DESARROLLO E IMPLEMENTACION DE UN SISTEMA PARA OPTIMIZAR LA PLANIFICACIÓN DE LA PRODUCCIÓN

Mónica Rojas Ramírez[1], Juan Alvarado Ortega[2]
[1]Ingeniera en Estadística Informática 2004
[2]Director de Tesis, Ingeniero en Sistemas, Escuela Superior Politécnica del Litoral, 2004

RESUMEN

The Industry Ecuadorian Cartonera, is devoted to the production of sheets and boxes of cardboard corrugado and microcorrugado to satisfy the necessities of packing of a great number of products that belong to the national and international market directed to different such sectors as camaronero, fishing, banana, florist, domestic among others.

At the moment the Industry Ecuadorian Cartonera has come experiencing serious problems to fulfill the times of delivery of the orders, basically provoked for the planning in the production, since to fulfill most of having requested mainly with the accidental ones to it finishes hour that you/they are usually urgent, she/he modifies the planning of this production, resulting in wastes of material and resources, bigger time and stop of machines.

The objective of the present work consists to develop and to implant a system that optimizes the planning of the Industrial Production, by means of the quick and exhaustive implementation of search algorithms inside the theories of optimization combinatorias.

INTRODUCCIÓN

La manera más antigua de realizar la díficil tarea de planificar una producción cumpliendo con restricciones de tiempo, ha sido siempre por medio de la experiencia del recurso humano especializado en dicha area, pero este puede cometer algún error o demorarse en tener una planificación óptima.

Así que la idea de la implementación de este sistema surgió por la necesidad de contar con una herramienta que permita la planificación óptima y rápida en la Producción Industrial (Asignación de pedidos a las unidades de trabajo)

Por ello el sistema implementado en esta tesis, es una alternativa a esa necesidad, que brinda funcionalidad a partir de información y herramientas de desarrollo de bajo costo y mayor disponibilidad; proveyendo además, mayor velocidad de procesamiento, logrando así un mayor alcance al momento de obtener los resultados ya que permite al usuario no solo utilizar una solución óptima sino también, seleccionar otra solución factible, entre varias soluciones factibles.

CONTENIDO

1. ANALISIS DEL PROCESO DE PRODUCCION

1.1 DESCRIPCIÓN DEL PRODUCTO

Los productos que maneja la ICE son básicamente los elaborados de cartón.

La empresa ofrece láminas y cajas en pared sencilla y doble en flauta (tipo de corrugado) C, B y BC; cajas regulares para el sector doméstico e industrial, cajas troqueladas y láminas de microcorrugado para el sector floricultor, cajas de doble pared que soportan grandes pesos para la exportación de productos agrícolas con o sin recubrimiento impermeabilizante, además hacen cajas laminadas, exhibidores y divisiones interiores de cajas (aditamentos).

1.2 CARACTERÍSTICAS Y PROPIEDADES DE LOS COMPONENTES DEL PRODUCTO

El cartón está compuesto por los siguientes componentes:

- Papel liner (Liner D.B., Liner S.F. “C”, Liner S.F. “B”)
- Papel médium (Médium S.F. “C”, Médium S.F. “B”)
- Almidón.
- Tinta.
- Goma.

Flauta: Corrugado u Ondulación del papel médium, la misma que sirve de separador entre los liners. Hay diversos tipos de flautas, a continuación se muestran en la tabla 1.

Flauta	flautas por pie lineal	cm. de profundidad
A	33	0.167
B	47	0.097
C	39	0.142
E	90	

Tabla 1: *Flautas*

1.3 CARACTERÍSTICAS Y PROPIEDADES DEL PRODUCTO.

Las cajas de cartón corrugado pueden ser clasificadas según sus componentes, sus propiedades, sus formas y utilización:

a) Según el número de componentes (papel) hay dos tipos:

i. **Pared Simple**, el mismo que está conformado por dos papeles liner y un médium (flauta C).
ii. **Pared Doble**, que está conformado por tres papeles liner y dos médium (uno flauta B y otro flauta C).

b) Según las propiedades; tenemos diferentes tipos de tests, resultado de las combinaciones de los papeles liner y médium, según el tipo de pared. A continuación se muestran en las tablas 2 y 3

Test	Liner	Medium	Liner
125	125	146-150	125
150	125	146-150	175-186
175	175-186	146-150	175-186
200	205	146-150	205
250	205	146-150	300-337
275	300-337	146-150	300-337

