18

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "A C A T L Á N"

MODELO MATEMÁTICO PARA OPTIMIZAR EL PROCESO DE ASIGNACIÓN DE CARGAS DE TRABAJO A LAS FUENTES DE PRODUCCIÓN DE UNA EMPRESA

T E S I S

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

P R E S E N T A N

PATRICIA HERNÁNDEZ JUÁREZ
DANIEL MONTOYA JIMENEZ

ASESOR: FÍS. MANUEL VALADEZ RODRÍGUEZ.

SANTA CRUZ ACATLÁN, EDO. DE MÉX. SEPTIEMBRE DE 2002.

PAGINACION DISCONTINUA

AGRADECIMIENTOS

A mi Mamá, por todo el amor y la confianza que siempre me ha brindado y a la que debo todo.

A mi Padre y mis Hermanos, por brindarme su cariño y por creer en mí.

A mis Cuñadas y Sobrinos, por ser parte fundamental en la vida de mis hermanos.

A mis grandes amigos: Vero, Hilda, Gisela y Oscar, por su apoyo y amistad.

A Daniel, por ser un gran compañero y amigo, y por brindarme la oportunidad de conocerle durante el desarrollo de este trabajo.

A mi asesor, Manuel Valadez Rodríguez, por su paciencia y por su gran aportación en este trabajo.

A Jaime, por llegar a mi vida y brindarme su amor y apoyo incondicional. T.Q.M.

A todas las personas con las que he convivido durante todo mi desarrollo escolar y profesional les doy las gracias, porque de todas he aprendido.

Patricia Hernández Juárez.

AGRADECIMIENTOS

A mis padres, Rosa Jiménez Paredes y Albino Montoya Vega por todo el apoyo, cariño y compresión que siempre me han dado.

A mis hermanos, Angélica y Luis, a mis Abuelitos, Lola, Pachita, Toño y Julio, y a toda mi familia por estar o haber estado cerca de mí.

A mi asesor, Manuel Valadez Rodríguez por sus excelentes aportaciones al presente trabajo.

A Patricia Hernández Juárez por ser una gran compañera y amiga.

A Nicolás Gutiérrez Chávez por su apoyo constante en mi superación personal y profesional.

A todos mis compañeros y amigos del CCH Naucalpan y de la ENEP Acatlán.

A cada uno de mis Profesores, por su labor y dedicación.

Daniel Montoya Jiménez

ÍNDICE

INTRODUCCIÓN.

Debido a la enorme cantidad de beneficios que representa la automatización de procesos en los diversos campos del quehacer humano, esta actividad se ha incrementado de manera sustancial en los últimos años hasta llegar a convertirse actualmente en un hito esencial. En particular, el hablar de procesos computarizados basados en modelos matemáticos es hacer referencia a una serie de conocimientos que se fundamentan en dos áreas: Matemáticas y Computación. El dominio de este campo de investigación redunda en grandes aportaciones a la apertura de la implantación de sistemas basados en modelos que permitan garantizar los mayores beneficios.

El presente proyecto se basa en la aplicación de un modelo matemático para realizar la asignación de actividades a las fuentes de producción de una empresa. Para la construcción del modelo se consideraron como ejes centrales los Métodos de Optimización, los cuales proporcionan los mejores elementos para su estructuración. Por otro lado se realiza la automatización del algoritmo para su aplicación.

El estudio dio inicio en una empresa que pretende automatizar el proceso de planeación de su producción; la situación por la que atraviesa es estable en la medida de que todavía se puede controlar manualmente esta actividad, pero cuando se tenga variabilidad en la producción se corre el riesgo de tener pérdidas. En la empresa existe el Departamento de Planeación de la Producción, quien se encarga de establecer la calendarización de actividades, ésta se realiza mes a mes y generalmente cada semana se realizan ciertos ajustes por incrementos o cambios en la demanda de los productos. Las metas trazadas en este departamento se obstaculizan debido a las restricciones de tiempo y equipo que se tienen para llevar a cabo la asignación de recursos. Por tal motivo, se proyecta implantar la automatización del proceso de asignación en la empresa ***Metales y Contactos*** de tal manera que se logren obtener los mayores beneficios con un mínimo de costos.

El primer capítulo, ***Métodos de Optimización***, presenta una descripción más o menos detallada de los métodos teóricos existentes que son requeridos para la estructuración del Modelo y, en el segundo, ***Propuesta de un Modelo de Asignación***, se propone un modelo general para que pueda ser implantado en cualquier organización que tenga necesidades semejantes a la empresa que nos ocupa.

El tercer capítulo, ***Caso de Estudio***, presenta la descripción de la organización así como la aplicación particular del Modelo Matemático. Por último, en el capítulo cuatro ***Automatización del Modelo***, se presentan la estructura del sistema y los puntos más destacables de la programación.

1. MODELOS DE OPTIMIZACIÓN.

La finalidad de este capítulo, es la de dar a conocer los elementos teóricos recopilados por los autores del presente trabajo durante el proceso de búsqueda del modelo de asignación. Los modelos estudiados fueron los de optimización, debido a que a esta clasificación pertenecen los modelos de asignación.

La optimización es un área dedicada, en parte, a ofrecer alternativas para la toma de decisiones y el modelado de sistemas determinísticos o probabilísticos que corresponden a dicha área y que se originan por diversas situaciones, nacen, en gran parte, por la necesidad de asignar recursos limitados.

El objetivo de la optimización es encontrar la solución óptima de un problema. Para ello, es necesario tratar de describirlo mediante un modelo matemático, ya que es así como se puede obtener una representación concisa de él. Pero así como un modelo puede facilitar la solución de un problema determinado, también puede tornarse un tanto complicado al construirlo ya que, en ocasiones, puede ser difícil abstraer todo el entorno de dicho problema, más aun cuando éste es muy complejo.

Por regla general, los modelos matemáticos representan sólo aproximaciones de las situaciones reales y su validez o utilidad está relacionada precisamente con el grado de aproximación.

1.1 Modelos determinísticos.

Los modelos determinísticos son aquellos que hacen alusión a lo que se puede predecir con certeza, pocas situaciones de la vida real tienen esta característica, pero vale la pena aclarar que en muchas ocasiones se hace este supuesto para manejar modelos más sencillos. Pertenecen a esta clasificación los modelos de Programación Matemática[1].

Definición. Una función que se establece a fin de describir matemáticamente un problema determinado y cuyo objetivo es minimizar costos o maximizar ganancias se conoce como ***función objetivo*** y a las funciones que limitan la selección de los valores de las variables que aparecen en ésta se les llama ***restricciones***. El conjunto de valores de las variables que satisface todas las restricciones es llamado ***conjunto factible***. Un modelo de optimización que considera todos estos elementos es conocido como ***Modelo de Programación Matemática***.

La Programación Matemática comprende tres casos especiales:

- Programación Clásica.
- Programación Lineal.
- Programación no Lineal.

[1] En este caso el término programación se define como un sinónimo de planeación y no tiene nada que ver con programación en computadora.

1.1.1 Programación Clásica.

La Programación Clásica comprende dos ramas importantes: optimización sin restricciones y optimización con restricciones.

Definición. En los modelos de programación matemática, a las funciones que dependen de una sola variable se les llama ***funciones univariadas*** y a las que dependen de dos o más variables se les conoce como ***funciones multivariadas***.

Definición. A un valor mínimo o máximo que alcanza una función univariada o multivariada, en una región determinada, se le conoce como ***óptimo (local)***.

Para la obtención de un óptimo en el ***caso no sujeto a restricciones*** para una función con una sola variable se plantea un modelo que consiste en determinar la solución a través de una pequeña variación de x^*, siendo $(x^*, f(x^*))$ un punto que pertenece al conjunto factible. Si $f(x^*)$ es un máximo local interior (máximo valor dentro del conjunto factible), entonces, para todos los puntos vecinos $(x^*+\Delta x, f(x^*+\Delta x))$ siendo Δx una pequeña variación arbitraria de x, en general, se tiene que

$$f(x^*) \geq f(x^*+\Delta x)$$

Supóngase que f es diferenciable en $x=x_0$, que $f'(x_0)=0$ y que $f''(x_0)$ existe,

a) Si $f''(x_0) < 0$, entonces f tiene un máximo local en x_0.

b) Si $f''(x_0) > 0$, entonces f tiene un mínimo local en x_0.

El caso de las funciones escalares de dos o más variables, puede tratarse en forma análoga.

Tanto para el caso univariado como para el multivariado existen diferentes métodos para la obtención del óptimo de una función no sujeta a restricciones. Por mencionar algunos, para el caso univariado se encuentran el método de Fibonacci y el método de la Sección de Oro, éstos son aplicables a funciones diferenciables y no diferenciables, el método de Newton Rapson, requiere una función continua y diferenciable para hallar un óptimo. Para el caso multivariado están algunos como el método de Ascenso o Descenso Acelerado, de Fletcher-Powell, de Fletcher-Reeves y de Direcciones Conjugadas, éstos son métodos de gradiente y se aplican a funciones diferenciables. Cada uno de estos métodos tiene sus características muy particulares y debe ser utilizado de acuerdo al problema en cuestión.

La ***optimización con restricciones*** se define a través de una función continua conocida como función objetivo y un conjunto de restricciones que son funciones dependientes de las variables que incluye el modelo.

El estudio de la optimización con restricciones, se basa en las condiciones necesarias y suficientes que se satisfacen en los puntos óptimos. Estas condiciones, además de su valor intrínseco para interpretar soluciones, definen los Multiplicadores de Lagrange y una matriz Hessiana que, juntos, conforman la base del análisis y desarrollo de algunos métodos existentes. Esto no es aplicable para funciones discretas, solo para funciones diferenciables.

Sea f la función objetivo y ϕ_i para i=1, 2, ..., n las funciones de restricción que incluye el modelo, de tal manera que

$$\begin{aligned}
&\mu = f(x_1, x_2, \dots, x_n) \\
&\text{sujeta a} \\
&\phi_1(x_1, x_2, \dots, x_n) \le 0 \\
&\phi_2(x_1, x_2, \dots, x_n) \le 0 \\
&\vdots \\
&\phi_m(x_1, x_2, \dots, x_n) \le 0
\end{aligned}$$

mediante el método de Multiplicadores de Lagrange, que es uno de los métodos clásicos para resolver este tipo de problemas, se obtiene la función F.

$$F = f + \lambda_1\phi_1 + \lambda_2\phi_2 + \dots + \lambda_m\phi_m \tag{1}$$

A la ecuación (1) se le llama función lagrangiana o lagrangiano. La ventaja de esta función es que ahora se tiene una función sin restricciones, aunque con más variables. La forma de resolverla es por medio de uno de los métodos ya mencionados, en la optimización no sujeta a restricciones.

La importancia de los valores λ es que brindan información valiosa sobre la sensibilidad del valor óptimo de la función objetivo respecto a los cambios en las constantes de restricción.

1.1.2 Programación Lineal.

La Programación Lineal trata con la clase de modelos de Programación Matemática que se ocupan de la asignación eficiente de recursos limitados en actividades competitivas, con el objeto de satisfacer las metas deseadas tales como maximizar beneficios o minimizar costos, es decir, optimizar.

