nodos en g subconjuntos, $\{A_1\}$, $\{A_2\}$, ... ,$\{A_g\}$, donde el conjunto $\{A_i\}$ contiene las operaciones asignadas a la estación i.

Para que las restricciones de precedencia se obedezcan es necesario tener lo siguiente:

$$\text{si } x > y, \quad x \in \{A_i\}, \qquad y \in \{A_j\}, \quad \text{entonces } i \le j$$

El tiempo de procesamiento asignado a la estación i es $p(A_i)$.

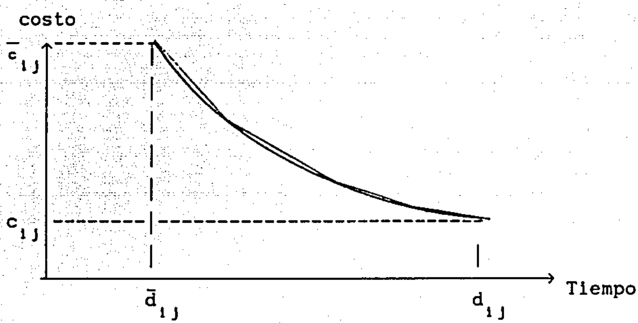
Existen dos diferentes versiones del problema de balanceo de ensamblado de línea:

i) Minimizar g sujeto a $p(A_i) \le$ una constante dada, para toda i

ii) $\min_{g} \max_{i} p(A_i)$.

Otro aspecto dentro de este método es la cuantificación de los costos, que están relacionados principalmente con los tiempos de procesamiento, en general en el costo total se incluye a los costos directos y a los costos indirectos. Este aspecto es muy importante porque cada arco de la red deberá estar etiquetado con un costo. Eseguidase hace un análisis de la relación costo - actividad. Cada actividad A_{ij} tiene una duración asignada d_{ij} unidades de tiempo, a un determinado costo c_{ij}. Al tener la necesidad de acortar la duración de una actividad, a una nueva $\bar{d}_{ij}$ con $\bar{d}_{ij} < d_{ij}$, el costo de esta actividad deberá de aumentar necesariamente a un nuevo valor $\bar{c}_{ij}$ con $c_{ij} < \bar{c}_{ij}$, esto es consecuencia de un uso más intenso de los recursos o bién de un incremento en los mismos. Los costos anteriores son únicamente directos.

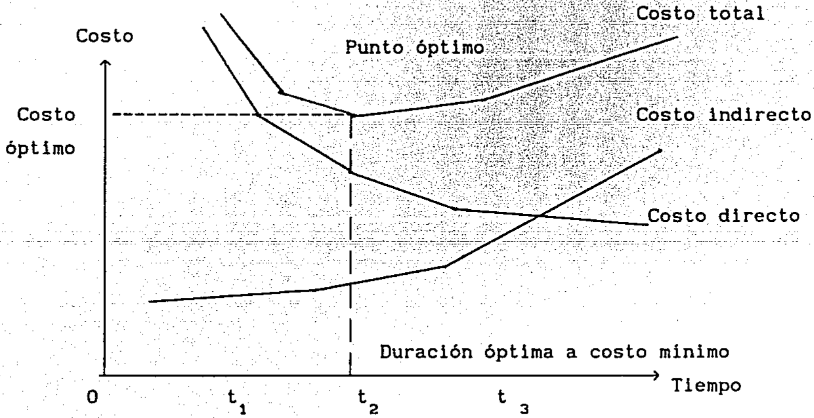
Cuando la relación costo - tiempo resulta no lineal, es decir en el caso de que la función sea convexa, se aproxima por una poligonal, donde la pendiente de los segmentos lineales que la forman es monótona decreciente, como se muestra en la siguiente figura:



La pendiente de cada una de las rectas asociadas a la relación costo - tiempo de cada actividad A_{ij} mide el costo por reducción unitaria de tiempo, es decir

$$P_{ij} = \text{Pendiente de la actividad} = A_{ij} = \frac{\bar{c}_{ij} - c_{ij}}{d_{ij} - \bar{d}_{ij}}$$

La suma de los costos directos y los costos indirectos se muestra en la siguiente figura:



Existen gran cantidad de soluciones no analiticas para este problema combinatorio, entre ellos se encuentran los métodos de solución de Teoria de Redes, ruta más corta y métodos heuristicos. El planteamiento del problema como un problema de programación lineal está muy ligado a la Teoria de Redes y enseguida se describe.

Planteamiento del problema de balanceo de línea como un problema de programación lineal

Otras técnicas de solución al problema de balanceo de línea, será las aplicables a la programación lineal, en donde el problema se podría seguir visualizando como una red en cuanto a su concepción, la diferencia fundamental estriba en que el planteamiento y la búsqueda de soluciones se hace mediante ecuaciones lineales, formadas por una función objetivo que se intenta optimizar y que está sujeta a una serie de restricciones.

Si se supone que la relación entre el costo y la duración de una actividad es lineal, entonces el problema de optimizar los costos, es un problema de *programación lineal.* Se supone que cada actividad tiene una serie de costos asociados, los costos directos que involucran costos de proceso, costo de material, costo de equipo, costos de recursos humanos y los costos indirectos que pueden ser costos por penalización e imprevistos.

Dado un conjunto finito T, una función f valuada en los reales positivos, definida sobre T, y un número c, encuentre una colección de subconjuntos de T $(S_1, S_2, \ldots, S_n)$ satisfaciendo las siguientes condiciones:

1) $\bigcup_{i=1}^{N} S_i = T$

2) $S_i \bigcap_{i \neq j} S_j = \phi$

3) $f(S) = \sum_{x \in S_i} f(x) \leq c$: $i=1,\ldots,N$

4) Si x precede a y, $x \in S_i$, $y \in S_j$, entonces $i \leq j$

5) $\left[Nc - \sum_{i=1}^{n} f(S) \right]$ a minimizar

T es el conjunto de tareas ($T_1, T_2, \ldots T_j, \ldots T_J$), las ordenes parciales sobre *T* son las relaciones de precedencia, la función *f* incide en los tiempos de ejecución [i.e., $t_j = f(T_j)$, $p_i = f(S_i)$ y c es el ciclo de tiempo. Se considera que los tiempos de ejecución de las tareas son aditivos, si se dá el caso de no aditividad para una larga ejecución, ésta puede ser subdividida en varias tareas más cortas con un tiempo más largo de ejecución total.

La restricción 1 asegura que todas las tareas deben ser asignadas a las estaciones, es decir, ninguna tarea queda exenta de participar en el problema. La restricción 2 implica que cada tarea debe ser asignada solo una vez. La restricción 3 manifiesta que el trabajo obtenido de cualquier estación singular no excederá el ciclo del tiempo. La restricción 4 señala que una tarea *x* precedida por una tarea *y* no puede ser ejecutada en una estación subsecuente a la estación en la cual *y* estuvo asignada. La última restricción es la función objetivo que se desea minimizar.

Planteada de esta manera la linea de producción es por configuración un problema de *secuencia de actividades* sin embargo existen diferencias fundamentales, dependiendo de la cantidad de productos que se procesen. Cuando se habla de balanceo de linea en un caso extremo se tiene la linea de producción singular (único producto), aunque también se pueden manejar N productos de manera simultánea.

Cuando una familia de productos, como en el caso de una línea de multiproductos, independientemente de que los productos sean distintos, ellos son tecnológicamente similares, requieren de diferentes cantidades de trabajo en algunas de las estaciones. En este caso se conoce el problema como un modelo de balanceo de ensamblado lineal mixto. Usando la mano de obra o sea el trabajo y el equipo eficientemente se minimiza el tiempo ocioso, para hacer esto, se selecciona un intervalo de tiempo T por ejemplo, un cambio que se hace a través del balanceo.

Considérese todas las unidades de los productos, de todos los modelos en la familia, los cuales han sido manufacturados durante el tiempo T. La estación i es una colección de tareas de todas estas unidades, asignadas a la estación conforme a las restricciones de precedencia y agregadas al tiempo de ejecución total P_i, el cual no debe exceder a T. Agrupando estas tareas en las estaciones se minimizaría

$$\sum_{i=1}^{N} (T - P_i)$$

El tiempo total ocioso a través de la línea.

Un modelo del problema de secuenciación mixta ha sido resuelto. Como se mencionó anteriormente, los varios productos o modelos, en la familia cargando las estaciones individuales diferentemente. Por lo tanto, una secuencia de los modelos de los productos ha sido encontrada la cual minimizará el tiempo ocioso en la estación y al mismo tiempo, prevendrá el congestionamiento.

