



A.F. 132585



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

**Facultad de Ingeniería en Electricidad y
Computación**

**“SISTEMA DE PLANEACIÓN AVANZADA Y SIMULACIÓN DE
SECUENCIAS DE PROCESOS PRODUCTIVOS”**

TESIS DE GRADO

Previa a la obtención del título de:

INGENIERO EN COMPUTACIÓN

Presentado por:

Carlos Bolívar Paredes Ortiz



GUAYAQUIL - ECUADOR

2006

Agradecimiento

A lo largo de mi etapa universitaria he sido guiado por diferentes catedráticos, pero contados maestros, como el Ing. Fabricio Echeverría.

Dedicatoria

*Esta tesis solo ha podido ser realizada por el apoyo incondicional
de mis seres queridos.*

A mis padres.....

Gracias por cobijarme a la sombra de vuestro ser.

*El tiempo, medida inexorable de la existencia, ha pasado pero no sin antes haberme
enseñado el milagro de la vida.*

Gracias a Dios

Seguiré adelante !!!!!

Todavía me queda por aprender.

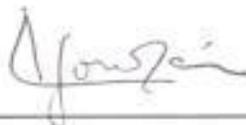
TRIBUNAL DE GRADUACIÓN



Ing. Holguer Cevallos
PRESIDENTE



Ing. Fabricia Echeverria
DIRECTOR DE TESIS

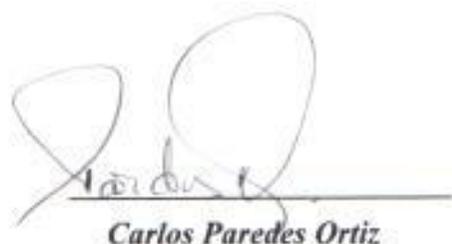


Ing. Carlos Jordán
MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.”

(Reglamento de Graduación de la ESPOL)



Carlos Paredes Ortiz

RESUMEN

En la actualidad, existen diversos métodos de optimización, algunos de los cuales son analizados por métodos exactos y otros que se han venido desarrollando desde los ochenta a partir de métodos heurísticos, éstos han generado una amplia gama de métodos conocidos como meta-heurística, los cuales tienen aplicación a diversos tipos de problemas, es decir es una “plantilla algorítmica” que se aplica en la resolución y optimización de diferentes aspectos.

La simulación y optimización aplicada en la secuencia de procesos productivos, es resultado, que en la actualidad no contamos con métodos diseñados para este tipo de procesos, de los cuales optimizamos la función de utilidad neta, es decir hacerla independiente del comportamiento de las variables aleatorias propias del proceso.

La simulación de procesos, es el resultado de diseñar un modelo de un sistema real, para realizar posteriores experimentos a partir de hipótesis. Por lo anterior resulta más eficiente experimentar sobre modelos basados en sistemas reales que en los mismos sistemas (es más eficiente por tiempo y dinero), por lo cual la combinación de ambas instancias SIMULACIÓN y OPTIMIZACIÓN da como resultado una herramienta poderosa, en la resolución de problemas de diversa naturaleza.

- 👉 En el capítulo 1, veremos los aspectos más relevantes sobre la simulación de procesos a partir de métodos estadísticos, como el Método Monte Carlo, enunciaremos las bases para la resolución de nuestro problema, como son la generación de números pseudos aleatorios aplicada a la formación de diferentes distribuciones, siempre considerando que las distribuciones son independientes entre sí.