El modelo de ***Programación Lineal*** consta de una función objetivo lineal y de un conjunto de restricciones con igualdades y desigualdades lineales:

$$
\begin{aligned}
&\text{Optimizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \\
&\text{sujeto a} \\
&\qquad a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \le b_1 \\
&\qquad a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \le b_2 \\
&\qquad \vdots \\
&\qquad a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \le b_m \\
&\qquad x_i \ge 0, \text{ para } i = 1,2,\dots,n
\end{aligned}
$$

Debido a que los modelos de Programación Lineal se presentan en gran variedad de formas[2], es necesario ajustar el modelo a una ***forma estándar***. Para lograrlo es necesario cambiar las restricciones de desigualdad a ecuaciones, introduciendo (sumando o restando) variables no negativas. Estas variables se conocen como variables de holgura y se suman si la restricción es menor o igual, o como variables de exceso y se restan si la restricción es mayor o igual.

[2] Las restricciones también pueden llevar los signos de $<$, $>$, $\ge$ ó $=$.

La forma estándar es muy importante debido a que es allí donde se puede emplear el método Simplex, que es un algoritmo de solución que sirve para encontrar el óptimo en un modelo de programación lineal.

Una vez que se tiene el planteamiento en la forma estándar, las ecuaciones son expresadas en forma de tabla. El método Simplex comienza eligiendo una solución factible básica inicial, la cual estará integrada por las variables de holgura, pero esto sólo será posible cuando las restricciones del planteamiento original sean de menor o igual porque de lo contrario no es sencillo encontrar una solución inicial básica factible. En estos casos se utiliza la técnica de variables artificiales, ésta consiste en introducir una variable llamada artificial en las restricciones con la finalidad de que sea la variable básica inicial para esa ecuación.

La adición de las variables artificiales hace que se infrinjan las restricciones correspondientes, esta dificultad se elimina asegurando que las variables artificiales sean cero en la solución final, para ello se les asignará un valor numérico muy grande en la función objetivo, tal valor se designará como $-M$ para modelos de maximización y $+M$ para modelos de minimización con M mayor que cero.

Dos modelos característicos e importantes de la programación lineal son: el de transporte y el de asignación.

El objetivo del ***modelo de Transporte*** es minimizar el costo de transportar una mercancía desde un cierto número de orígenes a varios destinos. Como ocurre generalmente en los modelos de programación lineal, en el de transporte es la estructura y no el contexto lo que lo hace útil.

El modelo de Transporte se aplica con frecuencia en distintos contextos y tiende a hacer uso de un número muy grande de variables y restricciones por lo que la aplicación directa del método Simplex representa un esfuerzo computacional extraordinario. Sin embargo, ha sido posible desarrollar versiones simplificadas del método Simplex, que logran un enorme ahorro de recursos al explotar la estructura especial del modelo haciendo más eficiente el algoritmo para llegar a su solución.

Otro tipo de modelo especial de la programación lineal, es el ***modelo de Asignación***, en el que los recursos se asignan a las actividades sobre una base de uno a uno, es decir, cada recurso deberá asignarse a una actividad en particular y cada actividad contará con sólo un recurso.

El objetivo de este modelo es determinar como deben hacerse las asignaciones con el fin de minimizar los costos totales. La única condición, y muy importante, para que se pueda aplicar este modelo es que la oferta sea igual a la demanda, es decir, se necesita que el modelo esté ***equilibrado*** ya que de esta manera se garantiza que cada recurso esté asignado sólo a una actividad.

La representación del modelo de asignación es la siguiente:

$$\text{optimizar} \quad z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

sujeto a :

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,2,\dots n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,2,\dots n$$

$x_{ij} = 0$ si el i - ésimo recurso no se le asigna a la j - ésima actividad.

$x_{ij} = 1$ si el i - ésimo recurso se asigna a la j - ésima actividad.

c_{ij} = Representa el costo de asignar el recurso i a la actividad j.

Si un recurso no se puede asignar a una actividad, la c_{ij} correspondiente toma el valor de M, que es un valor exorbitante.

Las variables x_{ij} en este modelo son llamadas binarias debido a que representan decisiones de "si" o "no", éstas son representadas por los valores 1 y 0 respectivamente.

1.1.3 Programación No Lineal.

El modelo de programación no lineal consiste en elegir valores no negativos para las variables de modo tal que se maximice o minimice una función no lineal sujeta a un conjunto de restricciones de desigualdad. Las variables citadas son los instrumentos reunidos en un vector, la función es conocida como función objetivo, la cual se denota por f donde f: $R^n \Rightarrow R$. Las funciones que integran las restricciones también son agrupadas a través de un vector al igual que las constantes. Las funciones se piden continuamente diferenciables y no contienen elementos aleatorios; las constantes son números reales.

De una manera general, el Modelo de Programación No Lineal consiste en encontrar $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, que optimice la función objetivo $f(\mathbf{x})$. Las restricciones son de dos tipos:

1. Restricciones de no negatividad: si $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, entonces $x_i \geq 0$ para toda i=1, 2, ..., n.

2. Restricciones de desigualdad

$$
\begin{gathered}
g_1(x_1, x_2, \ldots, x_n) \leq b_1 \\
g_2(x_1, x_2, \ldots, x_n) \leq b_2 \\
\vdots \\
g_m(x_1, x_2, \ldots, x_n) \leq b_m
\end{gathered}
$$

Por tanto el modelo general de un problema de programación no lineal se puede escribir de la siguiente forma:

$$
\begin{aligned}
&\text{Optimizar } f(x_1, x_2, \ldots, x_n)\\
&\text{s.a.}\\
&g_1(x_1, x_2, \ldots, x_n) \leq b_1\\
&g_2(x_1, x_2, \ldots, x_n) \leq b_2\\
&\qquad\vdots\\
&g_m(x_1, x_2, \ldots, x_n) \leq b_m\\
&x_i \geq 0 \quad \text{para toda } i = 1,2,\ldots, n
\end{aligned}
$$

1.2 Acercando los modelos a la realidad.

Cuando en los modelos de optimización los datos de entrada presentan variaciones aleatorias, el modelo se puede complicar, desafortunadamente, en la realidad, en muchas ocasiones, los fenómenos presentan éste comportamiento. Al hablar de un modelo matemático que puede aproximar la representación del comportamiento aleatorio de un fenómeno, estamos haciendo referencia a un modelo que contempla las reglas de la Teoría de Probabilidades.

Una disputa entre jugadores en 1654 llevó a dos famosos matemáticos franceses, Blaise Pascal y Pierre de Fermat, a la fundación del cálculo de probabilidades. Al igual que ha ocurrido con otras muchas ramas de la matemática, el desarrollo del cálculo de probabilidades ha sido estimulado por la enorme variedad de sus aplicaciones.[3]

Los ***modelos probabilísticos*** son aquellos que involucran un comportamiento aleatorio en su estructura. Cuando la solución a un problema no se puede predecir con certeza, resulta muy útil echar mano de ellos.

1.2.1 Probabilidad teórica.

El primer paso para representar la naturaleza de un conjunto de datos primarios es calcular su media y varianza. La ***media*** o valor promedio es una representación de la tendencia

[3] Nota tomada del libro **Cálculo con funciones de varias variables, Tom M. Apostol.**

central de los datos, en tanto que la ***varianza*** es una medida de dispersión o de variación aleatoria alrededor de la media.

Los experimentos se clasifican en determinísticos y aleatorios. Los primeros son aquellos en los cuales los resultados en igualdad de condiciones ocurren siempre de la misma manera y los segundos, son los que no ocurren siempre de la misma forma, es decir, su comportamiento se rige por el azar. La teoría de probabilidad está enfocada al estudio de los experimentos aleatorios.

Definición: Se le denomina ***experimento aleatorio*** a un proceso de observación o medición de un fenómeno aleatorio. Dicho experimento se utiliza en la teoría de probabilidades para describir cualquier proceso cuyo resultado no se conoce de antemano con certeza. A los posibles resultados de un experimento aleatorio se les conoce como ***eventos*** y el conjunto de todos los resultados posibles se denomina ***espacio muestral.***

Existe una primera clasificación para el espacio muestral: discreto y continuo. El espacio se llama discreto solamente si contiene un número finito o infinito de puntos que se pueden colocar en una sucesión simple $E_1, E_2, \ldots$ Por otro lado, el espacio se dice continuo cuando sus resultados caen en cualquier punto de un intervalo de números reales.

Definición: Las constantes que determinan la forma específica de una ley de probabilidades (F) se denominan ***parámetros.*** Donde $F: R \Rightarrow [0,1]$.

Los parámetros más comunes son la media y la varianza.

El concepto de probabilidad es importante para tener en cuenta la predicción de eventos a través de los parámetros correspondientes.

Definición: La probabilidad P[A] de cualquier evento A es la suma de las probabilidades de todos los puntos muestrales en él. La probabilidad de todo el espacio muestral es la unidad. Se sigue que, para cualquier evento A,

$$0 \leq P[A] \leq 1.$$

Definición: A una función definida en un espacio muestral se le llama ***variable aleatoria.***

Todas las variables aleatorias son funciones de la variable x. Por lo tanto se pueden expresar todas las probabilidades mediante la función de distribución:

$$F(x) = P[X \leq x] \qquad -\infty < x < \infty.$$

La relación entre la función de punto $F(\cdot)$ y la función de conjunto P [ ] se indica mediante

$$F(x) = P(-\infty, x], \qquad P[a, b] = F(b) - F(a).$$

Definición: Una función de punto F en la recta real es una ***función de distribución de probabilidades*** de la variable aleatoria X si,

(i) F es no decreciente, o sea que $a < b$ implica $F(a) \leq F(b)$.

(ii) F es continua por la derecha, o sea que $F(a)=F(a+)$ y el límite de $F(x)$ es igual a $F(a+)$ cuando x tiende $a+$.

(iii) $F(-\infty) = 0$ y $F(\infty) = 1$,

Existe una gran variedad de distribuciones de probabilidad llamadas ***teóricas*** las cuales son expresiones matemáticas que representan las características de los posibles resultados de un experimento. De acuerdo al comportamiento que presentan las variables aleatorias, en la mayoría de los casos éstas se pueden ajustar a alguna de las distribuciones de probabilidad teórica para conocer la probabilidad y el valor de los parámetros.

1.2.2 Probabilidad empírica.

Cuando un fenómeno presenta un comportamiento aleatorio, para su estudio, se busca una distribución de probabilidad ***teórica***, que se ajuste a los datos, sin embargo algunas veces esto no es posible debido a que existen casos que tienen propiedades únicas y tal vez los datos están distribuidos de manera que no existe una formulación que aproxime el comportamiento de éstos.

Para tener una interpretación descriptiva de los datos se deben usar los histogramas de frecuencia, la curva que se representa mediante éstos se conoce como ***distribución empírica.*** Cuando se habla de distribuciones empíricas, se hace referencia a aquellas que no han sido previamente descritas por una expresión matemática.