En esta etapa el problema se puede visualizar como uno de calendarización enfocado a la secuencia de actividades en la cual cada unidad a ser producida cuenta como una actividad y cada estación es considerada una máquina. Obviamente, dada la manera en la cual la línea opera solo la permutación de los calendarios hace sentido y minimizando los lapsos de tiempo para todas las unidades a ser producida en el tiempo T, se minimizará como consecuencia el tiempo ocioso de todas las estaciones.

Un ejemplo distinto de la línea de multiproducto supone que los productos en la familia son manufacturados intermitentemente en colecciones o series de inventarios en lugar de que sobre la línea de producción se estén entremezclando. La línea se fija basada en las características del grupo tecnológico, lo cual significa que los productos son similares en diseño y en tecnología, mostrando solo diferencias en el tamaño o en otros elementos no esenciales.

Ejemplos de este tipo de líneas se encuentran en la fabricación de familias de partes de pistones, cabezas de cilindros, engranes y herramientas similares. Para cambiar la línea de un miembro de una familia a otro, se requiere también de una alteración en el costo, lo cual hace atractivo manufacturar por medio de series de producción .

En la producción en series, se propicia a considerarla como una línea de producción y no como un flujo de actividades. Esto es verdad porque las condiciones se dedican a sacar un producto cada vez. Tomando en cuenta lo anterior la línea de multiproducto con procesos en series aparece en operación como una secuencia de líneas de productos singulares. Si las facilidades se corrieran como actividades de flujo, los productos se moverían en series (mejor que unidad por unidad) entre operaciones (estaciones) y arriba de N diferentes series pueden ser presentados en el taller en cualquier momento dado. También la conversión de un producto al siguiente, será hecho con operaciones individuales y no necesariamente con el taller completo en un tiempo.

Si se tiene interés en determinar el tamaño del lote óptimo a ser secuenciado en una línea de multiproducto, se puede pensar de la línea como una máquina singular y aplicar uno de los modelos para múltiplesartículos, compartiendo el mismo equipo.

La intención de la discusión precedente es dejar claro porque las líneas de producción han ganado un estatus separado en cuanto a la literatura.

Thangavelu y Shetty recientemente han ofrecido un planteamiento del problema de ensamblado de línea con una formulación de programación entera, descrito de la siguiente manera:

Sean n tareas { $J_1, J_2, \ldots J_n$ } con tiempos de ejecución $p_1, p_2, \ldots p_n$, y un conjunto de relaciones de precedencia. Si J_u precede a J_v, entonces los subíndices serán ordenados tal que u < v. Sea

$$X_{ij} = \begin{cases} 1 \text{ si tarea } J_j \text{ se asigna a la estación } i; \quad i = (1,2,\ldots k) \\ 0 \text{ en cualquier otro caso.} \end{cases}$$

donde k es un límite superior del número de estaciones requerido. El tiempo total de todas las tareas asignado a la estación no debe exceder al ciclo del tiempo, ó bién:

$$\sum_{j=1}^{n} X_{ij}\, p_j \leq c, \qquad i = 1, 2, \ldots k$$

Se requiere que todas las tareas sean ejecutadas. Esto se expresa

$$\sum_{i=1}^{k} X_{ij} = 1, \quad J = 1,2,...n$$

Las relaciones de precedencia son expresadas por

$$X_{hv} \le \sum_{i=1}^{h} x_{iu}, \qquad h = 1,2,...k; \ (u,v) \in R \qquad ...3.4a$$

donde el conjunto $R = \{(u,v) \mid$ la tarea J_u inmediatamente precederá a la tarea J_v $\}$. Si existen R relaciones de precedencia inmediata. se necesitarán kR desigualdades de la forma en la ecuación 3.4a para contar las restricciones de precedencia.

El objetivo es minimizar el número de estaciones requerido. El número de estaciones no puede ser menor que el entero mayor que

$$\sum_{j=1}^{n} p_j / c$$

Sea k_0 denota el número más pequeño de estaciones teóricamente posible. Entonces la función objetivo es:

$$\min Z = \sum_{i=1}^{k} \sum_{j=1}^{n} C_{ij} \, x_{ij}$$

Donde:

$$C_{ij} = \begin{cases} p_j \left[\sum_{h\in F} p_h + 1 \right]^{(i-k_0-1)} \\ 0 \text{ en caso contrario} \end{cases}$$

y $F = \{j|$ la tarea J_j no necesita preceder a cualquier otra tarea $\}$ Esta función objetivo hace uso de mas de k_0 estaciones muy costosas y las tareas se presionan a ser asignadas a la estación más temprana posible en la línea. En efecto, una unidad de una asignación mas tardada es mas costosa que la suma de todas las asignaciones precedentes; puesto que al menos las primeras k_0 estaciones, deben ser usadas, ellas no necesitan que se les asigne un costo. Las tareas que no tienen seguidores necesitan tener costos positivos, pueden ser asignados después en la línea.

Note que la ecuación 3.4a puede ser reemplazada por:

$$\sum_{i=1}^{k} (k - i + 1)(X_{iu} - X_{iv}) \geq 0, \qquad (u, v) \in R; \quad \ldots 3.2b$$

porque cualquier solución satisfaciendo la ecuación cumple:

$$X_{iu} = \begin{cases} 1 \text{ para algûn } i = r \\ 0 \text{ en caso contario} \end{cases}$$

$$X_{iv} = \begin{cases} 1 \text{ para algûn } i = h \\ 0 \text{ en caso contrario} \end{cases}$$

Entonces (2.3a) y (2.3b) se satisfacen si y solo si $r \leq h$, lo cual implica que J_r precede a J_h.

El problema queda finalmente planteado de la siguiente manera:

$$\min Z = \sum_{i=1}^{k} \sum_{j=1}^{n} C_{ij} X_{ij}$$

Sujeto a :

$$\sum_{j=1}^{n} X_{ij} p_j \le c, \qquad i = 1,2,\ldots,k$$

$$1 - \sum_{i=1}^{k} X_{ij} \ge 0, \qquad j = 1,2,\ldots,n$$

$$-n + \sum_{i=1}^{k} \sum_{j=1}^{n} X_{ij} \ge 0$$

$$\sum_{i=1}^{k} (k - i + 1)(X_{iu} - X_{iv}) \le 0 \qquad (u,v) \in R$$

$$X_{ij} = 0$$

Se requieren en general, $(k + n + R + 1)$ restricciones.

La restricción $\sum_{i=1}^{k} X_{ij} = 1; \quad j = 1,2,\ldots n$ ha sido reemplazada por la segunda y tercera restricciones.

En muchas aplicaciones prácticas de programación entera las variables de decisión de la solución óptima se restringen a tomar valores enteros, cuando necesariamente la variable representa un articulo que por su estructura debe ser no divisible, es decir el valor no puede ser fraccionario. Ejemplos de la vida real que requieren algoritmos enteros son los diseños de ingeniería, problemas de localización, calendarización de tareas - recursos, calendarización de vuelos de aviación, problemas de transporte, problemas de asignación, el problema del agente viajero, problemas tipo mochila, problemas de inversión y una gran lista que puede aparecer en la industria, en la ciencia y en los áreas administrativas de gobierno.

Existe gran diferencia entre la programación entera y la programación lineal Básicamente enla programación lineal se maximiza o minimiza una función sobre una región de factibilidad convexa, mientras que en la programación entera se maximiza una función sobre una región de factibilidad que generalmente no es convexa. Trayendo esto como consecuencia que los problemas enteros tienen una mayor dificultad para resolverlos.

La programación lineal, ha encontrado métodos de solución que por su eficiencia se han convertido en clásicos, como es el método simplex; lo cual no ha sucedido con la programación lineal entera, por la dificultad que presenta el hecho de manejar la restricción de que las variables deben tomar solo valores 0 y 1. Sin embargo, el método simplex puede aplicarse con algunos ajustes como el de agregar otra restricción que garantice que cada variable sea menor o igual a uno y posteriormente en la solución se redondean al valor cero ó uno, aquellas variables que resultaron con valor fraccionario en la solución óptima.

Sin embargo, al aplicar el truco anterior, se presentan algunos riesgos, uno de los cuales es que la solución no necesariamente sea factible, ya que es dificil visualizar hacia cual extremo debe aplicarse el redondeo, otro riesgo se presenta al no existir garantía de que la solución redondeada, sea óptima para el problema cero - uno.