- ✚ Capítulo 2, analizaremos el por qué de la optimización combinatorial, posteriormente se hará una reseña de los algoritmos meta heurísticos más conocidos así como el análisis de convergencia de estos (algoritmos meta heurísticos), el cual nos da como resultado una solución aunque no la más óptima resulta la más eficiente (tomando en consideración el tiempo y el costo para obtener dicha solución).
- ✚ En el capítulo 3, En base a los resultados que vamos ha encontrar y el procedimiento a simular seleccionaremos el algoritmo meta heurístico, El algoritmo Tabu Search nos ayuda en la exploración de una vecindad a través de sus matrices de memoria a corto y largo plazo, adicionalmente analizaremos una variante del algoritmo general denominado "Tabu con umbral". Enunciaremos algunos problemas específicos de aplicación del algoritmo.
- ✚ En el capítulo 4, nos enfocamos en el análisis y diseño de la aplicación, así como los escenarios donde se obtuvo dicha solución; tiempos de implementación de la aplicación y manual del usuario.
- ✚ Capítulo 5, en este trataremos la definición del ambiente en el cual deberá funcionar el algoritmo (planteamiento del problema), se definirá distribuciones estadísticas, la representación del problema, y posterior a esto veremos las ventajas y desventajas de los ARP vs. ERP.
- ✚ Capítulo 6, analizaremos los resultados de la simulación, partiendo de los resúmenes estadísticos y principalmente de los gráficos de sus funciones acumuladas (ojivas). Por último se realizará una comparación en función del tiempo de producción entre el sistema de simulación propuesto y el sistema real.

ÍNDICE GENERAL

	Pág.
RESUMEN.....	VI
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS.....	XI
INTRODUCCIÓN.....	I
CAPÍTULO I. GENERALIDADES DE LA SIMULACIÓN DE SISTEMAS.....	3
1.1 Simulación de sistemas.....	5
1.2 Simulación a eventos discretos y continuos.....	9
1.3 Simulación estadística: MÉTODO MONTE CARLO.....	11
1.4 Generación de números aleatorios.....	13
1.5 Sorteo de variables aleatorias discretas y continuas.....	15
CAPÍTULO II. OPTIMIZACIÓN COMBINATORIAL.....	18
2.1 Fundamentos y Aplicación.....	20
2.2 Programación Lineal.....	21
2.3 Algoritmos Meta-heurística.....	22
2.4 Factores de utilización.....	24
2.5 Convergencia: máximo y mínimos.....	25
CAPÍTULO III. META-HEURÍSTICA.....	27
3.1 Selección de algoritmo meta-heurístico.....	29
3.2 Algoritmo Tabú Search.....	29
3.3 Análisis del algoritmo.....	33
3.4 Aplicación del algoritmo.....	39

3.5	Herramientas: Simulación - Optimización.....	43
CAPÍTULO IV. APLICACIÓN: ANÁLISIS Y DISEÑO.....		44
4.1	Modelo de Análisis.....	46
4.2	Modelo de Diseño.....	80
4.3	Implementación: Arquitectura.....	91
4.4	Manual del Usuario.....	94
CAPÍTULO V. ESCENARIO.....		104
5.1	Definición del escenario.....	106
5.2	Planteamiento del problema.....	113
5.3	Representación de Heurística.....	115
5.4	Aplicación de la solución.....	118
5.5	Sistema de Planeación Avanzada.....	119
CAPÍTULO VI. ANÁLISIS DE SOLUCIÓN.....		121
6.1	Análisis Estadísticos de los resultados.....	123
6.2	Análisis Función Acumulada.....	130
6.3	Costos Relacionados: Simulación vs. Real.....	132
6.4	Pruebas y Resultados.....	135
Conclusiones y Recomendaciones.....		139
Anexos.....		144
Bibliografía.....		161

ANEXOS:

Anexo A: Introducción al análisis de datos y sus aplicaciones "Investigaciones Publicadas"

Anexo B: Modelo y Análisis Estadístico

Anexo C: Glosario.

ÍNDICE DE FIGURAS

	Pag.
Figura 1.1	Componentes del modelo.....6
Figura 1.2	Esquema del proceso experimental de la simulación.....7
Figura 1.3	Etapas de un estudio de simulación.....7
Figura 1.4	Tipos de simulación y su relación con el tiempo.....8
Figura 1.5	Tipos de modelo.....8
Figura 1.6	Clasificación de los modelos de simulación.....9
Figura 1.7	Simulación de modelos discretos.....10
Figura 1.8	Justificación método Monte Carlo.....13
Figura 1.9	Interpretación geométrica.....17
Figura 2.1	Clasificación de algoritmos.....24
Figura 2.2	Óptimo local.....26
Figura 3.1	Algoritmo TABU SEARCH.....31
Figura 3.2	Estrategias del algoritmo TABU SEARCH.....32
Figura 3.3	Combinación inicial.....41
Figura 3.4	Intercambio de ítems 5 y 6.....42
Figura 5.1	Técnicas de generación de variables aleatorias.....107
Figura 5.2	$f(x) = \lambda e^{-\lambda x}$108
Figura 5.3	Gráfico de plano polar.....112
Figura 5.4	Ingreso de condiciones.....116
Figura 5.5	Estructura TABU SEARCH.....117
Figura 5.6	Solución óptima.....118
Figura 5.7	Secuencia alterna.....118
Figura 6.1	Función de estabilización.....140