Mediante una prueba estadística, llamada de bondad de ajuste, es posible saber a que distribución teórica se adecua la curva del histograma de frecuencias. La prueba más utilizada es la ji cuadrada que se basa en comparar el histograma de frecuencia con las distribuciones de probabilidad teórica.

1.3 Casos especiales.

Cuando la aplicación de los diferentes tipos de modelos matemáticos explicados anteriormente no es suficiente para obtener una buena aproximación de la solución a un problema determinado, existen alternativas que pueden mejorar dicha aproximación. Cabe señalar que cada una de ellas implica realizar una formulación diferente, sin embargo, resulta interesante ajustar el modelo en busca de un mejor resultado. A estas alternativas les llamaremos ***Casos Especiales.*** Enseguida se describen brevemente algunos de estos; en particular el modelo de ***Sistemas Dinámicos***, el modelo de ***Inventarios*** y el modelo basado en las ***Cadenas de Markov.*** Estos tres se relacionan de alguna manera con el problema bajo estudio.

1.3.1 Control de inventario.

El primer caso especial está basado en la teoría de inventarios que consiste en la formulación de un modelo matemático que describe el comportamiento de un sistema basado en criterios de almacenamiento. Posteriormente se deriva una política óptima de inventario respecto al modelo y por último se hace uso de una computadora para mantener un registro de los niveles de inventario y señalar cuando y cuanto conviene reabastecer. Es por ello que se tiene una relación con el problema bajo estudio, porque en dicho problema es necesario tener un control de las unidades de producción que se consideran para almacenaje.

Dentro de una organización la política de inventario afecta el rendimiento, por tal motivo es importante elegir la política que maximice las ganancias y minimice los costos, de aquí que los componentes del modelo de inventarios son: los costos de fabricación, de almacenaje y de penalización por faltantes.

El Costo de Fabricación es el valor de producir una unidad determinada de producto y se puede representar a través de una función. La forma más sencilla de esta función es aquella que se obtiene de manera directa, es decir, el precio unitario por el total de productos fabricados. Otra suposición común es que dicho costo se compone de dos partes: un término que es directamente proporcional a la cantidad producida y un término que es una constante para una cantidad producida y cero para ninguna cantidad producida. En este caso el costo de fabricación es el costo de producir cierta cantidad de unidades.

El Costo de Almacén representa los costos asociados con el almacenamiento del inventario hasta que el producto es requerido o vendido. Se puede incluir el costo de capital invertido, del espacio, seguros, protección e impuestos atribuibles al almacenamiento. Estos costos pueden ser expresados con una función de cantidad máxima que se guarda durante un período, de la cantidad promedio en el almacén o del exceso acumulado de los recursos adicionales a los requeridos.

El Costo de Penalización por faltantes surge cuando la cantidad que se requiere de un bien es mayor que el inventario disponible. En el costo por faltantes la demanda excesiva no se pierde, sino que queda pendiente hasta que se pueda satisfacer con el siguiente reabastecimiento.

Comúnmente el mercado establece tanto el precio como la demanda de un producto y por ende estos factores están fuera de la organización, el rendimiento sobre las ventas es independiente de la política de inventarios de la empresa y por tanto puede dejarse fuera.

1.3.2 Cadenas de Markov.

Una de las tantas aplicaciones de las cadenas de Markov es en la toma de decisiones. Su aplicación es bajo el supuesto de que la evolución del fenómeno correspondiente tiene características de aleatoriedad. Para su aplicación se debe considerar que la evolución de un proceso se lleve a cabo de una manera probabilística. Las cadenas de Markov tienen la propiedad particular de que las probabilidades que describen las formas en que el proceso evolucionará en el futuro dependen sólo del estado actual en que se encuentre el proceso y, por lo tanto, son independientes de los eventos ocurridos en el pasado. Se hace mención de este caso, debido a que en el problema que debemos resolver intervienen factores aleatorios, por ejemplo, la demanda del producto, el número de máquinas disponibles, etc. Haciendo uso de cadenas de Markov será posible pronosticar la demanda para un periodo posterior al que se esté trabajando y de esta manera estaremos también en posibilidades de obtener mayores rendimientos en cuanto al uso de las fuentes de producción (máquinas).

Definición. Un ***proceso estocástico*** se define como una colección ordenada de variables aleatorias $\{X_t\}$, en donde el índice t toma valores de un conjunto T dado. Con frecuencia T se toma como el conjunto de enteros no negativos y X_t representa una característica de interés medible en el tiempo t. Al conjunto de valores posibles que puede tomar una variable aleatoria $\{X_t\}$, se le conoce por lo general como ***espacio de estados*** y al conjunto de valores que puede tomar el tiempo (o espacio) T que indica el momento de observación, se le denomina ***espacio paramétrico.***

Las cadenas de Markov representan un proceso estocástico en el cual se tiene un espacio de estados discreto y un espacio paramétrico. Su característica principal es que la probabilidad de pasar de un estado actual a un estado futuro depende únicamente del estado actual

$$P\{X_{t+1}=j \mid X_0=k_0, X_1=k_1,\dots,X_{t-1}=k_{t-1}, X_t=i\}=P\{X_{t+1}=j \mid X_t=i\}, \text{ para } t=0,1,\dots,n$$

es decir, no se considera la historia del sistema, es por eso que también se les llama procesos sin memoria.

Las probabilidades condicionales $P\{X_{t+1}=j \mid X_t=i\}$ se llaman ***probabilidades de transición.*** Si para cada i y j,

$$P\{X_{t+1}=j \mid X_t=i\}=P\{X_1=j \mid X_0=i\}, \text{ para } t=0,1,\dots,n,$$

se dice entonces que las probabilidades de transición (de un paso) son estacionarias y por lo general se denotan por p_{ij}. Así, tener probabilidades de transición estacionarias implica que las probabilidades de transición no cambian con el tiempo. La existencia de probabilidades de transición (de un paso) estacionarias también implica que, para cada i, j y m (m=0,1,2,...,n),

$$P\{X_{t+m}=j \mid X_t=i\}=P\{X_m=j \mid X_0=i\}, \text{ para } t=0,1,\dots,n$$

Estas probabilidades condicionales casi siempre se denotan por $p_{ij}^{(m)}$ y se llaman probabilidades de transición de m pasos. Así, $p_{ij}^{(m)}$ es la probabilidad condicional de que la variable aleatoria X, comenzando en el estado i, se encuentre en el estado j después de m pasos.

Como las $p_{ij}^{(m)}$ son probabilidades, deben ser no negativas y, como el proceso debe hacer una transición a algún estado, deben satisfacer las propiedades

$$p_{ij}^{(m)} \geq 0, \qquad \text{para toda } i \text{ y } j;\ m = 0,1,\ldots,n$$

y

$$\sum_{j=0}^{R} p_{ij}^{(m)} = 1, \qquad \text{para toda } i \text{ y } j;\ m = 0,1,\ldots,n$$

Una notación conveniente para representar las probabilidades de transición de m pasos es la forma matricial

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0R} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1R} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{R0} & p_{R1} & p_{R2} & \cdots & p_{RR} \end{bmatrix}$$

1.3.3 Sistemas dinámicos.

Actualmente existe una amplia variedad de disciplinas en donde suelen presentarse problemas muy complejos y grandes que no es posible resolver con ninguno de los modelos tratados anteriormente, sin embargo, la aplicación del modelo dinámico proporciona una herramienta de gran ayuda para la resolución a este tipo de problemas. Se han tenido grandes aportaciones dentro de su amplio campo de aplicación. Por mencionar un ejemplo, podemos hablar de la determinación de la trayectoria de un cohete a la Luna con un consumo mínimo de combustible.

El ***Modelo Dinámico*** consiste en distribuir recursos limitados a determinados objetivos planteados en él, y que compiten en un período de tiempo. El modelo está integrado por dos tipos de variables llamadas ***Variables de Estado*** y ***Variables de Control***, donde las variables de estado son funciones continuas del tiempo que caracterizan el estado del sistema y las variables de control son funciones del tiempo, continuas a trozos, que representan las posibles decisiones que se pueden tomar en el sistema.

Se contempla también un conjunto de ecuaciones diferenciales llamadas ***ecuaciones de Movimiento***, que son el correspondiente al conjunto de restricciones ya explicado en modelos anteriores, y una función objetivo dependiente de las variables de control y las variables de estado.

2. PROPUESTA DE UN MODELO DE ASIGNACIÓN.

En este capítulo se propone un modelo matemático que genera una asignación óptima de la carga de trabajo. Se pretende que éste sea una herramienta útil en cualquier organización que opere bajo características similares a la aquí tratada. Cabe señalar que cada caso tendría sus propias particularidades por lo que se le harían al modelo los ajustes pertinentes para cada situación.

2.1 Descripción del modelo.

De manera muy sencilla se puede definir un modelo como la representación concisa de los detalles que se consideran importantes, de un fenómeno. Existe una clasificación muy general en la cual se encuentran incluidos los modelos matemáticos. En estos, la representación de los detalles se hace por medio de ecuaciones matemáticas[4].

Los modelos matemáticos pueden ser también llamados ***de optimización***. Esto cuando el modelo tiene como objetivo lograr el mejor valor de una función, ya sea minimizando costos o maximizando ganancias.

Los modelos de optimización, por el tipo de análisis que requieren, pueden ser clasificados también como ***analíticos o numéricos; lineales o no lineales; discretos o continuos*** y

[4] **Los modelos matemáticos reciben el nombre de "modelos matemáticos ejecutables ", siempre y cuando éstos puedan ser instrumentados en un programa de computadora.**

estáticos o dinámicos y, de acuerdo al tratamiento que den al fenómeno, pueden ser *determinísticos o probabilísticos.*

2.2 Características del modelo propuesto.

En términos de la nomenclatura dada anteriormente se puede decir, de manera muy general, que el modelo que aquí se propondrá tiene las siguientes características:

Es ***Matemático,*** porque el problema está representado a través de ecuaciones. ***De Optimización,*** porque se tiene una función donde se busca minimizar el tiempo de producción. ***Numérico,*** porque la obtención del óptimo es a través de la evaluación de la función objetivo. ***No Lineal,*** porque la función objetivo está integrada mediante una composición de funciones. ***Discreto,*** porque las funciones son evaluadas puntualmente en el conjunto factible. ***Estático,*** porque no se considera la razón de cambio en el tiempo y ***Determinístico,*** porque no se tomaron en cuenta factores aleatorios.

El fenómeno es relativamente sencillo porque no se está contemplando, por ejemplo, la aleatoriedad, lo cual lo aleja un poco de la realidad, pero si alguna situación particular lo requiriese se podría incluir esta característica en el modelo.

2.3 Estructuración del modelo.

El modelo que se propone está compuesto por una función objetivo, en la cual se optimiza el tiempo de producción de todas las fuentes de trabajo que se tienen contempladas para la fabricación de un producto; las variables que se incluyen en el modelo están dadas por los conceptos particulares de producción de la empresa en estudio. La flexibilidad del modelo permite, por ejemplo, adicionar a la función objetivo constantes que representen costos de tiempo.