Aún cuando se puede caer en las aberraciones anteriores, existen algunos planteamientos para los cuales el enfoque no es del todo erróneo, ya que asegura que siempre se obtendrán soluciones con valores 0 ó 1. Enseguida se enuncian los siguientes teoremas, que justifican lo expuesto.

Teorema.- Sea A matriz m x n consistente de números enteros. Entonces los siguientes postulados son equivalentes.

i) A es totalmente unimodular

ii) Los puntos extremos de $S = \{ x \in R^n \mid Ax \leq b, x \geq 0 \}$ son enteros

iii) Toda submatriz no-singular de A tiene inversa entera

La matriz A se dice que es unimodular, si el determinante de cada submatriz cuadrada de A es 0, 1, -1. Esto significa que A es unimodular, si y solo si, cualquier submatriz es unimodular.

Considérese la solución a un problema de programación lineal. Si los coeficientes de la matriz A son unimodulares, entonces cualquier base B de A es también unimodular. También si x_B son las variables básicas, de $BX_B = b$, se tiene

$$X_B = B^{-1}b$$

De la regla de Cramer se tiene $B^{-1} = B^{*}/|B|$, donde B^{*} es la adjunta de B. Como B es unimodular, todos los elementos de B^{*} son 0, 1 ó -1 y $|B|$ es 1 ó -1. Entonces todos los elementos de B^{-1} son 0, 1 ó -1, de donde se tiene que X_B es entero, dado que se supueso que b era también entero. Por lo que cualquier solución básica factible y por consiguente la solución óptima de programación lineal, satisface los requerimientos de integrabilidad.

3.5 *Métodos de solución y ejemplos*

Enseguida se describen cuatro métodos heurísticos de solución.

Primer método heurístico

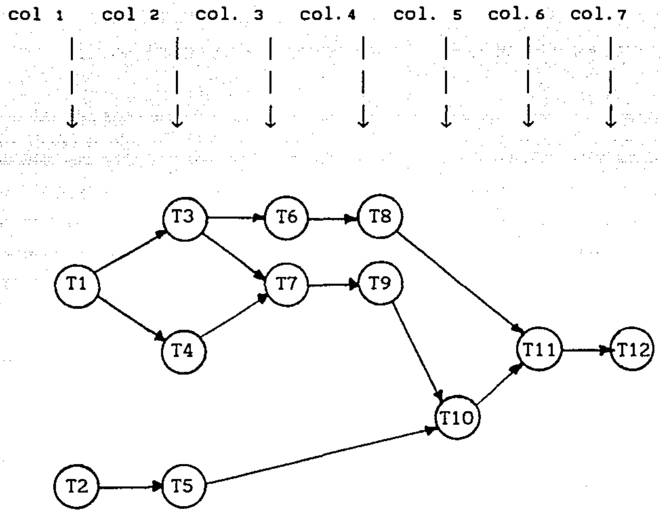
Para resolver el problema de líneas de ensamblado es necesario generar un balance que implique encontrar un orden ejecutable de tareas, llamado *secuencia factible* y después asignar las tareas en ese orden a las estaciones de trabajo con un tiempo total por estación no excediendo el ciclo de tiempo. Construyendo todas las secuencias factibles se garantizará que todos los posibles balances están enumerados y al final el mejor, en términos de tiempo ocioso mínimo, puede ser seleccionado. Este método computacionalmente hablando puede resultar impráctico, cuando existan un número muy grande de secuencias, por lo que es necesario eliminar de alguna manera temprana las secuencias no prometedoras.

Para ilustrarlo, se manejará el ejemplo de la figura 3.5a, en la cual se supone conocido un ciclo de tiempo igual a 12 y se desea tener un balance de la línea para el proceso de producción. Considere las siguientes dos reglas simples para construir la secuencia factible.

i) De entre las tareas calendarizables (entendiendo por ello, la tarea que aún no ha sido asignada y sus inmediatas predecesoras ya tienen posiciones en la línea de ensamblado) seleccionar la tarea con el número total más grande de sucesores.

ii) Enseguida se ordenan las tareas en función del tiempo de ejecución mas largo.

La esencia de la regla 1 es que sobre la marcha tiende a incrementar el número de tareas calendarizables (i.e. sucesores), ofreciendo una amplia gama para agrupar en las estaciones. La regla 2 intenta proporcionar más flexibilidad dejando tareas mas cortas hacia el final, cuando el tiempo residual en la estación decrece



digrama de precedencia para un proceso de producción.

Figura 3.5a

Se muestran enseguida las 12 tareas, el número total de sucesores, y las duraciones.

Tareas	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
Sucesores	9	4	7	5	3	3	4	2	3	2	1	0
Duración	6	9	4	5	4	2	3	7	3	1	10	1

Las reglas operan de la siguiente manera:

- Al inicio, las tareas T1 y T2 son calendarizables; T1 es seleccionada por regla 1.

- La lista calendarizable actualizada contendrá T2 y T3 y T4; donde T3 se seleccionó porque tiene el mayor número de sucesores.

- Enseguida las tareas T2, T4 y T6 con calendarizables; T4 está primero, porque tiene el número mas grande de sucesores, etc.

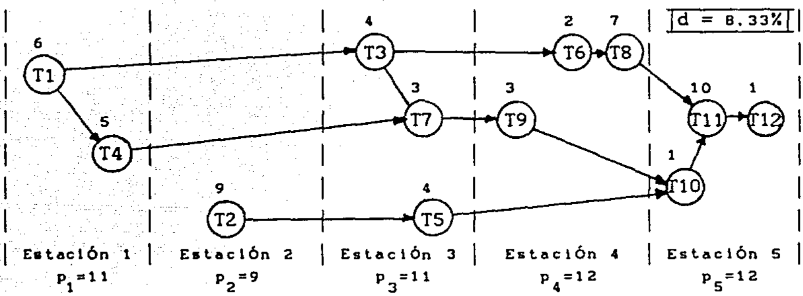
La secuencia resultante factible se muestra enseguida:

T1, T3, T4, T2, T7, T5, T9, T6, T8, T10, T11, T12

Observando la secuencia, se observa que las tareas están cargadas a las estaciones de trabajo arriba del límite del ciclo de tiempo.

T1, T3	T4	T2, T7	T5, T9, T6	T8, T10	T11, T12
Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6

Este procedimiento arroja una línea no óptima con una tardanza en el balance de 23.6%. Sin embargo un balance óptimo de cinco estaciones existe y es mostrado en la figura 3.5b



Balance óptimo de cinco estaciones de trabajo

Figura 3.5b

Mientras que la misma regla aplicada a otro problema, no puede alcanzar una solución óptima, también es verdad que resolviendo el problema anterior por otras reglas, se puede obtener otro balance. Tome por ejemplo las dos siguientes reglas, que sugieren otra heurística.

Segundo Método heurístico.- Extensión del primer método

Sean las siguientes dos reglas:

i) Del conjunto de tareas calendarizables, seleccione primero la tarea con el mayor número de sucesores inmediatos.

ii) Rómpase las ataduras de manera aleatoria.

Tareas	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
# de sucesores inmediatos	2	1	2	1	1	1	1	1	1	1	1	0

Con unos cuantos pasos se hará el *modus operandi* evidente:

- La primera lista calendarizable consiste de las tareas T1 y T2; con la regla 1, se selecciona T1.

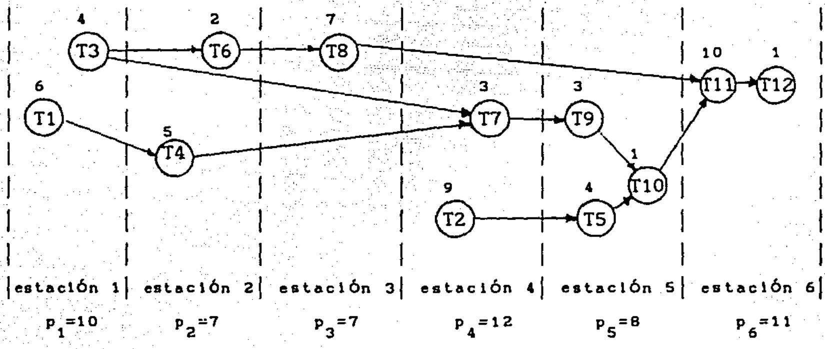
- El conjunto calendarizable a la fecha incluye T2 y T3 y T4, del cual T3 tiene el número mas grande de inmediatos sucesores y, por consiguiente, viene a ser el segundo en la secuencia.

- Ahora T2, T4, y T6 son calendarizables y por la regla 1 ellos estan en unión, un diseño variable T4, etc.