ÍNDICE DE TABLAS

	Pag.
Tabla # 1	Sistema de producción: Condiciones iniciales.....47
Tabla # 2	Implementación del sistema.....93
Tabla # 3	Sistema de producción: Condiciones iniciales.....125
Tabla # 4	Tiempo promedio de producción.....137

INTRODUCCIÓN

Conforme avanza la tecnología y el pensamiento científico retoma algoritmos, que inicialmente fueron diseñados para eventos específicos, dado su difícil manejo tecnológico para la época que aparecieron, observamos que éstos (algoritmos metaheurísticos) tienen cada vez más relevancia en la resolución de eventos de diversa naturaleza esto debido que en la actualidad ha evolucionado en gran medida la tecnología en hardware como en software. Aunque, como se detalla posteriormente los algoritmos metaheurísticos en algunos casos no proveen de la mejor solución, en la mayoría de ellos nos proporciona de soluciones que son óptimas aunque no necesariamente la más eficiente, por lo anterior es necesario validar la necesidad de obtener una solución en función de su tiempo y esfuerzo computacional.

Los algoritmos metaheurísticos proveen en corto tiempo de soluciones óptimas (no necesariamente eficientes), si a esto añadimos el proceso de simulación en el sistema a optimizar comprendemos que tenemos una herramienta eficiente en la resolución de estos procesos.

La simulación de los sistemas tiene su base en la estadística, y de ella obtenemos las bases de la simulación ¿cuándo y por qué simular un proceso?, es decir nos describe el marco referencial y los principios (escenarios) de la simulación de un proceso, independiente de la naturaleza de este. Uno de los grandes problemas en la simulación de sistema consiste en encontrar un método a través del cual podamos generar variables aleatorias independientes entre sí, como detallaremos en el capítulo I, adicional a esto; determinar una función probabilística que modele el comportamiento de las variables que interviene en el sistema, a partir de sus datos históricos.

Es necesario recalcar que el presente trabajo resalta el hecho, que por tiempo y costo financiero resulta más conveniente y óptimo de escoger las soluciones que nos entregan estos algoritmos metaheurístico, que definir complicadas estructura matemáticas las cuales son casi imposibles de interpretar dentro de un proceso de simulación, adicionalmente su implementación resulta altamente costoso en tiempo, dado que en ambos casos nos da como resultado soluciones con escaso margen de diferencia y en el mejor de los casos resulta la misma solución, por lo anterior se recomienda la utilización de este tipo de algoritmo metaheurísticos en la resolución de problemas con complicada interpretación matemática.

El análisis no solo comprenderá la entrega de la solución óptima por parte del algoritmo, sino adicionalmente, confirmaremos dicha solución a través de procedimientos estadísticos, como es la simulación de los procesos a intervenir en el sistema, es clave la idea de definir distribuciones estadísticas que encuadren el comportamiento de las variables de los procesos, así como la independencia de estas entre sí. Por lo cual se estudia el Método de MonteCarlo.

CAPÍTULO I

INTRODUCCIÓN

SIMULACIÓN DE SISTEMAS

I. INTRODUCCIÓN SIMULACIÓN DE SISTEMAS

Es importante observar el análisis de los costos en la cadena de producción, costos que implican cambios constantes de acuerdo a los procesos que están inmersos. Una de las falencias principales es el no contar con una herramienta que después del estudio del comportamiento de los procesos, determine los parámetros necesarios para obtener una utilidad óptima independiente del comportamiento del sistema (es decir en el peor de los escenarios, como en el más óptimo).