El modelo consta además de una restricción que se basa en fijar una cota superior en cuanto al tiempo; es decir, el tiempo total de producción no puede ser mayor que una cierta cota fijada.[5]

[5] Con base en cada situación particular, se podrían agregar tantas restricciones como se requiriesen.

A continuación se establece la notación que habrá de documentar la formulación del modelo matemático para nuestro problema de asignación.

x_j : Variable que identifica el concepto j asociado con la producción.
$Q = (x_1, ..., x_n)$: Punto de conceptos de producción.
C : El conjunto (finito) de todos los puntos de conceptos de producción.
$f_j : C \rightarrow \Re$: j-ésima función (en este caso, máquina)
p : Número total de funciones.
$t_{jk_1...k_s}$: Valor de la función j en el punto $(x_1,...,x_{k_1},...,x_{k_s},...,x_n)$ de C.

El problema consiste en

$$\text{Optimizar} \quad t(f_1,..., f_p) = \beta + \sum_{\substack{\text{Todas las variables } x_k \\ \text{que deban ser agotadas}}} t_{jk_1...k_s} \qquad (1)$$

Donde β es una cantidad fija que, en general, podría representar la suma de varias cantidades concernientes a un proceso particular.

Una variable x_j puede ser, por ejemplo, un número de pedido, el modelo del producto solicitado en un pedido, la cantidad solicitada de un producto en un pedido determinado, el rendimiento de una máquina, etc. Si x_j es, digamos, un número de pedido, en el punto

$(x_1,\ldots,x_n)$ dicha variable jugaría el papel de un identificador que no influye en el valor de f pero que se considera necesario para formular las salidas.

Cabe aclarar que el problema de optimización se puede llevar a otro que requiere en general de menos tiempo de máquina. Por ejemplo, para detener el proceso de cálculo se puede imponer una condición externa como

$$t(f_1,\ldots,f_p) \leq T,$$

donde T es una cantidad predeterminada. También se le pueden imponer condiciones externas al algoritmo, que indiquen políticas de producción; por ejemplo, se podría poner como condición el hecho de que participen el mayor número posible de máquinas en la producción, en un período dado, o también, que se omitan alguna o algunas de las máquinas en dicho período.

Para un problema de producción determinado, habrá conceptos que deban ser considerados en su totalidad; por ejemplo, si se habla de pedidos, la demanda de productos especificados en éstos debe quedar satisfecha, así, la leyenda que aparece al pie de la suma al lado derecho de la igualdad (1), se referiría, en este caso, a las cantidades solicitadas en tales pedidos.

3. CASO DE ESTUDIO.

Para verificar la funcionalidad del modelo propuesto, se realiza la instrumentación de éste en la empresa ***Metales y Contactos***, donde el objetivo es automatizar el proceso de asignación de carga de trabajo a las máquinas que están disponibles.

3.1 La Empresa Metales y Contactos.

Metales y Contactos es una industria que se dedica a la producción de remaches, su producción está en función de los pedidos solicitados por los clientes y la prioridad de la empresa es satisfacerlos en las fechas acordadas.

Cuando se contactó con la empresa, la forma de realizar la planeación de las actividades de las máquinas era de manera intuitiva por cuatro o cinco personas expertas en el proceso de producción, en todo un día de trabajo -martes- (con ajustes mínimos en el resto de la semana) y sin garantizar que fuera la mejor asignación, pero hasta ese momento, el número de clientes y la capacidad de producción que tenían, les permitía satisfacer la demanda del producto en las fechas acordadas. La visión de los empresarios fue que por la tendencia de crecimiento de la empresa, este proceso de asignación se dificultaría por lo que resultaba necesario automatizarlo.

La empresa está integrada por cuatro áreas: ventas, almacén, producción y programación de la producción. En esta última se fundamenta el análisis para la implementación del modelo propuesto.

3.2 Proceso de planeación de la producción.

La asignación de cargas de trabajo a las máquinas se encuentra inmersa en el proceso de planeación de la producción, que es una parte muy importante dentro de la empresa, debido a que la producción depende directamente de una buena programación. Aquí se determina, de acuerdo a la demanda del producto, qué máquina fabricará un modelo determinado para satisfacer los pedidos solicitados.

3.2.1 Áreas de interés.

El departamento de Programación de la Producción, necesita información de otras áreas para llevar a cabo la asignación de actividades en las máquinas. A continuación se enuncia la información que se requiere de cada una.

El departamento de Ventas se encarga de reportarle el total de pedidos que se tienen que cubrir en el mes.

Si en un momento dado la cantidad de piezas de un modelo que se asigna a una máquina no cubre el lote mínimo que se debe asignar a la misma, entonces el área de ventas se encarga de determinar un stock de consumo mensual si se trata de un modelo de alta demanda y uno de consumo trimestral cuando el modelo es de baja demanda.

El Almacén presenta dos datos importantes: existencia de remaches en embarque, que son los modelos que están contemplados para ser entregados al cliente y la existencia de remaches en producto terminado, que es el stock con que se cuenta para cada modelo.

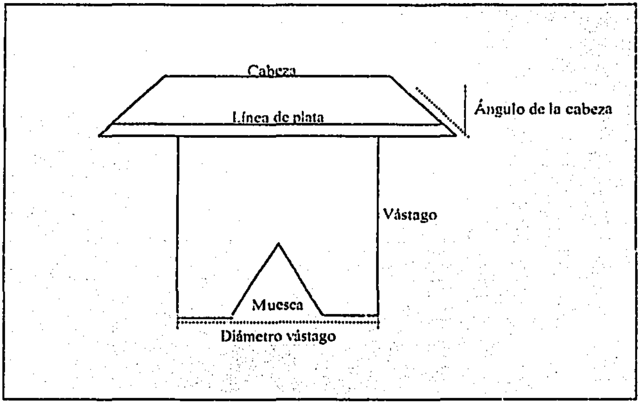
El área de Producción da a conocer la existencia de remaches en proceso, es decir, todos aquellos modelos que actualmente se están fabricando.

En el área de ***Programación de la Producción*** se requiere tener un control de la relación de remaches en cada máquina y su saldo respecto a la orden de trabajo original. Al momento de realizar el programa de actividades del mes se consideran las siguientes características para la fabricación de los remaches: la materia prima, el peso del material, las medidas principales y las herramientas necesarias para montarlos en las máquinas.

3.2.2 Conceptos Generales.

A continuación se describen de manera breve algunos conceptos importantes de producción.

Los productos, en este caso los remaches, se fabrican con un solo metal, plata o cobre, o bien con la combinación de éstos. Las partes principales que conforman el producto son: la cabeza, el vástago, el diámetro del vástago, el ángulo de la cabeza, la muesca y la línea de plata; en el caso de los modelos compuestos de los dos metales.



La Materia Prima está conformada por alambre de plata (Ag) en distintas aleaciones y de cobre (Cu). El grosor del alambre varía de acuerdo al producto y normalmente está en un rango de medida de 0.0001 a 0.003 inch. menores al diámetro del vástago del remache.

Existen 26 máquinas clasificadas en tres tipos: Japonesas, Chugay y Ayase. De las Ayase sólo dos pueden hacer remaches de un solo metal, el resto puede hacer de cualquier tipo. Las máquinas veloces hacen productos con dimensiones menores a las máquinas más lentas.

Cuando una máquina determinada se configura para fabricar un cierto tipo de remache, lo conveniente es fabricar una cantidad mínima de piezas, lo que se refleja en la siguiente tabla como lote mínimo, éste, junto con otras características generales de cada tipo de máquina se resumen en el siguiente cuadro:

Tipo de Máquina	**Cantidad**	**Diámetro del Vástago del Remache (pulgadas)**	**Piezas por Día**	**Lote Mínimo de fabricación diaria**
Japonesa	8	0.035 - 0.093	60,000 - 80,000	200,000
Chugay	7	0.035 - 0.153	15,000 - 60,000	100,000
Ayase	11	0.055 - 0.090	80,000 - 130,000	500,000

Los pedidos se formulan en el transcurso del mes, pero los de mayor consumo normalmente entran al inicio del período. El cliente envía una relación de su stock, su venta del último mes y su consumo estimado para el siguiente período, de aquí parte la información para la planeación.

Cada pedido contiene el tipo de remache, el número de piezas y la fecha en que se requiere dicho pedido. Normalmente un pedido puede dividirse en varios parciales para su entrega en distintas fechas.

En las Órdenes de Trabajo se especifica el número de piezas a fabricar, todo producto para entrar a proceso debe estar respaldado por un programa de producción y la orden mencionada. En caso de pedidos que sean menores al lote mínimo establecido por tipo de máquina, se hace un stock del producto, considerando el total del mes o del trimestre, según corresponda.

3.2.3 Proceso de Asignación de carga de trabajo a las máquinas.

El objetivo, de asignar de una cierta manera la carga de trabajo a las máquinas, es cumplir con las fechas de entrega requeridas por los clientes, pero buscando la combinación que permita la mayor productividad de los equipos. Cuando en una máquina se realiza un cambio de producto, algunas veces será necesario reconfigurarla[6], lo que implica inversión de tiempo.

[6] La reconfiguración de una máquina se refiere a los cambios de algunos de sus componentes para la elaboración de un nuevo modelo.

En el caso de que los pedidos, para un mes determinado rebasen la capacidad normal de producción, se debe buscar la posibilidad de aumentar la producción diaria de las máquinas mediante algún plan de productividad acordado con la gerencia de planta y el área de producción. Si a pesar de ello no es posible cumplir con los pedidos, se debe avisar al departamento de Ventas y a la dirección de la empresa a fin de que estos órganos indiquen que productos tienen prioridad o se proporcione otra alternativa de solución: (las prioridades podrían ser incluidas en las funciones f_j). Es importante mencionar que se trabaja con base en un objetivo de venta y de su relación con su capacidad de producción.

Otro objetivo es que el programa mensual, una vez diseñado, sufra el menor número posible de cambios, a fin de no alterar los planes de algunos departamentos que trabajan con base en dicho programa. Esto implica asignarle a una máquina el número mínimo de modelos de remaches o, al menos, variar lo menos posible el diámetro de los materiales de los remaches con la finalidad de reducir, en lo posible, los tiempos de configuración.

Para saber el número de piezas por fabricar, de cada tipo de remache, primero se toma la cantidad solicitada por el cliente, se le resta lo que está fabricado y lo que está en proceso y la diferencia será el número de piezas por fabricar. En el caso de los productos de mediano y bajo consumo se programa fabricar también un stock de tres meses de consumo de estos, excepto en aquellos tipos de remache cuyo contenido de plata es muy alto. En el caso de los productos de alto consumo normalmente se va al día. Si se dejan de pedir estos, su producción se programa siempre tratando de contar con un stock suficiente para completar un mes de ventas, siempre que ello no implique un alto consumo de plata.

Una vez que se tiene la lista de productos por fabricar y el estado actual del programa de producción, se procede a realizar la asignación. Generalmente se intenta respetar el orden del programa de la semana anterior pero, si por alguna razón se requiere cambiar o asignar un nuevo tipo de remache a una máquina, esto se hace conforme a los siguientes criterios y orden de importancia:

1.- Al momento de asignar un remache a una máquina se debe tener como prioridad que la fecha final de producción sea con algunos días de anticipación a la fecha de entrega del pedido, ya que la finalidad de la programación de la producción es satisfacer la demanda en las fechas acordadas.