Tabla 2: *Test de Pared Simple*

Test	Liner	Médium	Liner	Médium	Liner
350	205	146-150	205	146-150	205
394	205	146-150	205	146-150	300-337
405	300-337	146-150	205	146-150	300-337

Tabla 3: *Test de Pared Doble*

c) Según su forma y utilización :

i. Cajas Banano
ii. Cajas Domésticas

1.4 MÁQUINAS QUE INTERVIENEN EN EL PROCESO DE PRODUCCIÓN:

- Corrugadora
- Imprenta
- Rayadora y Cortadora
- Troqueladora, y Guillotina Bobst
- Mezcladora y Batidora

2. OPTIMIZACIÓN COMBINATORIA

2.1 RESUMEN

La optimización combinatoria es un campo vivo de la matemática aplicada, que estudia el modelado combinando técnicas de combinatoria, programación lineal, lineal-entera, la teoría general de problemas extrémales y la teoría de algoritmos, con el propósito de encontrar una solución algorítmica de problemas donde se busca optimizar (maximizar o minimizar) una función de varias variables definidas sobre un conjunto discreto, (la palabra combinatoria se refiere al hecho que únicamente existen un número finito de soluciones factibles).

Esta disciplina tiene numerosas aplicaciones a problemas que se presentan en la industria, logística, ciencias, ingenierías y en la administración de organizaciones.

2.2 PROBLEMAS P, NP Y NP-COMPLETOS

Para los problemas de carácter combinatorio, existen distintas formas de resolverlos, una de ellas es la búsqueda exhaustiva del conjunto de soluciones y así poder encontrar la óptima, es decir generar todas las soluciones factibles, calcular su costo respectivo asociado y de éstas elegir la mejor. Pero el tiempo de cálculo crece de manera exponencial de acuerdo al número de items del problema.

Podemos encontrar problemas en que se produce una explosión combinatoria (donde el tiempo de ejecución es no polinomial), de acuerdo al tamaño del problema, de los que solo se conocen algoritmos que encuentran una solución exacta en tiempos excesivamente largos.

Cuando nos enfrentamos a un problema concreto, habrá una serie de algoritmos aplicables. Se suele decir que el orden de complejidad de un problema es el del mejor algoritmo que se conozca para resolverlo. Así se clasifican los problemas, y los estudios sobre algoritmos que se aplican a la realidad. Estos estudios han llevado a la constatación de que existen problemas muy difíciles, problemas que desafían la utilización de los ordenadores para resolverlos. Estos problemas se clasifican en: Problemas P (los tratables que se pueden resolver con búsqueda exhaustiva) ; Problemas NP y NP-Completos (Intratables que deben resolverse con métodos de optimización combinatoria y algoritmos heurísticos).

2.3 HEURISTICAS DE ÉXITO (META-HEURISTICAS)

Los procedimientos Meta heurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los Meta heurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de: inteligencia artificial, evolución biológica y mecanismos estadísticos.

- Búsqueda Tabú
- Templado Simulado
- Algoritmos Genéticos
- GRASP

3. RECOCIDO SIMULADO (SIMULATED ANNEALING)

3.1 RESUMEN

El algoritmo heurístico denominado Recocido Simulado, (figura 3.1) es un método global de optimización combinatoria establecido a través de una analogía al proceso físico de recocido de sólidos. Cuando a un material fundido se le baja la temperatura muy lentamente sus partículas se agrupan en un arreglo donde la energía interna del sistema es mínima. Metrópolis desarrolló un algoritmo para simular la evolución de un sólido en un baño de calor, a una temperatura específica, a través de una caminata estocástica entre configuraciones de átomos. Esto es, el sistema pasa sucesivamente de una configuración a otra mediante el siguiente criterio: Estando en una configuración, i, con energía Ei, el sistema pasa a una nueva configuración, j, si $\Delta E \leq 0$ donde $\Delta E = Ej - Ei$; o si $P(\Delta E) > r$, donde $P(\Delta E) = \exp\{(Ei - Ej) / kB\ t\}$ es la probabilidad de aceptar el incremento ΔE en la energía del sistema (kB = 1.38 x 10-16 y t = temperatura) y $r \in \Re$ con $0 \leq r \leq 1$. La caminata se efectúa hasta el equilibrio térmico.
Usando la función de costo Z(xi) como la energía del sistema y definiendo la configuración de átomos mediante los valores del conjunto de variables {xi} se establece la analogía al sistema termodinámico. Mediante el algoritmo de Metrópolis se puede generar una sucesión de configuraciones en alguna temperatura efectiva, c = kB t, la cual no es otra cosa que un parámetro de control. Si c es descendido entre ejecuciones del algoritmo de Metrópolis se simula un proceso de recocido y las configuraciones generadas a lo largo del procedimiento tienden a la solución de costo mínimo.