Entre los parámetros, más predominantes que resaltaremos son el número de personas relacionadas a cada proceso productivo, así como el funcionamiento de los diferentes equipos relacionados, (es decir contemplar el mejor de los escenarios equipos en constante operación). Por sentido común conocemos que el comportamiento humano varía de acuerdo a las condiciones laborales, así como su estado general (psicológico y físico), una analogía puede ocurrir con los diferentes equipos relacionados a la cadena de producción con la relevante que los equipos tendrán periodos de funcionamiento y mantenimiento preventivo así como periodos de reparación.

Es importante comprender que existen diversas funciones de distribución que van de acuerdo a las distintas etapas del proceso; en general se requiere obtener una función tal que independiente del "comportamiento" del sistema y considerando su costo, optimice su funcionamiento y por ende su utilidad, esto a través de modelos estocásticos que representen los tiempos de funcionamiento y mantenimiento, así como la secuencia de producción ítem - máquina, en un modelo abstracto a partir de un sistema determinado.

"Sin embargo muchos modelos no se pueden resolver metódicamente, a pesar de hipótesis fuertes o cálculos aproximados, por lo que se recomienda utilizar simulación cuando el sistema no se puede resolver analíticamente." [53]

1.1 Simulación de sistemas.

“Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con el mismo con la finalidad de comprender el comportamiento del sistema o de evaluar nuevas estrategias – dentro de los límites impuestos por un criterio o conjunto de ellos, para el funcionamiento del sistema.” R. S. Shannon. [54]

Con el avance de la tecnología, los computadores tienen gran ventaja de que pueden manejar una gran cantidad de datos, eficientemente, no obstante la simulación es independiente de estas funciones. Cualquier modelo de decisión empresarial, representa una simulación de un problema real.

Consideremos la construcción de un modelo de una fábrica que elabora una serie de productos. “Una programación lineal, podría desarrollar una combinación óptima de productos.”

Nuestro modelo, detalla los aspectos específicos de cómo se programaría la fábrica para obtener la combinación deseada, teniendo en cuenta los tiempos de preparación de las máquinas, el tiempo de espera antes del procesamiento y otros detalles que no se pueden incluir en la formulación de la programación lineal.

Antes de continuar definamos los componentes del modelo:

VARIABLES: las variables del sistema, estas se clasifican en variables de entrada (independientes o explicativas) y variables de salida (dependientes o de respuesta).

RELACIONES: las relaciones entre las variables, pueden venir dada por ecuaciones diferenciales.

SISTEMA: "Es un conjunto de objetos o componentes unidos que interactúan dentro de ciertos límites establecidos para obtener algún objetivo en común"

ENTIDAD: Sirve para recalcar un objeto de interés del sistema. Las órdenes que llegan a una fábrica son por ejemplo de una entidad. Las máquinas son entidades.

ESTADO DEL SISTEMA: Es empleado para expresar una descripción de todas las entidades, actividades y atributos en la forma como existen en un momento cualquiera. El progreso del sistema es estudiado siguiendo los cambios en el estado del sistema.

EVENTOS: Al igual que las entidades tienen atributos. En efecto, cada evento debe tener un atributo que defina el tiempo en que va a ocurrir.

1.- Elementos (físicos) del sistema

- ✚ ENTIDADES (piezas, ensambles, productos, clientes) *entities*
- ✚ RECURSOS (operarios, máquinas, capacidades, etc.) *resources*
- ✚ LÍNEAS DE ESPERA (colas, inventarios) *queues*

2.- Elementos (operacionales) del sistema

- ✚ ATRIBUTOS (de entidades)
 - VARIABLES (cantidades asociadas al sistema como un todo, pueden denotarse por un número, una matriz o una expresión)

CREADAS POR EL USUARIO

- ✚ EVENTOS (arribos, fallas, paros programados, cambios de recursos, etc.)