2.- Si existe una variación en el lote de producción, no se debe afectar la entrega del producto en las fechas acordadas. En caso de que una de las máquinas dejara de operar por la cancelación de un pedido o por descompostura, entonces, en la calendarización, la carga de trabajo se debe redistribuir en las que sí están funcionando.

3.- Si el pedido es muy grande y la fecha en que se terminará no cumple con la fecha de entrega se debe buscar la posibilidad de montar ese remache en una máquina adicional a la que ya está asignado. El número máximo de máquinas en que se puede asignar un tipo de remache es de tres.

4.- Se pretende que los modelos que están en proceso no se dejen de producir hasta que se haya cubierto el lote de producción y, en caso de tener que hacerlo, se deben considerar dos días, como mínimo, para poder reconfigurar la máquina. A su vez, se debe tener en cuenta que esta reconfiguración va a retrasar la manufactura de algunos de los productos que estén todavía pendientes de fabricar.

5.- Se debe buscar que el diámetro del alambre para el remache a procesar en la siguiente asignación, sea igual, o lo más cercano posible, al que la máquina estaba trabajando.

6.-Para la fabricación de un remache en una máquina se toma como prioridad la fecha de entrega del pedido.

7.- Se identifica si el remache está compuesto de uno o dos metales, hay dos máquinas que sólo admiten remaches de un solo metal. Todas las demás pueden hacer de los dos tipos de remaches.

8.- En caso de que dos o más remaches compitan por una misma máquina y una vez agotados los criterios anteriores se debe consultar con ventas, la gerencia y la dirección general, acerca de la prioridad para fabricarlos.

También existen restricciones particulares para el caso de las máquinas Ayase que se deben considerar al momento de llevar a cabo la asignación. Éstas se enuncian a continuación:

a) Se deben asignar todos los remaches que no tengan ángulo en la cabeza, es decir, remaches rectos, al igual que los remaches que tengan muescas profundas debido a que es el único tipo de máquina que los puede fabricar.

b) Por la capacidad de producción que tiene este tipo de máquina se recomienda asignar lotes grandes, para que la reconfiguración de éstas sea mínima.

c) Es posible asignar aquellos tipos de remache que tengan líneas de plata muy pequeñas.

d) Existen dos modelos de remache que sólo se pueden hacer en este tipo de máquina.

3.3 Planteamiento del modelo para el problema particular.

Haciendo referencia al modelo matemático propuesto en el capítulo dos, enseguida se establece el planteamiento correspondiente al caso de la empresa bajo estudio.

3.3.1 Definición de variables.

Las ***variables*** x_j identifican el pedido, el modelo, la cantidad de modelo-pedido y el rendimiento de una máquina para un modelo específico.

M_j : j-ésimo modelo de producto.

P_k : k-ésimo pedido.

p_{kj} : cantidad de producto del modelo k solicitada en el pedido j.

r_{ij} : rendimiento de la i-ésima máquina en la fabricación del modelo j.

De acuerdo a la estructuración del modelo el conjunto C de puntos de conceptos de producción está dado por

$$C = \{ (M_j,\ P_k,\ p_{kj},\ r_{ij}) \}.$$

Las *máquinas* o fuentes de producción se denotan por

f_i: i-ésima máquina ($i = 1, 2, ..., 26$).

p: número total de máquinas ($p = 26$).

Recordando que el objetivo de la asignación de carga de trabajo a las máquinas es cumplir con las fechas de entrega, se plantea la ***función objetivo*** de la siguiente manera:

$$\text{Optimizar} \quad t(f_1, f_2, ..., f_p) = \sum_{\text{Toda la demanda}} t_{ijk}$$

$$t_{ijk} = f_i(M_j, P_k, p_{jk}, r_{ij}) + \delta$$

donde $\delta = 1$ si se debe realizar ajuste a la máquina y $\delta = 0$ en cualquier otro caso.
El valor de la función f_i en el punto $(M_j, P_k, p_{jk}, r_{ij})$ lo definimos, para este caso, mediante la expresión

$$f_i(M_j, P_k, p_{jk}, r_{ij}) = \frac{p_{jk}}{r_{ij}} \quad .$$

Como restricción se impone el hecho de que se ocupe el mayor número posible de máquinas.

El *Algoritmo de cálculo para el consumo de tiempo en una máquina* establece los criterios a seguir en la instrumentación del modelo particular.

Notación

TDI_i: Tiempo disponible inicial en la máquina i.

TDA_{ij}: Tiempo disponible en la máquina i, al concluir el paso j.

La máquina i empieza con un tiempo disponible de **TDI_i** unidades.

Se asigna la manufactura del modelo M_{j_1}, correspondiente al pedido P_{k_1} a la máquina i y se consume un tiempo de $t_{ij_1k_1}$ unidades.

$$t_{ij_1k_1}=f_i(M_{j_1}, P_{k_1}, p_{j_1k_1}, r_{ij_1}) + \delta$$

Tiempo disponible actual en la máquina i

$$TDA_{i_1} = TDI_i - t_{ij_1k_1}$$

Si TDA_{i_1} es:

Menor que cero, entonces la tarea se asigna a otra máquina.

Igual que cero, entonces la tarea se asigna a la máquina i y ya no se asignan más tareas.

Mayor que cero, entonces la tarea se asigna a la máquina i.

Ejemplo

- Supóngase que TDA $_{i_1} > 0$, entonces:

Se asigna la manufactura del modelo M_{j_2} correspondiente al pedido P_{k_2} a la máquina i y se consume un tiempo de $t_{ij_2k_2}$ unidades.

$$t_{ij_2k_2} = f_i\,(M_{j_2}, P_{k_2}, p_{j_2k_2}, r_{ij_2}) + \delta Ta$$

Tiempo disponible actual en la máquina i

$$TDA_{i_2} = TDA_{i_1} - t_{ij_2k_2}$$

- Supóngase TDA $_{i_2} < 0$, entonces:

La tarea se asigna a otra máquina.

- Supóngase TDA $_{i_2} = 0$, entonces:

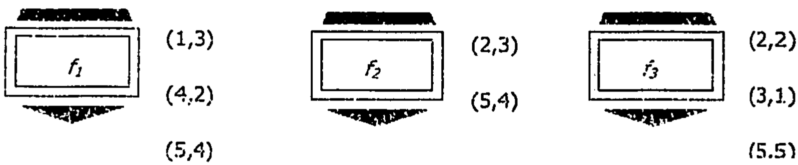
La tarea se asigna a la máquina i y ya no se asignan más tareas.

3.4 Evaluación del modelo.

Para hacer una prueba correspondiente a la aplicación del modelo se considera el caso donde existen tres máquinas disponibles para solventar cinco pedidos que contienen un total de cinco modelos por entregar.

Características asociadas a las máquinas

Matrices de Modelo-Rendimiento



(M_j, r_{ij})=(j-ésimo modelo de producto, rendimiento de la máquina i para el producto j)

A_i: matriz modelo-rendimiento para la máquina i

Formulación de pedidos

$$A_1 = \begin{pmatrix} 1 & 3 \\ 4 & 2 \\ 5 & 4 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 2 & 3 \\ 5 & 4 \end{pmatrix} \qquad A_3 = \begin{pmatrix} 2 & 2 \\ 3 & 1 \\ 5 & 5 \end{pmatrix}$$

Pedido 1

Modelo	Cantidad
2	15
4	10

Pedido 2

Modelo	Cantidad
3	20

Pedido 3

Modelo	Cantidad
1	10
2	20

Pedido 4

Modelo	Cantidad
5	20

Pedido 5

Modelo	Cantidad
1	15

Tabla de información de Modelo-Pedido

M \ P	1	2	3	4	5
1			10		15
2	15		20		
3		20			
4	10				
5				20	

Matriz de Modelo-Pedido

$$B = \begin{pmatrix} 0 & 0 & 10 & 0 & 15 \\ 15 & 0 & 20 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 \end{pmatrix}$$

$B=(p_{ij})$

p_{ij} es la cantidad que se solicita del modelo i en el pedido j.

B es la matriz de cantidades modelo-pedido.

Pruebas

Para calcular las cantidades t_{ijk} se usaron las matrices A_i de modelo-rendimiento y la matriz B de modelo-pedido. Las variables M_i, P_j y p_{ij} aparecen en la matriz B y los valores de las r_{ij} se encuentran en las matrices A_i.

Prueba 1

Asignaciones fijando una columna y barriendo en orden los renglones.

Las columnas se eligieron también en orden 1, 2, ... , 5.

$t_{i21} = t_{221} = f_2\,(M_2, P_1, 15, 3) = 5$

$t_{i41} = t_{141} = f_1\,(M_4, P_1, 10, 2) = 5$

$t_{i32} = t_{332} = f_3\,(M_3, P_2, 20, 1) = 20$

$t_{i13} = t_{113} = f_1\,(M_1, P_3, 10, 3) + 2 = 16/3$ c.c.[7]

[7] Con configuración.

$t_{i23} = t_{223} = f_2(M_2, P_3, 20, 3) = 20/3$ s.c.[8]

$t_{i54} = t_{154} = f_1(M_5, P_4, 20, 4) + 2 = 7$ c.c.

$t_{i15} = t_{115} = f_1(M_1, P_5, 15, 3) + 2 = 7$ c.c.

$$\sum t_{ijk} = 56$$

Prueba 2

Asignaciones fijando un renglón y barriendo en orden las columnas.

Los renglones se eligieron también en orden 1, 2, ... , 5.

$t_{i13} = t_{113} = f_1 (M_1, P_3, 10, 3) = 10/3$

$t_{i15} = t_{115} = f_1 (M_1, P_5, 15, 3) = 5$ s.c.

$t_{i21} = t_{221} = f_2 (M_2, P_1, 15, 3) = 5$

$t_{i23} = t_{223} = f_2 (M_2, P_3, 20, 3) = 20/3$ s.c.

$t_{i32} = t_{332} = f_3(M_3, P_2, 20, 1) = 20$

$t_{i41} = t_{141} = f_1(M_4, P_1, 10, 2) + 2 = 7$ c.c.

$t_{i54} = t_{354} = f_3(M_5, P_4, 20, 5) + 2 = 6$ c.c.

[8] Sin configuración.

$$\sum t_{ijk} = 53$$

Prueba 3

Asignaciones fijando un renglón y barriendo en orden las columnas.

El orden de los renglones se tomó en sentido inverso 5, 4, ... , 1.

$t_{i54} = t_{154} = f_1(M_5, P_4, 20, 4) = 5$

$t_{i41} = t_{141} = f_1(M_4, P_1, 10, 2) + 2 = 7$ c.c.

$t_{i32} = t_{332} = f_3(M_3, P_2, 20, 1) = 20$

$t_{i21} = t_{221} = f_2(M_2, P_1, 15, 3) = 5$

$t_{i23} = t_{323} = f_3(M_2, P_3, 20, 2) + 2 = 12$ c.c.