La secuencia factible generalizada es:

T1, T3, T4, T6, T8, T2, T7, T9, T5, T10, T11, T12

Entonces las tareas son asignadas a las estaciones de trabajo en ese orden. El resultado se representa en la Figura 3.5c. Recuérdese que esta técnica heurística encontró un balance óptimo con cinco estaciones para un ciclo de tiempo de 12.



Una linea de seis estaciones de trabajo

Figura 3.5c

La metodología que se utilizó en las dos heurísticas anteriores, básicamente consiste en ir construyendo primero las secuencias factibles y después asignando las tareas a las estaciones. Existe otra alternativa por medio de la cual la secuencia de tareas es creada y asignada, es otro método heurístico.

Tercer método heurístico

Los pasos son los siguientes:

i) Hacer la lista de tareas caledarizables.

ii) De la lista calendarizable, tomar las tareas, las cuales tengan duraciones individuales no excediendo del tiempo disponible en la estación, siendo asignada la tarea; llámese a estas tareas candidatos. Si no es posible encontrar ningún candidato, eso significa que la estación está llena y es necesario iniciar con otra.

iii) Seleccione un candidato al azar y asígnese a la estación corriente

iv) Repita los pasos del 1 al 3 hasta que todas las tareas hayan sido agotadas.

Se describirán enseguida algunas iteraciones:
Se inicia con la estación 1, el tiempo disponible es 12.

Lista calendarizable	duraciön	candidatos	Selección arbitraria	
T1	6	T1	T1–	estación 1
T2	9	T2		

Después de que T1 es asignado a la estación 1, el tiempo disponible se reduce a 6.

Lista calendarizable	duración	candidatos	Selecciön arbitraria	
T2	9	T3	T3–	estación 1
T3	4	T4		
T4	5			

Con ambos T1 y T3 en estación 1, el tiempo disponible es 2.

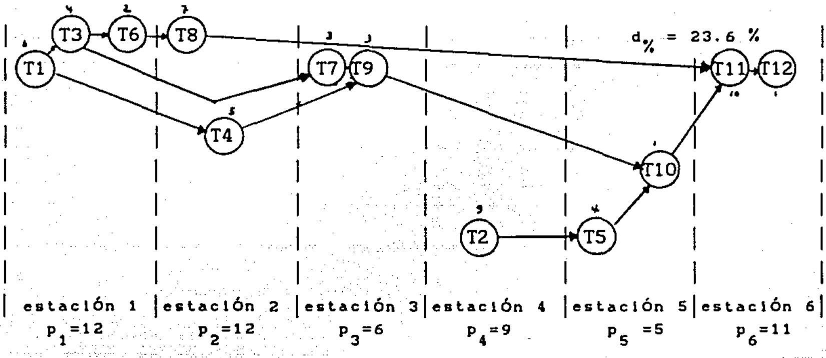
Lista calendarizable	duración	candidatos	Selecciön arbitraria	
T2	9	T6	T6–	estación 1
T4	5			
T6	2			

La estación 1 contiene T1, T3 y T6 está lleno, la estación 2 tiene tiempo disponible de 12.

Lista calendarizable	duración	candidatos	Selección arbitraria	
T2	9	T2	T8–	estación 2
T4	5	T4		
T8	7	T8		

Si el proceso de asignación es repetido de la misma manera, la secuencia factible que sigue es generada simultáneamente con el balance de seis estaciones mostrado en la figura 3.5d:

T1, T3, T6, T8, T4, T7, T9, T2, T5, T10, T11, T12



Línea de ensamblado con seis estaciones

Figura 3.5d

Las reglas de selección, éstas y las otras, pueden ser combinadas de varias maneras, algunas más complejas, pero más efectivas. Por ejemplo se pueden combinar de manera probabilística, dondelos tiempos se convierten en mecanismos aleatorios.

Cuarto método heurístico

Procedimiento de optimización hecho por Nevins. Este es un método bastante aceptado, es el método llamado del botón, enseguida se aplicará al ejemplo. El ciclo del tiempo es el mismo, c = 12.

Sean

D' = Suma de duraciones de todas las tareas.

D = Suma de duraciones de las tareas, las cuales han sido asignadas.

N' = Cota superior del número de estaciones permitido en cualquier balance.

s = Número de estaciones de trabajo ya generado.

En el ejemplo, D' = 55. Puesto que la solución al problema de cinco estaciones existe, la mejor búsqueda será producir una. Entonces se fija N' = 5.

Una búsqueda ó brote es representado como una lista (*s*, T1, T2,...,Tn), indicando las tareas que ya han sido asignadas a la primera estación *s* de la línea de producción. El problema asociado con esta búsqueda es asignar las J - n tareas a las N'- *s* estaciones.

Un brote ó búsqueda, llamado ⊖padre brote⊖ puede germinar brotes descendientes inmediatos, un número *m*, de nuevas tareas son agregadas al conjunto de tareas correspondientes al brote principal. Estas nuevas tareas constituyen una nueva estación de trabajo. Por ejemplo el brote A germina al brote B, puede observarse la figura 3.5e, agregando una nueva tarea, T2. La tarea T2 constituye una estación.

La figura 3.5e muestra que la búsqueda se inicia con el brote A, representado por una lista vacia, ninguna asignación se ha hecho aún. Construya una lista de todas las tareas calendarizables arregladas en orden de duración no creciente. Muévase hacia abajo de la lista viendo hacia una tarea, llamada A_1, cuya duración no excede al tiempo disponible en la estación, habiendo sido asignado el trabajo. Asigne A_1 a la estación. De aqui hasta que todos los brotes descentiente inmediatos son generados, se guardan tanto las tareas asignadas como las tareas calendarizables en una pequeña tabla llamada LISTA.

Para saber si alguna de las tareas ha sido asignada, se usará una notación especial. Por ejemplo, la primera tarea asignada a la estación fué denotada por A_1, entonces la notación será la tarea: A_1. Inserta todas las tareas que se hicieron calendarizables por la asignación de A_1 sobre la LISTA después de A_1, tal que la porción de la LISTA, después de A_1 permanece en orden de duración no creciente.

Continúe la LISTA hacia abajo, hasta el fin, agregando nuevas tareas A_2, A_3, ...,A_m a la estación de trabajo hasta que esté llena. Las tareas $A_1,A_2,...,A_m$ serán combinadas con las tareas en el brote raíz, probando que esta estación de trabajo no es un subconjunto de alguna otra estación de trabajo construída en esta iteración.

De donde:

i) Remueva de la estación generada la última tarea asignada A_m.

ii) Borre de LISTA los sucesores inmediatos de A_m.

iii) Localice sobre la LISTA la tarea A_m y continúe sobre la porción de la LISTA, después de que esta tarea ha tratado de generar una nueva estación de trabajo.

Si la LISTA después de la tarea A_m es vacía, repita los pasos del 1 al 3 con las tareas A_{m-1}, y si vuelve a suceder, repita con A_{m-2}, A_{m-3} etc. Si alcanzamos la tarea A_1 en la LISTA, cuando está vacía, ningunas estaciones pueden ser encontradas en esta etapa, esto es, los brotes raices no tendrán mas inmediatos descendientes.

Enseguida otro brote tiene que ser seleccionado para germinar en la siguiente iteración. Para hacer la selección, cualquier brote es asignado a una cuenta definida por el radio (D'-D) / (N'-s). La cuenta es simplemente el promedio del tiempo de proceso que se dejó para ser asignado a las estaciones de trabajo restantes. El brote no germinado con la menor cuenta es seleccionado como el nuevo raíz brote y el proceso es repetido iniciando otra vez con una lista de tareas calendarizables en orden de tiempo de ejecución no decreciente y así sucesivamente.

Dos puntos pueden probar la utilidad, durante la búsqueda.

a) Si $D'-D \leq c$ y $s < N$, la solución inmediata está a la mano, asigne todas las tareas restantes a una estación de trabajo.

b) Si $(D'-D)/(N'-s) > c$, ninguna solución puede ser obtenida a lo largo de la rama de búsqueda de árbol.

Se ilustrará el ejemplo con unas iteraciones. Para cualquier brote se debe tener la siguiente información: El símbolo del brote, su cuenta, y la lista de representación asociada.

El brote inicial A tiene una cuenta de (55-0)/(5-0) = 11. Encontrar sus inmediatos descendientes significa enumerar todas las estaciones factibles.

etapa 1

Lista ordenada calendarizable	duraciones	
T2	9	– Asigne T2 a la estación 1
T1	6	Inserte la tarea T5 a LISTA

Lista	duraciones	
T2: A_1	9	– Ninguna otra asignación posible
T1	6	
T5	4	

etapa 2

Remueva la tarea T2: A_1 de la estación 1. Remueva de LISTA los sucesores de la tarea T2, T5. La estación 1 está vacía.