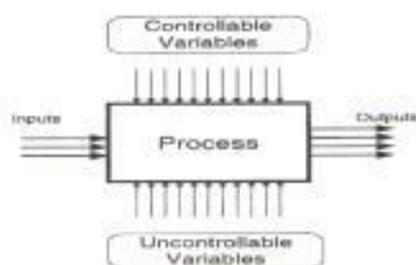


Fig. 1.1 COMPONENTES DEL MODELO

Entonces, para realizar una simulación, existen varias tareas muy importantes que son:

- 1.- La identificación de las entidades y los atributos, y
- 2.- La codificación de los valores de los atributos para poder caracterizar los estados del sistema.

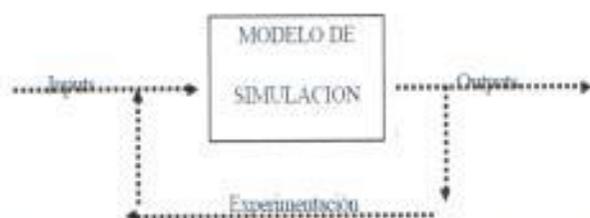


Fig. 1.2 ESQUEMA DEL PROCESO EXPERIMENTAL DE LA SIMULACIÓN

El proceso no es, en general, secuencial, sino interactivo, según se refleja en la figura, en el se refleja que algunos pasos deben repetirse en función de los datos intermedios.

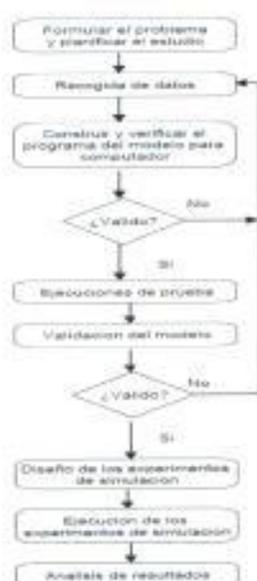


Fig. 1.3 ETAPAS DE UN ESTUDIO DE SIMULACIÓN

TIPOS DE SIMULACIÓN

Las técnicas de simulación se dividen en cuatro grupos:

- Simulación de sistemas continuos.
- Simulación de procesos por lotes (eventos discretos).
- Simulación combinada.

📌 Simulación de Monte Carlo.

Siendo el tiempo una variable continua, por ende; la simulación de un sistema real se debería manejar de forma completamente continua. A continuación se muestra un diagrama de los distintos tipos de simulación, en función del tiempo.



Fig. 1.4 TIPOS DE SIMULACIÓN Y SU RELACIÓN CON EL TIEMPO [5]

TIPOS DE MODELOS

“Un modelo es una abstracción de la realidad que captura la esencia funcional del sistema, con el detalle suficiente como para que pueda utilizarse en la investigación y la experimentación en lugar del sistema real, con menos riesgo, tiempo y coste.”

Existe una gran cantidad de técnicas de modelado, y por ello es posible construir una gran cantidad de modelos para un sistema dado

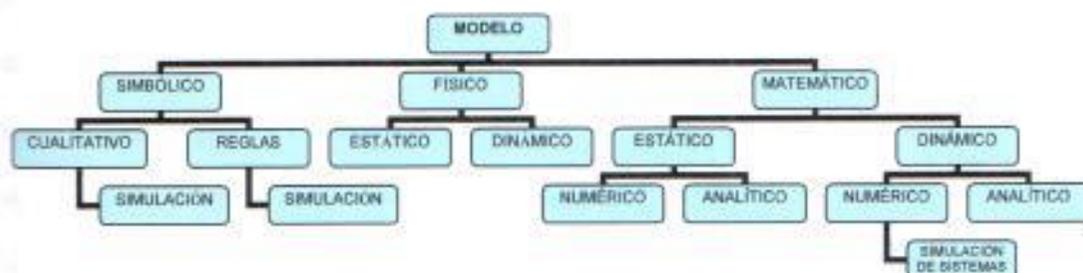


Fig. 1.5 TIPOS DE MODELO [4]

📌 Modelo determinístico

Modelo estocástico



Fig. 1.6 CLASIFICACIÓN DE LOS MODELOS DE SIMULACIÓN [4]

Puede ser complicado y en ciertos casos imposible, calcular las distribuciones de las variables de salida, incluso aquellas distribuciones sencillas y aun en el caso de hallar estas distribuciones, hallar las relaciones de salida con ellas puede ser complicado.