$t_{i13} = t_{113} = f_1(M_1, P_3, 10, 3) + 2 = 16/3$ c.c.

$t_{i15} = t_{115} = f_1(M_1, P_5, 15, 3) = 5$ s.c.

$$\sum t_{ijk} = 59.33$$

Resultados

Suponiendo que se hubiera llegado al óptimo t^*, éste se habría dado en la segunda prueba, luego, la solución al problema quedaría como

Máquina	Modelo	Cantidad a producir
1	1	25
1	4	10
2	2	35
3	3	20
3	5	20

La cantidad total de unidades que se van a producir son 110

$$\sum b_{ij} = 110.$$

El tiempo necesario para producir esas 110 unidades es

$$t^* = 53$$

4. AUTOMATIZACIÓN DEL MODELO.

Sin lugar a duda la rentabilidad de un modelo matemático se vuelve satisfactoria cuando éste se implementa en un sistema de computadora. Se destaca la importancia de llevar a cabo la automatización considerando que del sistema se pueda obtener una propuesta que auxilie en la toma de decisiones para buscar maximizar beneficios y minimizar costos. Se busca, además, que los beneficios sean duraderos, es decir, a medida que el sistema tenga dentro de sus atributos una buena planeación, análisis, diseño e implementación que se vuelva una excelente inversión para quienes tengan como objetivo una estabilidad con tendencias de crecimiento.

Para establecer una adecuada automatización del modelo se requiere la ayuda de una herramienta que permita que el desarrollo del software sea extendible y reutilizable, esto es, extendible para que permita que el sistema pueda evolucionar a través del tiempo, y la integración de nuevos procesos sea sencilla y reutilizable para que al adicionarlos no sea necesario volver a desarrollar el sistema.

4.1 Fases del sistema.

En el desarrollo de un sistema intervienen tres partes muy importantes:

- ***Comité directivo de Sistemas de Información (SI).***
- ***Poseedor del problema.***
- ***Analistas de sistemas.***

En el primer grupo, se encuentran los administrativos de un departamento de sistemas. El segundo grupo de personas, serán los usuarios del sistema. El último grupo, se encuentra integrado por un líder de proyecto y un grupo de programadores.

El Sistema de Información será desarrollado en cuatro fases, en las cuales cada una de las partes mencionadas anteriormente tiene que llevar a cabo una serie de actividades como se muestran a continuación:

La fase de planeación

<table>
<tr><th>Comité Directivo de SI</th><th>Poseedor del problema</th><th>Analista de sistemas</th></tr>
<tr><td></td><td>Reconocer la existencia del problema</td><td rowspan="2">Conocer el problema</td></tr>
<tr><td></td><td>Definir el problema</td></tr>
<tr><td></td><td colspan="2">Definir los objetivos del sistema</td></tr>
<tr><td></td><td colspan="2">Identificar las restricciones del sistema</td></tr>
<tr><td></td><td></td><td>Realizar un estudio de factibilidad</td></tr>
<tr><td></td><td></td><td>Preparar una propuesta del proyecto</td></tr>
<tr><td colspan="2">Aprobar o desaprobar la propuesta del proyecto</td><td></td></tr>
</table>

La fase de análisis

<table>
<tr><th>Comité Directivo de SI</th><th>Poseedor del problema</th><th>Analista de sistemas</th></tr>
<tr><td colspan="2">Dar a conocer la propuesta del proyecto</td><td></td></tr>
<tr><td></td><td colspan="2">Organizar el equipo de desarrollo</td></tr>
<tr><td></td><td colspan="2">Definir las necesidades de información</td></tr>
<tr><td></td><td colspan="2">Limitar funciones y responsabilidades entre ambos equipos</td></tr>
<tr><td></td><td></td><td>Preparar la propuesta de diseño</td></tr>
<tr><td colspan="2">Aprobar o desaprobar el proyecto de diseño</td><td></td></tr>
</table>

La fase de diseño

<table>
<tr><th>Comité Directivo de SI</th><th>Poseedor del problema</th><th>Analista de sistemas</th></tr>
<tr><td></td><td></td><td>Preparar el diseño detallado del sistema</td></tr>
<tr><td></td><td></td><td>Determinar alternativas del sistema</td></tr>
<tr><td></td><td></td><td>Evaluar las alternativas del sistema</td></tr>
<tr><td></td><td></td><td>Seleccionar la mejor alternativa</td></tr>
<tr><td></td><td></td><td>Preparar la propuesta de implementación</td></tr>
<tr><td colspan="2">Aprobar o desaprobar la implementación del sistema</td><td></td></tr>
</table>

La fase de implementación

Comité Directivo de SI	Poseedor del problema	Analista de sistemas
Anunciar la implementación		
Controlar y Supervisar		Obtener el hardware
		Obtener el software
		Preparar la base de datos
		Preparar las conexiones físicas
		Capacitar a participantes y usuarios
		Preparar la propuesta de terminación
Aprobar o desaprobar la propuesta de terminación		
	Implantar el nuevo sistema	

4.2 Algoritmo de programación para el modelo de asignación.

Para establecer el algoritmo de programación que nos permita automatizar el proceso de asignación de las cargas de trabajo, se hace referencia a las características del modelo matemático a fin de interpretar la estructura del modelo a través de una serie de instrucciones. Se consideran, además, los criterios de producción y las prioridades de asignación mencionados en el capítulo anterior que son particulares del caso de estudio, algunos de ellos son: cumplir con fechas de entrega de los pedidos, producción continua y partición de pedidos. Otra característica particular de la empresa ***Metales y Contactos*** es que cada grupo de máquinas tiene sus propias restricciones.

Con base en las restricciones del problema particular se requieren datos de entrada como son: el total de máquinas que están contempladas para llevar a cabo el proceso de producción, el programa de producción (matriz modelo-pedido), el tiempo disponible de cada una de las máquinas y la capacidad de producción (matriz modelo-rendimiento).

La parte central del sistema son los procesos que habrán de transformar los datos de entrada. Los procesos son: el de ***Máquinas disponibles***, éste consiste en establecer del conjunto de las máquinas, cuáles son las que cuentan con tiempo disponible para realizar un modelo requerido, además ordenar éstas de acuerdo al rendimiento de producción.

```
void MaquinasDisponibles() {
    for (i=0; i<maquinas; i++)
                for (j=0;j<cap_maq;j++)
                    for (k=0;k<2;k++)
                        fscanf(maqdis,"%d",&Maquinas[i][j][k]);

        for (i=0; i<maquinas; i++)
                fscanf(tdemaq,"%f",&TIEDIS[i]);

    }

void Ordenacion() {
        for (fh=0; fh<maquinas;fh++)
            for (fi=0; fi<cap_maq; ++fi) {
                    ftem=Maquinas[fh][fi][1];
                    gtem=Maquinas[fh][fi][0];
                    fk=fi-1;
                    while ((fk>=0) && (ftem>Maquinas[fh][fk][1])) {
                            Maquinas[fh][fk+1][1]=Maquinas[fh][fk][1];
                            Maquinas[fh][fk+1][0]=Maquinas[fh][fk][0];
                            fk--;
                            }
```

```
            Maquinas[fh][fk+1][1]=ftem;
            Maquinas[fh][fk+1][0]=gtem;
            }
    }
```

El proceso de ***Generación del Espacio***, en él se realiza una permutación ordinaria de los elementos de la matriz modelo-pedido que consiste en un proceso iterativo para la evaluación de diferentes asignaciones.

```
void Permutacion(int Mqs, Maquin &Org)
    {
        IntLar pi;
        Maquin Temporal;

        Iguala(Org, Temporal);
        if (Mqs != 2) {
           Permutacion(Mqs-1, Org);
           pi = maquinas - Mqs;
           for (int pj=pi+1; pj<maquinas; pj++) {
                  Iguala(Temporal, Org);
                  Intercambia(pi, pj, Org);
                  Permutacion(Mqs-1, Org);
                  }
           }
        else { // Recorrido de la matriz Org
            Inicializa();
            Asignacion_de_Pedidos();
            GuardaAsignacion();
            //ParticionP();
            tot++;
            Intercambia(cap_maq-2, cap_maq-1, Org); // Recorrido del matriz Org
            Inicializa();
            Asignacion_de_Pedidos();
            GuardaAsignacion();
            //ParticionP();
            tot++;
            }
        }
```

```
void Intercambia(int u,int v, Maquin &Maquinas) {

     IntLar tem;

     for (int si=0;si<cap_maq;si++ )
          for (int sj=0; sj<2; sj++) {
               tem = Maquinas[u][si][sj];
               Maquinas[u][si][sj] = Maquinas[v][si][sj];
               Maquinas[v][si][sj] = tem;
               }
          }

void Iguala(Maquin &n, Maquin &m) {

     IntLar wh,wi,wj;

     for (wh=0;wh<maquinas;wh++)
          for (wi=0; wi<cap_maq; wi++)
               for (wj=0; wj<2; wj++)
                  m[wh][wi][wj] = n[wh][wi][wj];
}

void factorial(int Numfac) {

     int fac;
     RLar Solfac=1;

     for (fac=1; fac<=Numfac; fac++)
          Solfac=Solfac*fac;
     DimEspacio=Solfac;
}
```

El proceso de ***Asignación de pedidos***, aquí se determina qué máquina reúne las características necesarias para la fabricación del modelo solicitado. además de contemplar los tiempos disponibles de las máquinas que tienen asignada alguna tarea. todo ello con la finalidad de asignar un modelo a la máquina que lo realice en el menor tiempo posible.

```
void Asignacion_de_Pedidos() {

     for (i=0;i<modelos;i++) {
             incrmn=0;
             for (j=1;j<total_p+1;j++) {
                     if ((Modelo_Pedido[i][j] != 0) && (ModeAsig[i][j] != 1)) {
// Determinar si existe un modelo por producir: considerar los casos donde modelo_pedido = 0
                             for (t=0;t<cap_maq;t++)
// Número de modelos que fabrica una máquina i
                                     for (r=0; r<maquinas;r++) {
// Total de máquinas para la producción
                                         if (ModeAsig[i][j] == 1) break;
                                            if ((Maquinas[r][t][0] == Modelo_Pedido[i][0])&&(ModeAsig[i][j] != 1)) {
                                                  Tiempo_Maquina(Maquinas[r][t][0], Modelo_Pedido[i][j],
                                                  Maquinas[r][t][1], r);
// Parámetros de la función (Modelo, Cantidad, rendimiento)
                                                  }
// Se evalúa la función únicamente una vez y sale del ciclo del número de modelos
                                            }
                                          } // Se recorrió toda la matriz modelo pedido
                                  }
                          }
                     }


void GuardaAsignacion() {

     for (int final=0;final<maquinas;final++)
               total=total+M_Produccion[final][3];// Suma de tiempos de producción

          fprintf (tiemin," TIEMPO DE PRODUCCIÓN %6.2f \n ", total);

          if (total<tiempo_t) {
             tiempo_t=total;
             }
     }


void Tiempo_Maquina (IntLar Modelo, double Cantidad,IntLar Rendimiento,int r) {
     if (M_Produccion[r][2] != Modelo) {
          if (M_Produccion[r][1]==0) {
             TDI[r]=TIEDIS[r]; // Tiempo disponible Inicial de la r-ésima Máquina
             Delta = 0; // Valor del Ajuste
             }
```

```
        else if (M_Produccion[r][1]==1) {
                Delta = 1;//Valor del ajuste por cambio de modelo
                }
        Verifica_Tiempo(Cantidad/Rendimiento + Delta * 2, Modelo, Cantidad, Rendimiento);
        }
        else if (M_Produccion[r][2] == Modelo) {
                Delta = 0;
                Verifica_Tiempo(Cantidad/Rendimiento + Delta * 2, Modelo, Cantidad,
Rendimiento);
                }
        }


void Verifica_Tiempo(double tiempo, IntLar Mode, double Cantidad, IntLar Rendimiento) {
    TDA[r]=TDI[r]-tiempo;
    if (TDA[r]<0) { // De cumplirse la condición no es posible asignar otro pedido a esta máquina
        return;
        }
    if (TDA[r]>=0) {
        M_Produccion[r][0]=r+1;
        M_Produccion[r][1]=1;
        M_Produccion[r][2]=Mode;
        M_Produccion[r][3]=M_Produccion[r][3]+Cantidad/Rendimiento + Delta * 2;
        if (ModeloAnterior != M_Produccion[r][2])
//Agregar la restricción de que sea la misma máquina
        M_Produccion[r][4]=incrmn++;
        TDI[r]=TDA[r],
        ModeAsig[i][j]=1;
        Reporte[i][j]=r+1;
        ModeloAnterior=M_Produccion[r][2];
        }
    }
```