3.2 ALGORITMO RECOCIDO SIMULADO

$INPUT(T_0, a, L, T_f)$

$T \leftarrow T_0$

$S_{act} \leftarrow Genera\_solución\_inicial$

$WHILE\ T \geq T_f\ DO$

$\quad BEGIN$

$\quad FOR\ \ cont \leftarrow 1\ TO\ L(T)\ DO$

$\quad BEGIN$

$S_{cand} \leftarrow Selecciona\_solución\_N(Sact)$
$d \leftarrow \cos to(S_{cand}) - \cos to(S_{act})$
$IF\ U(0,1) < e^{-d/T}\ OR$
$d < 0\ THEN\ S_{act} \leftarrow Scand$
END
$T \leftarrow a(T)$
END
$\{Escribe\_como\_soluión,\_la\_mejor\_de\_las\_S_{act}\_visitadas\}$

3.2.1 Temperatura Inicial, T_0

$$f = e^{-d/T_0} = e^{(-m/T_0)C(S_{act})} \qquad \Rightarrow \qquad T_0 = \frac{m}{-\ln(f)}C(S_{act})$$

Hemos considerado un $f = 13\%$ y un $m = 1\%$ peor que la actual, tendríamos entonces un:

$$T_0 = \frac{0.01}{-\ln(0.13)}C(S_{act}) \approx 0.005C(S_{act})$$

3.2.2 Velocidad de Enfriamiento

$L(T)$, número de iteraciones al menos igual a $card(\Omega)^2$

$a(T) = kT,\ 0 < k < 1.$ valores más convenientes para el parámetro k son los correspondientes al intervalo $[0.8, 0.99]$,

3.2.3 Temperatura Final, T_F

e	Error Estimado
q	Probabilidad de obtener una solución cuyo costo, menos el de la óptima sea ?
$n = card(\Omega)$	número de elementos del espacio de soluciones.

$$T_f < -\frac{e}{\ln(n-1) - \ln(q) + \ln(1-q)} \approx \frac{e}{\ln(n)}$$

4. FORMULACION DEL MODELO MATEMÁTICO

4.1 SCHEDULING (PLANEACION - ASIGNACION)

Scheduling es un problema de complejidad NP-completo o NP-Duro que se presenta en sistemas de planificación en la producción industrial, y planificación de transporte, etc. Es resuelto mediante la combinación de técnicas heurísticas locales y globales, que intentan reducir el espacio de búsqueda. Es un proceso de toma de decisiones que tiene como meta la optimización de uno o más objetivos. Consiste en asignar recursos limitados a tareas, con restricciones de tiempo. El resultado será la obtención de una solución que minimice el tiempo necesario para completar la ejecución de todas las tareas del problema.

4.2 FUNCION DE COSTO

$\boldsymbol{Q_{ij}}$**:** Cantidad en mts. de p_{ij}

$\boldsymbol{Q_{ij}}$**:** $q_{ij} * L_{ij}$; $\boldsymbol{i} = (1,\dots m \text{ pedidos}), \boldsymbol{j} = (1,2 \text{ corrugadoras})$

- $\boldsymbol{q_{ij}}$**:** cantidad en unidades de ítem o productos de p_{ij}
- $\boldsymbol{L_{ij}}$**:** largo de plancha en mts. de p_{ij} , y se obtiene con el siguiente calculo:
  - $2 * l_{ij} + 2 * a_{ij} + tol_{ij}$
  - $\boldsymbol{l_{ij}}$**:** largo del item o producto de p_{ij}
  - $\boldsymbol{a_{ij}}$**:** ancho del item o producto de p_{ij}
  - $\boldsymbol{to.l_{ij}}$**:** tolerancia del largo de la plancha, que depende del test de p_{ij}.
  - $\boldsymbol{p_{ij}}$**:** pedido i en la corrugadora j

$\boldsymbol{Tp_{ij}}$**:** Tiempo de Producción de p_{ij}

$\boldsymbol{Tp_{ij}}$**:** $Q_{ij} \;/\; W_j$; $\boldsymbol{i} = (1,\dots m \text{ pedidos}), \boldsymbol{j} = (1,2 \text{ corrugadoras})$

- $\boldsymbol{W_j}$: capacidad de producción de la corrugadora j dada en metros por hora.