1.2 Simulación a eventos discretos y continuos.

En estudios previos hemos analizado simulaciones estáticas (donde las variables aleatorias no dependen del tiempo), en el caso de simulaciones dinámicas es importante definir:

Estado: característica relevante del sistema

Eventos: acciones que pueden producir cambios de estado.

En la estadística la información puede ser recolectada a través de métodos cuantitativos o cualitativos, los datos cuantitativos que consiste en medidas que toman valores numéricos, en donde la media y la desviación estándar toma sentido, pueden ser puestos en orden y subdivididos en grupos: continuos y discretos

Los datos discretos son recolectados por conteo.

Los datos continuos son recolectados por medición y se expresan en escala continua.

Por lo anterior se desprende que la simulación a eventos discretos puede ser analizada por una cadena de eventos y en definitiva esta consiste en seguir los cambios en los estados del sistema producidos por la serie de eventos. A diferencia el modelamiento de sistemas

continuos, se lo realiza a través de ecuaciones diferenciales las cuales se resuelven simultáneamente y del cual su principal interés son los cambios suaves.

Como se subrayó anteriormente, la simulación de un sistema real depende del tiempo, ésta (simulación) debe proveer de un mecanismo que, de acuerdo al transcurso del tiempo, deberá moverse de un evento a otro, por lo que, una de las ventajas en la simulación de sistemas reales, sería que en cuestión de segundos de un sistema simulado por computadora tendríamos los resultados que tardaría varios años en obtener, analizando sistemas reales.

Existen varios métodos para llevar un control del tiempo llamado "Control del flujo del tiempo", entre los cuales tenemos:

- El reloj avanza cuando ocurre el siguiente evento, llamado "mecanismo orientado evento".
- El reloj avanza en intervalos pequeños y en el cual se determina si debe o no debe ocurrir el evento, llamado "mecanismo orientado intervalos"

En el caso de sistemas con eventos discretos, se requiere llevar una lista de los tiempos en los cuales sucederán los próximos eventos. Para sistemas que cambian continuamente como aquellos dirigidos por ecuaciones diferenciales, este procedimiento no puede ser aplicado, por lo tanto adicional de los tiempos de eventos futuros se requiere monitorear los cambios del sistema (estado).



Fig. 1.7 SIMULACIÓN DE EVENTOS DISCRETOS

A diferencia, en los modelos de sistemas continuos, donde cuyo interés principal son los cambios suaves, para lo cual nos ayudamos de ecuaciones diferenciales que se resuelven simultáneamente. Posteriormente revisaremos la generación de variables aleatorias continuas.

1.3 Simulación estadística: MÉTODO MONTE CARLO

Frecuentemente es útil simular las características de un experimento antes de llevarlo a cabo. Esto nos ayuda a decidir el tamaño de los errores permitidos para observar un efecto en particular. La técnica de Monte Carlo es un formalismo para generar números con una distribución probabilística, ya definida, que simule los resultados de una secuencia de eventos, de variables discreta o continuas, dado que en la actualidad existe un sin número de programas comerciales que poseen generador de números aleatorios con probabilidades predefinidas.

Von Neuman y Ulam publicaron un artículo titulado "The Monte Carlo method" en 1949, los primeros ensayo de simulación que se realizaron en los Estados Unidos bajo el nombre de análisis Monte Carlo fueron en el año de 1940. El método ya era conocido en estadística de ahí su nombre de Simulación Estadística o Método Monte Carlo.

Las primeras aplicaciones se hacen en base al comportamiento aleatorio de los neutrones, métodos numéricos permiten resolver problemas mediante la simulación y el muestreo de variables aleatorias. El método de Monte Carlo es eficaz en la solución de problemas en los que se necesita conocer los resultados con una precisión del 5 al 10% (intervalo de confianza del 95% al 97.5%), la cual (precisión) se puede mejorar con técnicas de reducción de la varianza.