Cuando se termina la asignación pueden quedar modelos fuera del programa, debido a que son pedidos muy grandes que no pueden ser realizados por una sola máquina por lo que es necesario llevar a cabo el ***Proceso de Partición de pedidos*** de tal manera que puedan ser contemplados en la producción.

```
void ParticionP() {

int inc=0;
double capacidad;
struct partmod *r;
struct lismaq *u;

// Almacenar los modelos que no se terminaron
  ModsnoTerminados();

// Almacenar las máquinas que aún tienen tiempo disponible
  OrdenMaquinas();

// Ordenar las máquinas con base a su rendimiento
  RendAscendente();

//Algoritmo de Partición

r = ModeNT -> Next;
u = MaqSel -> Next;

if (r==NULL)
            printf("\nNO EXISTEN MODELOS SIN TERMINAR\n");
if (u==NULL)
            printf("\nNO EXISTEN MAQUINAS CON TIEMPO DISPONIBLE\n");
while (r != NULL) {
 while (u!= NULL) {
        if (( u->MaqPtr[inc][1] == r->Model ) && ( u->MaqPtr[inc][1] > 0) && ( TDI[inc] > 0 ) ) {
                capacidad = u -> MaqPtr[inc][2] * TDI[inc];
// Como primera instancia una máquina no puede terminar la cantidad del modelo por estar dentro
de la partición
                TDI[inc] = TDI[inc] - (capacidad / u->MaqPtr[inc][2]);
                r -> Cantid = r -> Cantid - capacidad;
                if ( u->MaqPtr[inc+1][2] > 0 )
```

```
// Comienza la partición del resto de la cantidad del modelo
                              if ( (r->Cantid/u->MaqPtr[inc+1][2]) <= TDI[inc+1] ) {
// Al entrar, indica que será  suficiente con 2 máquinas para terminar el modelo

                              capacidad = (u->MaqPtr[inc+1][2] * TDI[inc+1]) - (r->Cantid);
                              if ( capacidad >= 0 ) { /*El pedido se cubrió*/
                                        TDI[inc+1] = TDI[inc+1] - ( r->Cantid / u -> MaqPtr[inc+1][2] );
                    r -> Cantid = r ->Cantid - (r->Cantid / u -> MaqPtr[inc+1][2] ) * u->MaqPtr[inc+1][2];
                                        }
 if ( capacidad  < 0 ) {
 // Al entrar aquí  indica que el modelo no se cubrió y por tanto existe un error

        r -> Cantid = r -> Cantid - (u->MaqPtr[inc+1][2] * TDI[inc+1]);
          TDI[inc+1] = TDI[inc+1] - (capacidad / u->MaqPtr[inc+2][2]);
                            } // FIN: Se termina con dos máquinas
                              }
else if ( (r->Cantid / u->MaqPtr[inc+1][2] ) > TDI[inc+1] ) {

// Al entrar aquí la capacidad necesariamente es negativa y no se terminara el modelo con dos
máquinas
        capacidad = (u->MaqPtr[inc+1][2] * TDI[inc+1]) - (r -> Cantid);
         if (capacidad >= 0 ) { /*El pedido se cubrio*/
                TDI[inc+1] = TDI[inc+2] - ( r->Cantid / u -> MaqPtr[inc+2][2] );
                r -> Cantid = r ->Cantid - (r->Cantid / u -> MaqPtr[inc+2][2] ) * u->MaqPtr[inc+1][2];
         }
        if (capacidad  < 0 ) {
                r -> Cantid = r -> Cantid - (u->MaqPtr[inc+1][2] * TDI[inc+1]);
                TDI[inc+1] = TDI[inc+1] - (capacidad / u->MaqPtr[inc+2][2]);
        }
        if ( u->MaqPtr[inc+2][2] > 0 )
        if ( (r->Cantid / u->MaqPtr[inc+2][2] ) <= TDI[inc+2] ) {
// Al entrar, indica que es suficiente con 3 máquinas para terminar el modelo

        capacidad = (u->MaqPtr[inc+2][2] * TDI[inc+2]) - (r->Cantid);
         if (capacidad >= 0 ) { /*El pedido se cubrio*/
                TDI[inc+1] = TDI[inc+2] - ( r->Cantid / u -> MaqPtr[inc+2][2] );
                r -> Cantid = r ->Cantid - (r->Cantid / u -> MaqPtr[inc+2][2] ) * u->MaqPtr[inc+1][2];
                }
          if (capacidad  < 0 ) {
// Al entrar aquí  indica que el modelo no se cubrió y por tanto existe un error

                r -> Cantid = r -> Cantid - (u->MaqPtr[inc+2][2] * TDI[inc+2]);
                TDI[inc+2] = TDI[inc+2] - (capacidad / u->MaqPtr[inc+2][2]);
```

```
            } // FIN: Se termina con dos máquinas
        }

        else if ( (r->Cantid / u->MaqPtr[inc+2][2] ) > TDI[inc+2] ) {
// Al entrar aquí la capacidad necesariamente es negativa y no se terminara el modelo con tres
máquinas

                capacidad = (u->MaqPtr[inc+1][2] * TDI[inc+1]) - (r -> Cantid);
                if (capacidad >= 0 ) { /*El pedido se cubrio*/ }

                if (capacidad  < 0 ) {
                        r -> Cantid = r -> Cantid - (u->MaqPtr[inc+2][2] * TDI[inc+2]);
                        TDI[inc+2] = TDI[inc+2] - (capacidad / u->MaqPtr[inc+2][2]);
                        } // FIN: Se requieren más de 3 máquinas para terminar
                    }
            }
        }
      u = u -> Next;
      }
   u = MaqSel -> Next;
   r = r -> Next;
  }
}

void ModsnoTerminados() {

    int id,jd;
    struct partmod *r, *t;

    for (id=0;id<modelos;id++)
        for (jd=1;jd<total_p+1;jd++)
            if ((Modelo_Pedido[id][jd] != 0) && (Reporte[id][jd] == 0)) { //Pedidos que se han
quedado sin asignar
                r = t = ModeNT -> Next;
                if (r == NULL) {
                    r = (struct partmod *)calloc(1, sizeof(struct partmod));
                    r -> Cantid = Modelo_Pedido[id][jd];
                    r -> Model = int(Modelo_Pedido[id][0]);
                    ModeNT -> Next = r;
                    r -> Next = NULL;
                    }
                else if (r != NULL) {
                     r = (struct partmod *)calloc(1, sizeof(struct partmod));
```

```
                r -> Cantid = Modelo_Pedido[id][jd];
                r -> Model = int(Modelo_Pedido[id][0]);
                ModeNT -> Next = r;
                r -> Next = t;
                }
            }
            return;
        }

void OrdenMaquinas() {

    struct partmod *r;
    struct lismaq *u,*v;
    int fh,fi,fk=0;

    r = ModeNT -> Next;

    while( r != NULL) {
            u= v = MaqSel -> Next;
            if (u == NULL) {
                    u = (struct lismaq *)calloc(1, sizeof(struct lismaq));
                    fk=0;
                    for (fh=0; fh<maquinas;fh++)
                        for (fi=0; fi<cap_maq; ++fi)
                                if ((r -> Model == int(Maquinas[fh][fi][0])) && (TDI[fh] > 0)) {
                                        if (fk<3) {
                                           u -> MaqPtr[fk][0] = fh;
                                           u -> MaqPtr[fk][1] = Maquinas[fh][fi][0];
                                           u -> MaqPtr[fk][2] = Maquinas[fh][fi][1];
                                           fk++;
                                           }
                                        }
                                        MaqSel -> Next = u;
                                        u -> Next = NULL;
                                }
                                else if (u != NULL) {
                                           u = (struct lismaq *)calloc(1, sizeof(struct lismaq));
                                           fk=0;
                                           for (fh=0; fh<maquinas;fh++)
                                                   for (fi=0; fi<cap_maq; ++fi)
                                             if ((r -> Model ==int (Maquinas[fh][fi][0])) && (TDI[fh] > 0)) {
                                                                if (fk<3) {
                                                                        u -> MaqPtr[fk][0] = fh;
```

```
                                                u -> MaqPtr[fk][1] = Maquinas[fh][fi][0];
                                                u -> MaqPtr[fk][2] = Maquinas[fh][fi][1];
                                                fk++;
                                                }
                                        }
                                MaqSel -> Next = u;
                                u -> Next = v;
                                }
                        r = r -> Next;
                        }
                return;
                }

void RendAscendente() {

    int id, jd, rentem, maqtem, modtem;
    struct lismaq *u;

    u = MaqSel -> Next;
    if ( u == NULL )
          printf("\nNO EXISTE NINGUN ELEMENTO EN LA LISTA\n");
    while( u != NULL) {
            for( jd=0; jd<3; ++jd )
                    for( id=2; id>0; --id )
                            if ( (u -> MaqPtr[id-1][2] > u ->MaqPtr[id][2]) && (u -> MaqPtr[id][2] > 0) &&
                            (u ->MaqPtr[id-1][2]>0)) {
                                maqtem = u -> MaqPtr[id-1][0];
                                modtem = u -> MaqPtr[id-1][1];
                                rentem = u -> MaqPtr[id-1][2];
                                u -> MaqPtr[id-1][0] = u -> MaqPtr[id][0];
                                u -> MaqPtr[id-1][1] = u -> MaqPtr[id][1];
                                u -> MaqPtr[id-1][2] = u -> MaqPtr[id][2];
                                u -> MaqPtr[id][0] = maqtem;
                                u -> MaqPtr[id][1] = modtem;
                                u -> MaqPtr[id][2] = rentem;
                                } // La Pila u debe ser MaqSel //MaqSel -> Next = u;
                            u = u -> Next;
                            } //while
                    }
```

Finalmente se presenta como resultado la asignación de los modelos a las máquinas que permita realizar la producción calendarizada en un tiempo óptimo.