Lista	duraciones	
T2		
↓T1	6↓	⟶ Fijar la porción de la lista, después de la tares T2. Asignar T1 a la estación 1. Inserte tareas T3 y T4 a la lista

Lista	Duraciones	
T2		
↓ T1: A_1	6↓	Asignar T4 a la estación 1
T4	5	⟶ Ninguna otra tarea resulta
T3	4	calendarizable.

Lista	Duraciones	
T2		
T1: A_1	6↓	
T4: A_2	5	⟶ Ninguna otra asignación posible sin el ciclo del tiempo. El brote C ha sido generado.
T3	4	

Etapa 3

Remueva la tarea T4: A_2 de la estación. Guarde T1 en la estación 1 y pruebe asignarlos a las otras tareas.

Lista	Duraciones	
T2		
T1: A_1	6	
T4		⟶ Fije la porción de la lista después de la tarea T4. Asigne T3 a la estación 1. Inserte la tarea T6 en la lista.
↓T3	4↓	

Lista	Duraciones	
T2		
T1: A_1	6	
T4		
T3: A_2	4↓	⟶ Asigne T6 a la estación 1 Inserte la tarea T8 en la lista.
T6	2	

Lista	Duraciones		
T2			
T1: A_1	6		
T4			
↓T3: A_2	4↓	⟶	Ninguna otra asignación posible
T6: A_3	2		El brote ha sido generado.
T8	7		

Etapa 4

Remueva la tarea T6: A_3 de la estación 1. Remueva de la lista, los sucesores de la tarea T6, T8. Guarde T1 y T3 en la estación 1 y pruebe asignarlo en otras tareas.

Lista	Duraciones		
T2			
T1: A_1	6		
T4			
T3: A_2	4	⟶	Fije la porción de la lista
T6			después de la tarea T6.
↓	↓		Está vacía.

Etapa 5

Remueva tareas T3: A_2 de la estación 1. Remueva de la lista las tareas sucesoras de T3, T6. Guarde T1 en la estación 1 y pruebe asignarlo a otras tareas.

Lista	Duraciones		
T2	9		
T1:A_1	6		
T4	5		
T3	4	⟶	Fije la porción de la lista, después de la tarea T3
↓	↓		está vacía.

Etapa 6

Remueva las tareas T1:A_1 de la estación 1. Remueva de la lista los sucesores de las tareas T1, T4 y T3. La estación 1 está vacía.

Lista	Duraciones		
T2	9		
T1	6		
↓	↓	⟶	Fije la porción de la lista, después de las tareas T1. Está vacía.

Ningún otra estación número 1, puede ser generada, tal que el brote A no puede hacer germinar ningún otro brote, que sea su descendiente inmediato.

En este punto el brote D, tiene la calificación más baja. Por consiguiente, se convierte en el brote padre para germinar, lo siguiente.

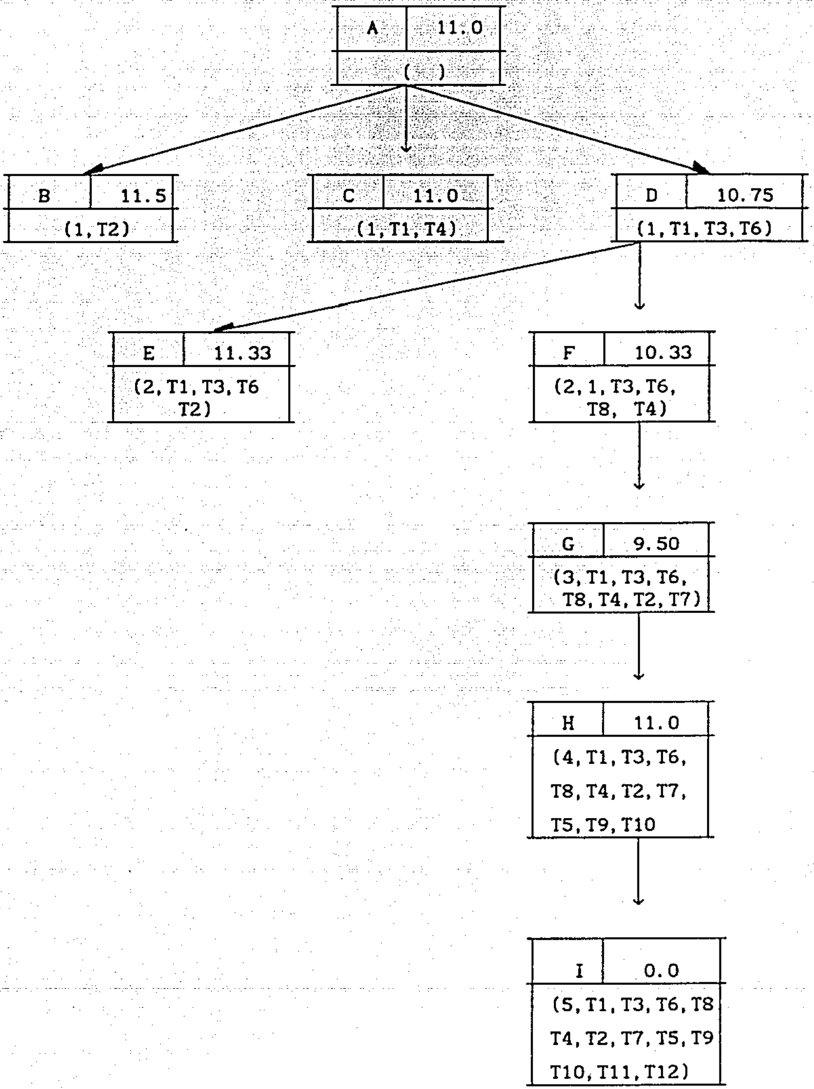
Etapa 7

Lista Ordenada Calendarizable	Duraciones		
T2	9		
T8	7	⟶	Asigne T2 a la estación 2
T4	5		Inserte las tareas T5 en la Lista, etc.

En la figura 3.5g se presenta el árbol de registro desarrollado, durante la ejecución del procedimiento. Se tiene una secuencia factible y un balance óptimo de cinco estaciones de trabajo que han sido generados:

T1 T3 T6	T8 T4	T2 T7	T5 T9 T10	T11 T12
estación 1	estación 2	estación 3	estación 4	estación 5
P1 = 12	P2 = 12	P3 = 12	P4 = 8	P5 = 11

Una comparación con el balanceo em Figura 3.5b muestra que la solución óptima no es necesariamente única.



Procedimiento de árbol de enumeración para mejor brote

Figura 3.5g

Definiciones:

Las siguientes definiciones son aplicables a todas las líneas de ensamblado.

1. *Producto ensamblado:* El producto que pasa a través de una secuencia de estaciones de trabajo donde las tareas son ejecutadas en el producto hasta que éste es completado en la estación de trabajo final. El término *producto ensamblado*, no implicará que el producto al llegar a la estación final sea un producto totalmente terminado. (El producto ensamblado final puede ser un subensamble de un producto principal).

2. *Elemento de trabajo*: Una parte del trabajo total contenido en un proceso de ensamblado. Se define N como el número total de elementos de trabajo requerido para completar el ensamblado y sea i el elemento de trabajo numerado con i en el proceso. Note que $1 \leq i \leq N$.

3. *Estación de trabajo* (WS)*:* Una localización sobre la línea de ensamblado donde un elemento o elementos de trabajo son ejecutados sobre el producto. El mínimo número de estaciones de trabajo, K, es mayor que o igual a 1.

4. *Ciclo de tiempo* (CT): El tiempo de acabado entre dos ensamblados sucesivos, se supone constante para todos los ensamblados para un transportado de velocidad dado. El mínimo valor del ciclo del tiempo puede ser mayor que ó igual a la estación con el tiempo más largo.

5. *Tiempo de la estación* (ST): Es la suma de los tiempos de los elementos de trabajo los cuales son ejecutados en la misma estación de trabajo. Es obvio que el tiempo de la estación (ST) no excederá al ciclo del tiempo (CT).

6. *Tiempo de demora de una estación*: Es la diferencia entre el ciclo del tiempo (CT) y el tiempo de la estación (ST) (El tiempo ocioso de la estación = CT - ST).