$\boldsymbol{Tcp_{ij}}$**:** Tiempo de Cambio en Programación de p_{ij}

$$\boldsymbol{Tcp_{ij}}: \begin{cases} 15 \text{ min. ; si } trimaje_{ij} \neq trimaje_{i+1j} \\ 0 \text{ min. ; si } trimaje_{ij} = trimaje_{i+1j} \end{cases}$$

$\boldsymbol{trimaje_{ij}}$ **:** *flauta, test,* m^2 *de* p_{ij}

$\boldsymbol{T_{ij}}$ **:** Función de Tiempo de p_{ij}

$\boldsymbol{T_{ij}}$ **:** $Tp_{ij} + Tcp_{ij}$; $\boldsymbol{i} = (1,\dots m \text{ pedidos}), \boldsymbol{j} = (1,2 \text{ corrugadoras})$

C: Función de Tiempo para los pedidos asignados

$$\boldsymbol{C}: \sum_{J=0}^{2} \sum_{i=0}^{m} \boldsymbol{T}_{ij}; \quad \boldsymbol{i} = (1,\dots m \text{ pedidos}), \boldsymbol{j} = (1,2 \text{ corrugadoras})$$

Min. C

4.3 SOLUCIÓN INICIAL

Para obtener una solución inicial, se debe determinar cuales son los pedidos que deben ser asignados a cada unidad, para lo cual se debe considerar las características de las unidades. Se tiene dos Corrugadoras, la ***S & S*** y la ***Langston***, la primera es usada solo para corrugados de flauta ***C***, y la segunda para los corrugados de flauta ***B, C*** y ***BC***.

De manera que lo primero que se debe hacer es particionar el conjunto

$\boldsymbol{P} = \{ p_1, p_2, p_3, \ldots, p_n \}$ de pedidos en 2 subconjuntos,
$\boldsymbol{P} = \{ LSS, L \}$;

$\boldsymbol{LSS} = \{ p_1, p_2, p_3, \ldots, p_m \}$ y $\boldsymbol{L} = \{ p_1, p_2, p_3, \ldots, p_r \}$; $n = m+r$

Si $p_i \in LSS \Rightarrow (flauta.p_i = B \vee flauta.p_i = C \vee flauta\ p_i = BC)$

Si $p_i \in L \Rightarrow flauta.p_i = C$

Luego cada partición o subconjunto ***LSS*** y ***L*** se particionará en grupos por fecha de despacho y estos grupos deben ordenarse de menor a mayor.

$\boldsymbol{LSS} = \{ LSS_1, LSS_2, \ldots, LSS_p \}$ y $\boldsymbol{L} = \{ L_1, L_2, \ldots, L_q \}$;

$\boldsymbol{LSS_k} = \{ p_{k1}, p_{k2}, \ldots, p_{kn} \}$

Si $p_i, p_j \in LSS_k \Rightarrow FechaDespacho.\ p_i = FechaDespacho.\ p_j$; $k = (1,2, \ldots,p)$

$\boldsymbol{L_k} = \{ p_{k1}, p_{k2}, \ldots, p_{km} \}$

Si $p_i, p_j \in L_k \Rightarrow FechaDespacho.\ p_i = FechaDespacho.\ p_j$; $k = (1,2, \ldots,q)$

Luego serán agrupados o combinados por test, luego por flauta, y por último por medidas

$\boldsymbol{LSS_k} = \{ O_{k1}, O_{k2}, \ldots, O_{kn} \}$;

$\boldsymbol{O_{k1}} = \{ p_1, p_2, \ldots, p_a \}$ $\boldsymbol{O_{k2}} = \{ p_1, p_2, \ldots, p_b \}$,..., $\boldsymbol{O_{kn}} = \{ p_1, p_2, \ldots, p_c \}$;
$p = a+b+\ldots+c.$

Si $p_i, p_j \in O_k \Rightarrow (test.p_i = test.p_j \wedge flauta.p_i = flauta.p_j \wedge m^2.p_i = m^2.p_j)$;
$k = (1,2, \ldots,p)$

$\boldsymbol{L_i} = \{ O_{i1}, O_{i2}, \ldots, O_{im} \}$;

$\boldsymbol{O_{k1}} = \{ p_1, p_2, \ldots, p_d \}$ $\boldsymbol{O_{k2}} = \{ p_1, p_2, \ldots, p_e \}$,..., $\boldsymbol{O_{kn}} = \{ p_1, p_2, \ldots, p_f \}$;
$q = d+e+\ldots+f.$