4.3 El entorno del Sistema de Asignación de Cargas de Trabajo (SACT).

Actualmente el ambiente de un sistema por lo general se presenta mediante una interfase que es sencilla y amigable para el usuario. Por tal motivo en el presente sistema se esquematizaron accesos simples que permiten al usuario establecer la entrada de datos, ejecutar los procesos y obtener los resultados de una manera eficiente.

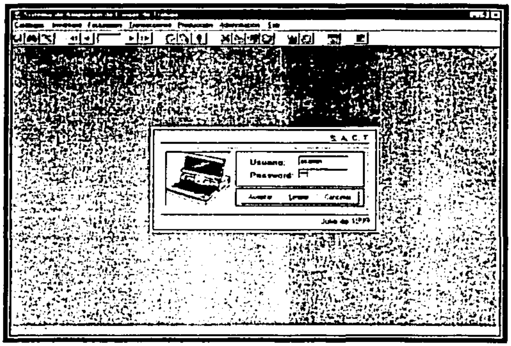


Fig. A. Pantalla de acceso del Sistema de Asignación de Cargas de Trabajo.

En seguida se presenta un ejemplo real que permite visualizar las principales características del sistema.

Como primer dato requerido se debe definir el periodo en el cual se debe programar la producción; generalmente éste es de un mes (la unidad de tiempo utilizada en el modelo de optimización es de una hora). La forma de acceso a los datos es mediante un cuadro de diálogo bajo la plataforma de Windows.

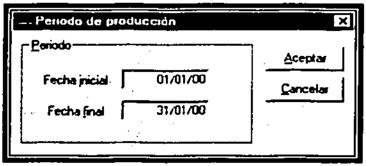


Fig. B. Cuadro de captura.

En seguida se deberá desplegar el total de modelos a producir, y que previamente fueron capturados en la solicitud de pedidos.

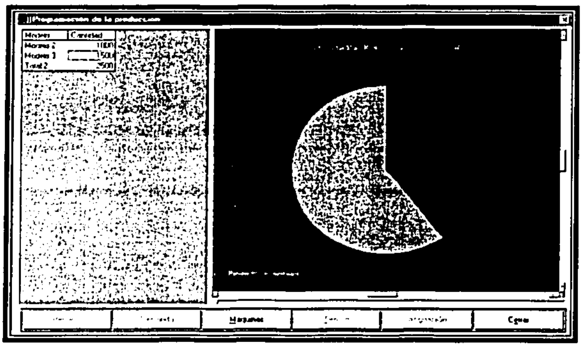


Fig. C. Total de pedidos a producir en un periodo.

El número de máquinas disponibles y sus respectivos rendimientos por modelo se presentan en una sola tabla como se muestra en seguida.

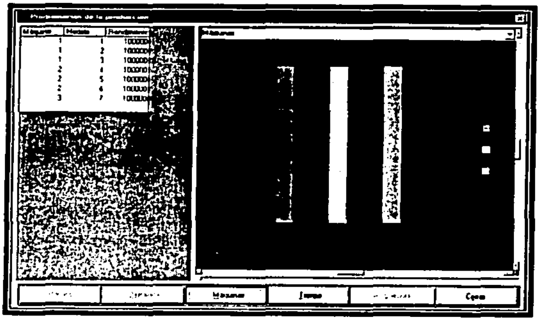


Fig. D. Relación de máquinas por Modelo-Rendimiento.

El tiempo disponible por máquina se actualiza con base en los modelos de producto asignados a ésta, como se muestra en el cuadro respectivo.

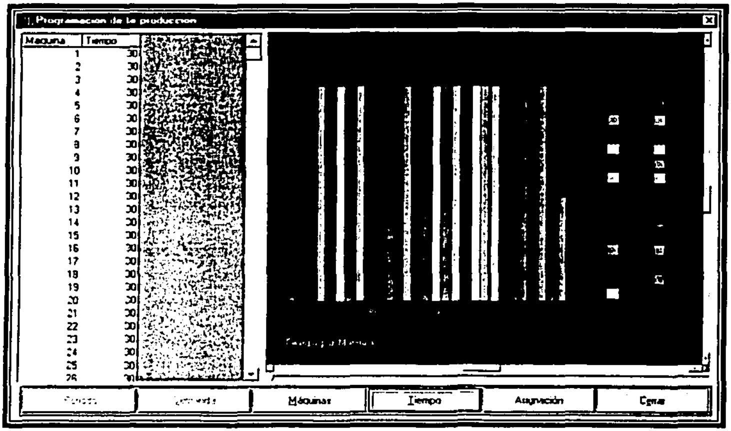


Fig. E. Relación de tiempo disponible por máquinas.

Por último, se realizan varias asignaciones, a fin de obtener la óptima. Para validar dicha acción se despliega el siguiente cuadro que indica la conclusión del proceso.

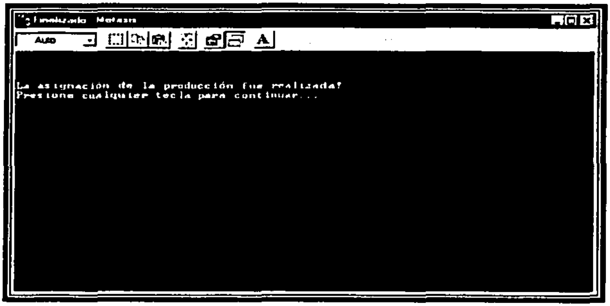


Fig. F. Conclusión del proceso.

En la obtención de resultados se genera una relación, lo que según el modelo es la asignación óptima de los modelos de producto solicitados a las máquinas disponibles y el tiempo en que se realiza la producción. Estos datos se presentan en la siguiente tabla:

Metales y contactos
Reporte de la Asignación de la Producción
Tiempo óptimo de la producción: 382.84 unidades de tiempo

MATRIZ DE ASIGNACION

Modelo	Máquina	Producción
1	11	200000
2	7	30000
4	9	7000
5	2	17000
6	8	8000
9	8	38000
10	21	80000
11	9	60000
12	0	97600
13	9	50000
14	16	20000
15	13	20000
17	7	5000
20	3	60000
21	7	60000
22	9	50000
23	9	30000
24	13	50000
25	9	10000
26	16	120000
27	1	10000
32	15	27500
35	21	42000
36	17	10000
37	14	300000
38	6	75000
40	6	50000

Modelo	Máquina	Producción
65	15	48000
66	13	50000
69	13	50000
70	0	1864173
71	21	1073950
74	18	218571
75	10	240000
79	10	450000
80	24	575000
82	21	160000
83	4	7952
86	13	15000
89	4	20000
92	23	400000
93	19	60000
94	24	500000
95	24	4008000
99	22	620000
102	8	120000
114	26	500000
119	26	616740
123	16	550000
171	18	75000
172	0	72000
179	0	500000
180	3	500000
185	17	2000

CONCLUSIONES.

La formación teórica en matemáticas y la formación general en computación aunadas a un poco de creatividad, hacen posible que los alumnos de la carrera de Matemáticas Aplicadas y Computación puedan crear modelos fuera de los convencionales, ya que al utilizar los modelos conocidos y ya probados en un problema particular puede resultar más costoso que elaborar un modelo nuevo.

Con base en los elementos teóricos recopilados en el presente trabajo y en el planteamiento del modelo matemático, se logró obtener una aproximación a la solución óptima de la asignación de las cargas de trabajo a las fuentes de producción de la empresa Metales y Contactos. Cabe destacar que resultaría muy interesante instrumentar un esquema de probabilidad al modelo de tal manera que se busque mejorar la solución al problema bajo estudio para que, de alguna manera, el planteamiento sea cada vez más apegado a la realidad.

De los temas que fueron tratados, sin lugar a dudas el tema de Modelos de Optimización fue el que nos pareció más interesante ya que pensamos no es muy complicado establecer un planteamiento de los factores que intervienen en un modelo de asignación. También es importante señalar que en la medida que se tenga un mejor conocimiento de cada uno de los temas, será posible generar más y mejores alternativas encaminadas a resolver un problema determinado.

Además del modelo matemático, se está aportando a la empresa una herramienta adicional muy útil, para la toma de decisiones: un sistema de información. De lo más importante que se puede mencionar de ambas herramientas es el hecho de que se pueden adaptar a cualquier problema de asignación semejante al que aquí se presenta, tomando como base el modelo general propuesto en el capítulo dos.

Con los resultados obtenidos a través de la implantación del modelo en el sistema, ahora se pueden presentar a la gerencia una serie de propuestas para mejorar la producción, con lo que se elimina la limitante de contemplar una sola alternativa para cubrir la demanda de los productos. El tiempo que anteriormente empleaba el personal de la compañía en asignar las cargas de trabajo a las distintas máquinas, ahora se destina a mejorar los procesos de producción.

Por otra parte, como el modelo matemático puede ser instrumentado en cualquier organización a través de un sistema de computadora, éste se vuelve sumamente rentable. Por lo mismo resulta también interesante tratar de contactar empresas que tengan problemas de planeación de la producción semejantes al de la organización en cuestión a fin de tratar de mejorar los procesos.

Entre las limitaciones del modelo está la de que, hasta el momento, no se pueden garantizar formalmente características de convergencia. Además, teóricamente tampoco se pueden garantizar cuestiones relativas a la rapidez de convergencia.

En el último ejemplo presentado se detectó que en la aproximación existe una máquina que el sistema no contempla para el proceso de producción, de igual manera se observó que hubieron tipos de producto que quedaron sin asignar. Por supuesto estas peculiaridades se pueden resolver introduciendo restricciones adecuadas en el modelo.

Finalmente, el éxito que se alcance con la aplicación del modelo matemático dependerá de la habilidad para establecer líneas de comunicación entre las fuentes de información y los individuos responsables de instrumentar las soluciones recomendadas.

BIBLIOGRAFÍA.

Luenberger, David E. Programación lineal y no lineal. Editorial Addison-Wesley Iberoamericana.

Intriligator, Michael D. Optimización Matemática y Teoría Económica. Editorial Prentice/Hall Internacional.

Gass, Saúl I. Operations Research Mathematics and Models. Editorial Providence Rhode Island.

Taha, Hamdy A. Investigación de Operaciones. Editorial Alfaomega Grupo Editor.

Hillier, Frederick S. y Lieberman, Gerald J. Investigación de Operaciones. Editorial Mc. Graw Hill.

Kaufmann, A. & Cruon, A. La Programación Dinámica. Editorial Continental.

Dantzig, George. Linear Programming and Extensions. Editorial Princeton.

Feller, W. Una Introducción a la Probabilidad y sus Aplicaciones. Editorial LIMUSA.

Shiltdt, Herbert. Introducción al Lenguaje C. Editorial Osborne/Mc Graw Hill.

Shildt, Herbert. Lenguaje C/Programación Avanzada. Editorial Osborne/Mc Graw Hill.