7. *Diagramas de precedencia*: Un diagrama que describe el orden en el cual los elementos de trabajo serán ejecutados. Algunas actividades no pueden ser ejecutadas hasta que sus actividades predecesoras hayan sido completadas. En efecto, el plan de las estaciones de trabajo a través de la línea de ensamblado depende del diagrama de precedencia.

CAPITULO IV

APLICACIONES

El mayor esfuerzo que se ha hecho en el trabajo de investigación del balanceo de línea es el que corresponde al caso determinístico, el cual supone que todos los tiempos de ejecución están fijos y son conocidos, y que las estaciones de trabajo no presentan paros en el proceso. Dada la estructura de los problemas de balanceo de líneas de ensamblado, se trata de un caso combinatorio, para cuyo análisis y búsqueda de soluciones, varias aproximaciones han sido investigadas, entre ellas el algoritmo de Balas.

El capítulo se desarrolla como sigue: En la primer etapa se describe la estructura matemática del algoritmo de Balas, enseguida se presenta un ejemplo numérico, resuelto con el algoritmo y con el programa de computadora, después se lista el programa de computación que utiliza el mismo algortimo de Balas para resolver un problema de programación entera binaria.

4.1 *Algoritmo de Balas*

Los métodos de solución a problemas que se presentan en todo Sistema Productivo, podrán enmarcarse en tres grandes bloques.

i) *Históricos*. Son los pioneros en el tema, representaron un avance en la teoría al ser descubiertos.

ii) *Prácticos*. Resuelven eficientemente ciertos problemas .

iii) *De vanguardia*. Representan el material que abrirá brecha en el futuro.

Dentro del segundo inciso, que se refiere a los métodos prácticos, podría hacerse una segunda clasificación de los métodos de solución de naturaleza entera: 1) *Planos de corte*, que no son eficientes para resolver problemas de tamaño modesto, pero que históricamente fueron los primeros en abrir brecha en ese campo. 2) Procedimientos de búsqueda de árbol óptimo como el *método de bifurcación y acotación* , que en la práctica aparentemente son los que se comportan mejor en la solución de problemas de dimensión intermedia, el método es elegante y simple, redondea y acota variables enteras, resultante de la solución de los problemas lineales correspondientes. 3) *Métodos de enumeración implícita*, son métodos heurísticos que trabajan adecuadamente para problemas de tipo binario (cero - uno), basados en la lógica, que como los métodos de bifurcación y acotación resuelven el problema entero sin tener que analizar todas las posibles alternativas 4) *Métodos de Teoria de Grupos* que requieren de un conocimiento profundo de álgebra moderna. 6) *Algoritmos de trayectoria más corta* que aplican la Teoria de Redes de Flujo 7) Casos especiales de *Programación lineal y de programación dinámica.*

El problema de programación lineal, anteriormente planteado se resolverá con el Algoritmo de Balas que a continuación se describe y que pertenece a los métodos de enumeración implicita que solucionan problemas de programación lineal entera de tipo binario.

Sean los vectores $u^t = (u_1, u_2, \ldots, u_n)$ y $x^t = (x_1, x_2, \ldots, x_n)$. Si una solución parcial consiste de las variables $x_{j1}, x_{j2}, \ldots, x_{js}$, asignados en ese orden, los componentes de x correspondiente a las variables asignadas tienen los valores 0 o 1. Las variables residuales en x correspondiendo a las variables libres tienen valores de -1. Los componentes de u son:

$$u_k = \begin{cases} j_k, & \text{si } x_{jk} = 1, \text{ su complemento no ha sido considerado} \\ -j_k, & \text{si } x_{jk} = 1 \text{ ó } 0, \text{ su complemento ha sido considerado} \\ 0, & \text{si } k > s \end{cases}$$

El algoritmo se describe de la siguiente manera:

Paso inicial

Primer paso:

Verifique si $b > 0$. Si es así, la solución óptima es $\hat{x} = 0$. O de otra manera $u^T = (0,0, \ldots ,0)$, $x^T = (-1,-1, \ldots ,-1)$, y $\hat{z} = \infty$, (un número suficientemente grande)

Pasos iterativos

Segundo paso:

Calcule

$$y_i = b_i - \sum_{j \notin J} a_{ij} x_{ij}$$

$$\bar{y} = \min y_i ; \quad i = 1,2,\ldots$$

Tercer paso

Crear un subconjunto T de las variables libres x_j, definida como

$$T = \{ j : z + c_j < \hat{z}, \; a_{ij} < o \text{ para } i \text{ tal que } y_i < o \}$$

Cuarto paso

Prueba de infactibilidad. Probar si existe un índice i tal que $y_i < 0$ y $y_i - \sum_{j \in T} \min (0, a_{ij})$. Si la respuesta es si, ir al sexto paso. De otra manera continuar.

Quinto paso

Prueba de ramificación de Balas. Para cada variable libre x_j crear el conjunto M_j,

$$M_j = \{i: y_i - a_{ij} < 0\}$$

Si todos los conjuntos M_j están vacíos, ir al sexto paso. De otra manera, calcular para cada variable libre x_j,

$$v_j = \sum_{i \in M_j} (y_i - a_{ij})$$

donde $v_j = 0$ si $M_j = \phi$. Agregue a la solución parcial la variable x_j que maximiza v_j. Regresar al paso dos.

Paso seis

a) Modifique el vector u, cambiando el signo de los componentes positivos. Todos los elementos de la derecha están fijados a cero. Esta es la nueva solución parcial. Regrese al paso dos.

b) Si no hay elementos positivos en u, la enumeración implícita es completa. La solución óptima es el vector corriente $\hat{x}$, y el valor de la función correspondiente es $\hat{z}$. Si $\hat{z}$ = infinito, no existe solución factible.

4.2 *Aplicación Concreta*

En muchas aplicaciones prácticas de programación entera las variables de decisión de la solución óptima se restringen a tomar valores enteros, cuando necesariamente la variable representa un artículo que por su estructura no puede ser divisible, es decir el valor no puede ser fraccionario. Ejemplos de la vida real que requieren algoritmos enteros son los diseños de ingeniería, problemas de localización, calendarización de tareas - recursos, calendarización de vuelos de aviación, el problema del agente viajero, problemas tipo mochila, problemas de inversión, reubicación y una gran lista que puede aparecer en la industria, en la ciencia y en los áreas administrativas de gobierno.

Ejemplo Numérico

Con el objeto de ilustrar la técnica desarrollas en este trabajo, se describe a continuación un ejemplo que permite mostrar todas las etapas del procedimiento mencionado; la idea de haber agregado un ejemplo numérico es dar a conocer las dificultades de cálculo y la necesidad de métodos especializados que permitan obtener una solución aproximada a los problemas de este género que se presentan usualmente en los Sistemas de Producción.

Para ilustrar el algoritmo, considere el siguiente caso, donde la función objetivo se plantea como sigue:

$$\text{Minimizar } z = 5x_1 + 7x_2 + 10x_3 + 3x_4 + x_5$$

sujeto a:

$$- x_1 + 3x_2 - 5x_3 - x_4 + 4x_5 \le -2$$

$$2x_1 - 6x_2 + 3x_3 + 2x_4 - 2x_5 \le 0$$

$$x_2 - 2x_2 + x_4 + x_5 \le -1$$

$$x_1, x_2, x_3, x_4, x_5 = 0 \text{ ó } 1$$

Iteración uno

El subconjunto T es $\{1,3,4\}$. La prueba de infactibilidad falla. Los conjuntos M_j y los valores para v_j para $j = 1,2,\dots,5$ son

$$\begin{aligned} M_1 &= \{1,2\}, & v_1 &= -4 \\ M_2 &= \{1,3\}, & v_2 &= -7 \\ M_3 &= \{2\}, & v_3 &= -3 \\ M_4 &= \{1,2,3\}, & v_4 &= -5 \\ M_5 &= \{1,3\}, & v_5 &= -8 \end{aligned}$$

$$\max_{1 \le j \le 5} v_j = v_3 = -3$$

La variable $x_3 = 1$ es agregada a la solución parcial. Otras variables son libres con sus variables fijas en cero.

Iteración dos

$T = \{2,5\}$. La prueba de infactibilidad falla

$$\begin{array}{ll} M_1 = \{2\}, & v_1 = -5 \\ M_2 = \{\phi\} & v_2 = 0 \\ M_4 = \{2\} & v_4 = -5 \\ M_5 = \{1,2\} & v_5 = -2 \end{array}$$

La siguiente variable agregada a la solución parcial es $x_2 = 1$, puesto que

$$\max v_j = 0, \qquad j = 1,2,3,5$$

Iteración tres

El vector $x^T = (0,1,1,0,0,)$ es factible y se convierte en una solución obligatoria $\hat{x}$ con $\hat{z} = 17$. El paso 6 es ejecutado. Se prueba que $x_3 = 1$, $x_2 = 0$ y las variables residuales son libres.