Si $p_i, p_j \in O_k \Rightarrow (Test.p_i = Test.p_j \wedge Flauta.p_i = Flauta.p_j \wedge m^2.p_i = m^2.p_j)$;
$k = (1,2, \ldots,q)$

Al conjunto obtenido, lo denominamos ***Conjunto Solución Inicial***

5. DISEÑO E IMPLEMENTACIÓN

5.1 DISEÑO DE LA BASE DE DATOS

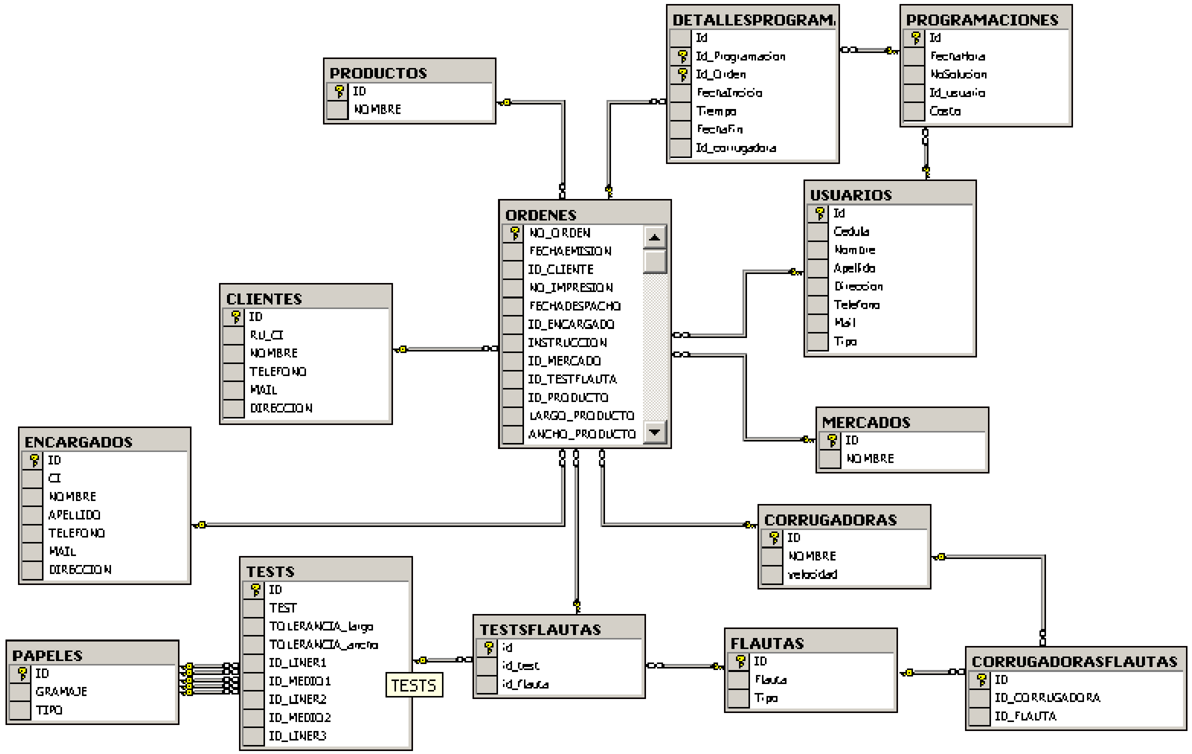

Figura 1. *Estructura de la base de datos.*

5.2 DISEÑO DE LA INTERFAZ DEL SISTEMA

Pantallas del Sistema:

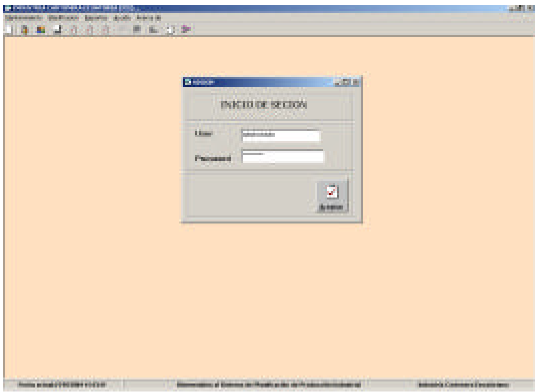

Figura 2. *Pantalla de Ingreso.*

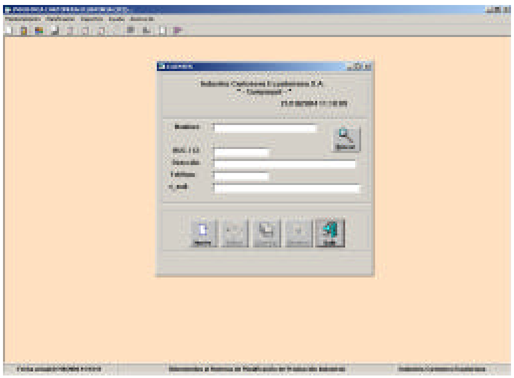

Figura 3. *Pantalla de Mantenimiento de Cliente*

La Programación de Ordenes existentes solamente le compete al usuario Administrador que son los encargados y responsables de planificar la programación de las ordenes dentro de una Producción. Esta sección se encuentra en la opción Programar dentro del Menú Planificación del Sistema.

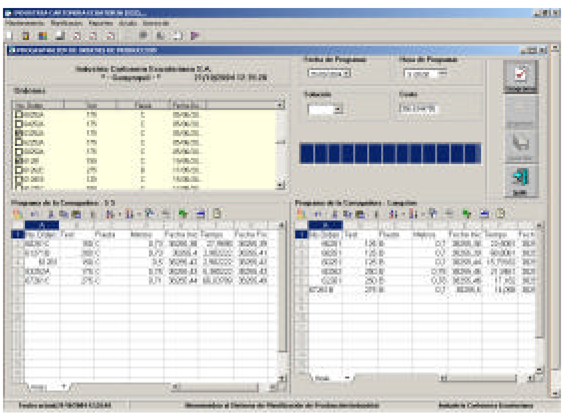

Figura 4. *Pantalla de Programación de Ordenes*

El usuario podrá visualizar en un list view todas las órdenes que dentro de la Base tienen un estado de Abiertas, es decir que aun no han sido programadas, de manera que el usuario tiene la opción de escoger las ordenes que desea programar, una vez seleccionadas, debe ingresar la fecha y hora en que desea que inicie la programación, finalmente da click en el botón Programar y los resultados de cada máquina denominados Solución Optima serán visualizados en los Web Components correspondientes a cada máquina; y además podrá observar el costo o tiempo que demorara en producir las ordenes bajo esa programación
Pero además el usuario tiene la opción escoger las otras soluciones que fueron aceptadas antes de la optima, según el algoritmo. Así el podrá determinar que solución le conviene de acuerdo al costo. Y dando clic en el botón Imprimir podrá visualizar las soluciones impresas y guardarlas en la Base de Datos con el botón Guardar. De manera que si posteriormente desea revisar las programaciones anteriores puede hacerlo en la opción Programas del Menú Reportes del Sistema.

CONCLUSIONES

- Como resultado del desarrollo de este proyecto de tesis tenemos un sistema de información que permite planificar una producción de una Industria, cumpliendo con ciertas restricciones de tiempo, requerimientos y especificaciones de la Industria. Es decir que este sistema permite la asignación óptima de los pedidos a las unidades de trabajo en la Producción Industrial.

- La Teoría de Optimización Combinatoria mediante la minimización de la función de costo que para nuestro caso es una función de tiempo; permite encontrar una solución factible para la asignación de pedidos en la Producción Industrial. La solución factible será una solución Inicial para la Implementación del Algorítmo Meta-Heurístico Recocido Simulado.

- La Algorítmo Recocido Simulado, necesita como parámetros: una solución Inicial, el numero de iteraciones, una solución final aproximada, un error estimado; mediante la calibración exacta de estos parámetros el algoritmo puede buscar una mejor solución hasta encontrar una solución óptima.

- Se recomienda que la implementación del algoritmo Recocido Simulado sea desarrollada en memoria, mediante la utilización de Clases (Listas Doblemente Enlazadas); debido a que por el número de iteraciones al que está sujeto el algoritmo, el desempeño de la implementación en el motor de la base de datos es mínimo y totalmente contrario al desempeño de la implementación en memoria.

REFERENCIAS

- Tesis: Estudio y Mejoras del Proceso de Planeamiento de la Producción Industrial
  Autor: Ana Fabiola Terán Alvarado

- WILLIAM R. VAUGHN, “Programación de SQL Server 7.0 con Visual Basic 6.0”

- BRIAN SILER – JEFF SPOTTS (1999), “Edición Especial Visual Basic 6”, Prentice Hall

- ADENSO DIAZ (1996), “Edición Paraninfo; Optimización Heurística y Redes Neuronales”