Iteración cuatro

$T = \{5\}$. La prueba de infactibilidad se satisface para $i = 2$ (la segunda restricción) y el paso seis (de detección de huellas) es tomado. Ahora x_2 y x_3 son fijadas a cero y las variables residuales son libres.

Iteración cinco

$T = \{1,4\}$. La prueba de infactibilidad es satisfecha para $i = 3$ y el paso de detección de huellas es tomada. La enumeración es completa puesto que no existen huellas posibles.

La óptima solución es :

$$\hat{x}_1 = 0, \quad \hat{x}_2 = 1, \quad \hat{x}_3 = 1, \quad \hat{x}_4 = 0 \quad \hat{x}_5$$

Con valor de la función objetivo:

$$\hat{z} = 17.$$

4.3 Resultados y Programa CALBAL

Se lista el Programa de computación CALBAL, que está programado en Lenguaje Turbo Pascal y minimiza un sistema de ecuaciones lineales enteras binarias, utilizando como método el algoritmo de Balas. El uso del paquete es de fácil acceso, con instruccines lógicas de seguir.

Enseguida se muestran los resultados del problema que se resolvió directamente con el algoritmo; comparándolos se comprueba que se obtiene el mismo valor en la función objetivo.

Ventajas y limitaciones de la Técnica

Los métodos exactos que estan disponibles para resolver los problemas de balanceo de linea, comúnmente dan solución a problemas de tamaño moderado.

No es sorprendente encontrar que en general. los algoritmos óptimos son computacionalmente infactibles cuando trabajan problemas reales con 100 tareas o más y 10 estaciones de trabajo. Esto explica porque la predominancia de los procedimientos heurísticos, pero estos tampoco pueden manejar problemas·de balanceo involucrando estaciones de trabajo múltiples, reorientación en las piezas de trabajo, trabajadores parciales a varias estaciones, etc.

El algoritmo de Balas es aproximadamente un 50 por ciento más rápido que el procedimiento de programación dinámica, parece ser que de los métodos de solución exactos disponibles es de los más eficientes y es capaz de resolver problemas con más de 50 actividades en pocos segundos de tiempo de computadora.

Los resultados comparativos entre los métodos de Programación Lineal ó Dinámica y en general los métodos exactos contra los algoritmos heurísticos; es que los primeros requieren sustancialmente mayor cantidad de tiempo de uso de computadora en problemas reales que involucran 100 tareas o más y los heurísticos aunque manejan menos variables de decisión son más prácticos, el uso de los métodos heurísticos es más frecuente por esa razón.

```
{     PROGRAMA QUE UTILIZA EL ALGORITMO DE BALAS, PARA RESOLVER UN
        PROBLEMA DE BALANCEO DE ENSAMBLADO DE LINEA, PLANTEADO CON
                  PROGRAMACION LINEAL ENTERA BOOLEANA.
}

PROGRAM PROGLIN(INPUT,OUTPUT);

USES
  CRT;

TYPE
  ARRMN = ARRAY[1..50,1..50] OF INTEGER;
  ARRM  = ARRAY[1..50] OF INTEGER;
  ARRN  = ARRAY[1..50] OF INTEGER;
  ARRN1 = ARRAY[1..50] OF INTEGER;

VAR
  I,J,M,N,INF : INTEGER;
  A           : ARRMN;
  B           : ARRM;
  C,X         : ARRN;
  FVAL        : INTEGER;
  EXIST       : BOOLEAN;


PROCEDURE BALAS(M,N,INF    : INTEGER;
                VAR A      : ARRMN;
                VAR B      : ARRM;
                VAR C,X    : ARRN;
                VAR FVAL   : INTEGER;
                VAR EXIST  : BOOLEAN);
```

```
LABEL 10,20;

VAR
   ALFA,BETA,GAMMA,I,J,MNR,NR,P,R,R1,R2,S,T,Z : INTEGER;
   Y,W,ZR : ARRM;
   II,JJ,XX : ARRN;
   KK : ARRN1;

BEGIN
   FOR I:= 1 TO M DO
   Y[I]:=B[I];
   Z:=1;
   FOR J:=1 TO N DO
     BEGIN
        XX[J]:=0;
        Z:=Z + C[J];
     END;
   FVAL:= Z + Z;
   S:=0;
   T:=0;
   Z:=0;
   KK[1]:=0;
   EXIST:=FALSE;
10:
   P:=0;
   MNR:=0;
   FOR I:=1 TO M DO
     BEGIN
       R:=Y[I];
       IF R < 0 THEN
         BEGIN
           P:=P + 1;
```

```
GAMMA:=0;
ALFA:=R;
BETA:= -INF;
FOR J:=1 TO N DO
  IF XX[J] <= 0 THEN
    IF C[J] + Z >= FVAL THEN
      BEGIN
        XX[J]:= 2;
        KK[S+1]:=KK[S+1] + 1;
        T:=T+1;
        JJ[T]:= J
      END
    ELSE
      BEGIN
        R1:= A[I,J];
        IF R1 < 0 THEN
          BEGIN
            ALFA:= ALFA-R1;
            GAMMA:= GAMMA + C[J];
            IF BETA < R1 THEN
              BETA:= R1;
          END;
      END;
IF ALFA < 0 THEN
  GOTO 20;
IF ALFA + BETA < 0 THEN
  BEGIN
    IF GAMMA+Z >= FVAL THEN
      GOTO 20;
    FOR J:= 1 TO N DO
      BEGIN
        R1:= A[I,J];
```

```
                R2:= XX[J];
                IF R1 < 0 THEN
                  IF R2 = 0 THEN
                    BEGIN
                      XX[J]:= -2;
                      FOR NR:= 1 TO MNR DO
                        BEGIN
                          ZR[NR]:= ZR[NR] - A[W[NR],J];
                          IF ZR[NR] < 0 THEN
                            GOTO 20
                        END
                    END
                ELSE
                  IF R2 < 0 THEN
                    BEGIN
                      ALFA:= ALFA - R1;
                      IF ALFA < 0 THEN
                        GOTO 20;
                      GAMMA:= GAMMA + C[J];
                      IF GAMMA + Z >= FVAL THEN
                        GOTO 20
                    END
              END;
            MNR:= MNR + 1;
            W[MNR]:= I;
            ZR[MNR]:= ALFA;
          END;
      END;
  END;
IF P= 0 THEN
  BEGIN
    FVAL:= Z;
```

```
    EXIST:= TRUE;
    FOR J:= 1 TO N DO
      IF XX[J]= 1 THEN
        X[J]:= 1
      ELSE
        X[J]:= 0;
    GOTO 20
  END;

IF MNR = 0 THEN
  BEGIN
    P:= 0;
    GAMMA:= -INF;
    FOR J:= 1 TO N DO
      IF XX[J]= 0 THEN
        BEGIN
          BETA:= 0;
          FOR I:= 1 TO M DO
            BEGIN
              R:= Y[I];
              R1:= A[I,J];
              IF R < R1 THEN
                BETA:= BETA + R-R1
            END;
          R:= C[J];
          IF (BETA > GAMMA) OR (BETA = GAMMA) AND (R < ALFA) THEN
            BEGIN
              ALFA:= R;
              GAMMA:= BETA;
              P:= J;
            END;
        END;
```

```
            IF P = 0 THEN
              GOTO 20;
            S:=S+1;
            KK[S+1]:= 0;
            T:=T + 1;
            JJ[T]:= P;
            II[S]:= 1; XX[P]:= 1;
            Z:=Z + C[P];
            FOR I:= 1 TO M DO
              Y[I]:= Y[I] - A[I,P]
        END
      ELSE
        BEGIN
          S:= S + 1;
          II[S]:= 0; KK[S + 1]:= 0;
          FOR J:= 1 TO N DO
            IF XX[J] < 0 THEN
              BEGIN
                T:= T + 1;
                JJ[T]:= J;
                II[S]:= II[S] - 1;
                Z:=Z + C[J];
                XX[J]:= 1;
                FOR I:= 1 TO M DO
                  Y[I]:= Y[I] - A[I,J]
              END;
        END;
      GOTO 10;
   20:
      FOR J:=1 TO N DO
        IF XX[J] < 0 THEN XX[J]:=0;
      IF S > 0 THEN
```

```
    REPEAT
      P:= T;
      T:= T - KK[S + 1];
      FOR J:= T + 1 TO P DO
        XX[JJ[J]]:= 0;
      P:= ABS(II[S]);
      KK[S]:= KK[S] + P;
      FOR J:= T - P + 1 TO T DO
        BEGIN
          P:=JJ[J];
          XX[P]:=2;
          Z:=Z-C[P];
          FOR I:=1 TO M DO
             Y[I]:= Y[I] + A[I,P];
        END;
      S:=S - 1;
    UNTIL (II[S+1] < 0) OR (S = 0);
    IF S > 0 THEN
      GOTO 10;
END;


(******************** PROGRAMA PRINCIPAL *********************)

BEGIN
   CLRSCR;
   WRITELN('  PROGRAMA QUE UTILIZA EL ALGORITMO DE BALAS, PARA RESOLVER UN');
   WRITELN('     PROBLEMA DE BALANCEO DE ENSAMBLADO DE LINEA, PLANTEADO CON');
   WRITELN('                     PROGRAMACION LINEAL ENTERA BOOLEANA.');
   WRITELN;
   WRITELN;
   WRITELN;
```

```
WRITELN('                    -----     ENTRADA DE DATOS     -----');
WRITELN;
WRITELN;
WRITE (' NUMERO (N) DE VARIABLES =    ');
READLN (N);
WRITELN;
WRITE (' NUMERO (M) DE RESTRICCIONES =    ');
READLN (M);
INF:=MAXINT;
WRITELN;
WRITELN (' MATRIZ DE COEFICIENTES DE LAS RESTRICCIONES ');
FOR I:=1 TO M DO
   FOR J:=1 TO N DO
      READLN (A[I,J]);
WRITELN;
WRITELN (' VECTOR DE LOS RECURSOS DISPONIBLES ');
FOR J:=1 TO M DO

   READLN (B[J]);
WRITELN;
WRITELN (' VECTOR DE COSTOS ');
FOR I:=1 TO N DO
   READLN (C[I]);
BALAS(M,N,INF,A,B,C,X,FVAL,EXIST);
CLRSCR;
WRITELN(' PROGRAMA QUE UTILIZA EL ALGORITMO DE BALAS, PARA RESOLVER UN');
WRITELN('   PROBLEMA DE BALANCEO DE ENSAMBLADO DE LINEA, PLANTEADO CON');
WRITELN('                PROGRAMACION LINEAL ENTERA BOOLEANA.');
WRITELN;
WRITELN;
WRITELN ('                       -----* SOLUCION *-----');
```

```
WRITELN;




WRITELN;
WRITELN ('              --- VECTOR DE LA SOLUCION OPTIMA (TRANSFORMADO)');
WRITELN;
FOR J:=1 TO N DO
   WRITELN ( '                X[',J,'] = ',X[J]);
WRITELN;
WRITELN;
WRITELN ('              --- VALOR OPTIMO DE LA FUNCION OBJETIVO = ',FVAL);
WRITELN;
WRITELN;
IF EXIST THEN
  WRITELN ('              --- EXITE SOLUCION OPTIMA')
ELSE
  WRITELN ('              --- NO EXISTE SOLUCION FACTIBLE');
GOTOXY(18,23);
WRITELN('PARA CONTINUAR PRESIONE <ENTER>');
READLN;
CLRSCR;
WRITELN ('              --- MATRIZ DE COEFICIENTES DE LAS RESTRICCIONES =');
WRITELN;
FOR I:=1 TO M DO
  BEGIN
    WRITE('            ');
    FOR J:=1 TO N DO
      WRITE (A[I,J]:6);
    WRITELN;
  END;
WRITELN;
WRITELN;
WRITELN ('              --- VECTOR DE TERMINOS INDEPENDIENTES =');
```

```
   WRITELN;
   FOR J:=1 TO M DO







      WRITELN ('                    b[',J,'] = ',B[J]);
   WRITELN;
   WRITELN;
   WRITELN ('             --- VECTOR DE COSTOS =');
   WRITELN;
   FOR I:=1 TO N DO
      WRITELN ('                    C[',I,'] = ',C[I]);
   REPEAT UNTIL KEYPRESSED;
END.
```

----- * SOLUCION * -----

--- VECTOR DE LA SOLUCION OPTIMA (TRANSFORMADO)

X [1] = 0

X [2] = 1

X [3] = 1

X [4] = 0

X [5] = 0

--- VALOR OPTIMO DE LA FUNCION OBJETIVO = 17

--- EXISTE SOLUCION OPTIMA

--- MATRIZ DE COEFICIENTES DE LAS RESTRICCIONES =

-1	3	-5	-1	4
2	-6	3	2	-2
0	1	-2	1	1

--- VECTOR DE TERMINOS INDEPENDIENTES =

b [1] = -2

b [2] = 0

b [3] = -1

--- VECTOR DE COSTOS =

C [1] = 5

C [2] = 7

C [3] = 10

C [4] = 3

C [5] = 1

V.- *CONCLUSIONES*

Los problemas de calendarización que se presentan en los Sistemas Productivos, se resuelven la mayoría de ellos de manera empírica, con sus respectivos resultados de carácter práctico que se ajusta a las necesidades; sin embargo, existen dentro de las técnicas de la Investigación de Operaciones en la parte matemática, la forma de modelar este tipo de problemas, entendiendo por ello, la conceptualización del problema y su formalización en un lenguaje algebráico.

En este sentido el presente trabajo proporciona una serie de pautas, métodos y técnicas, que norman un criterio para modelar problemas de calendarización. La ventaja que se obtiene al crear los modelos respecto a la parte práctica es la aplicación del método científico en la solución de los problemas..

Cabe mencionar que los factores humanos son susceptibles de representarse en los modelos matemáticos, como variables especiales. El objetivo es tratar de representar la realidad de la problemática en un modelo simbólico, ajustándolo y retroalimentándolo con la práctica, utilizando para ello, las diversas técnicas de sensibilidad y simulación conocidas.

La aportación de la presente Tesis, es la recopilación y ordenamiento de la información sobre el tema, proporcionando una serie de técnicas heurísticas y modelos matemáticos para los tres bloques en los cuales se dividió la Teoría de Calendarización. Se anexa un programa de computación que resuelve problemas de programación lineal entera binaria, utilizando el algoritmo de Balas.

BIBLIOGRAFIA

LIBROS:

- Mathematical Modeling with computers by S amuel L.S. Jacoby

- Syslo, Maciej M. Deo Narsingh y Kowalik, Janusz S.
Discrete Optimization Algoriths with Pascal Programs. Prentice Hall

- Discrete Optimization Algorithms with Pascal programs
Maciej M. Syslo, Narsingh Deo, Jaanusz S. Kowalik

-Production and Inventory Managemente de A. Neva, P.Hax Neva y George Candea. 1983.

- Operation Research in Production Planning Scheduling, and Inventory Control por Lynwood A. Johnson y Douglas C. Montgomery 1974.

- Vehicle Routing: Methods and Studies por Bruce Golden y Arjang Assad 1988.

- Foundation of Integer Programming por Harvey M Salkin y Kamlesh Mathur 1989.

- Introduction to Sequencing and scheduling por Kenneth R. Baker 1974

- Theory of Scheduling por Richard Conway, William Maxwell y Louis Miller 1967.

- Programaciön en Pascal de Peter Grogono 1984

- Sistemas Integrados de Control de Producción de David Bedworth y James Bailey 1988.

- Hacia una Comunicación Administrativa Integral de Sergio Flores de Gortari y Emiliano Orozco Gutiérrez.

- Métodos y Modelos de Investigación de Operaciones de Juan Prawda 1987.

- Investigación de Operaciones de Taha 1981.

- Guía de la Investigación Científica de Gamboa, Taboada y Dieterich.

ARTICULOS

- Sequencing en an assembly line with blocking to minimize cycle time McCormick, Pinedom Ahenker, Oper. Res. Vol 37, 6 de Nov. 1989.

- An efficient decision support system for academic course scheduling Oper. Res. Vol. 37, 6 de Nov de 1989.

- An algoritm for solving the job - shop problem Carlier and Pinson Managament Science Vol. 35 Febrero 1989.

- Solving large scale optimization problems by divide and conquer neural network by Simon Foo form Naval Research Laboratory code 5756

- An adaptable scheduling algoritm for flexible flow lines by Wittrock IBM Thomas Watson Research Center, Oper. Res. Vol. 36 Mayo, Junio 1988.