Todo experimento debe generar muestras independientes y plenamente identificables para aplicar el teorema del límite central, se garantiza la independencia aplicando semillas diferentes (posteriormente explicaremos la generación de números aleatorios)

Detalle del algoritmo "Simulación Monte Carlo"

1. Diseñar el modelo lógico de decisión
2. Especificar distribuciones de probabilidad para las variables aleatorias relevantes.
3. Incluir posibles dependencias entre variables.
4. Muestrear valores de las variables aleatorias
5. Calcular el resultado del modelo según los valores del muestreo (iteración) y registrar el resultado
6. Repetir el proceso hasta tener una muestra estadísticamente representativa N veces (independientes entre si)
7. Obtener la distribución de frecuencias del resultado de las iteraciones
8. Calcular media, desvío y curva de percentil acumulado

Sea X una v.a. cuya esperanza

$$E(X) = m \text{ y varianza } \text{Var}(X) = \sigma^2$$

Se toma una secuencia de n variable aleatoria X_i ,

$$E(X_i) = m \text{ y } \text{Var}(X_i) = \sigma^2$$

Por el teorema del límite central la v.a.

$$Z = X_1 + X_2 + X_3 + \dots + X_n$$

Se aproxima (asintóticamente igual) a una variable aleatoria de distribución normal $N(nm, n\sigma^2)$

La "regla de las 3σ ", en Y v.a. $N(a, \sigma^2)$:

$$\int_{a-3\sigma}^{a+3\sigma} f_Y(t) dt = 0,997$$

$f_Y(t)$ la función de densidad de la v.a. Y ,

$$P(a - 3\sigma \leq Y \leq a + 3\sigma) \approx 0,997$$

Justificación Método Monte Carlo

$$\begin{aligned} P\left(nm - 3\sigma \sqrt{n} \leq Z \leq nm + 3\sigma \sqrt{n}\right) &= 0,997 \\ P\left(m - \frac{3\sigma}{\sqrt{n}} \leq \frac{Z}{n} \leq m + \frac{3\sigma}{\sqrt{n}}\right) &= 0,997 \\ P\left(-\frac{3\sigma}{\sqrt{n}} \leq \frac{Z}{n} - m \leq \frac{3\sigma}{\sqrt{n}}\right) &= 0,997 \\ P\left(\left|\frac{Z}{n} - m\right| \leq \frac{3\sigma}{\sqrt{n}}\right) &= 0,997 \\ P\left(\left|\frac{\sum_{i=1}^n X_i}{n} - m\right| \leq \frac{3\sigma}{\sqrt{n}}\right) &= 0,997 \end{aligned}$$

Fig. 1.8 JUSTIFICACIÓN MÉTODO MONTE CARLO

Por lo tanto, en un mismo modelo se pueden usar algunas distribuciones de probabilidad para reflejar el carácter estocástico del sistema analizado y otras distribuciones de probabilidad para reflejar la incertidumbre. Los resultados que se obtengan de un modelo de este tipo reflejarían la incertidumbre total: el efecto conjunto de la variabilidad y la incertidumbre.

1.4 Generación de números pseudos aleatorios.

La generación de números aleatorios tiene un rol importante en los procesos de simulación, para lo cual necesitamos números aleatorios como semillas para tomar las muestras de las variables aleatorias.

Características de un generador de números aleatorios son:

- ✚ Valores de distribución uniforme: Cualquier número dentro del rango deberá tener la misma probabilidad
- ✚ Asegura inexistencia de correlación serial: Un número en la secuencia, no afecta la probabilidad de seleccionar otro número o el mismo.

En la actualidad existen tres tipos de procedimientos:

- ✚ Utilización de tablas
- ✚ Dispositivos especiales

Tablas

Se generan en base de métodos aleatorios puros (ruletas, dados, etc.), entre sus desventajas es una sucesión de números finitos, se tienen que cargar toda la tabla en la memoria del computador.

Dispositivos Especiales

En base de un circuito o mecanismo de computador, se generan los números que son totalmente aleatorios, básicamente consiste en detener en forma aleatoria un proceso uniforme, por lo tanto son números aleatorios puros, el mayor inconveniente es que no se puede generar la misma secuencia más de una vez, por lo que si se requiere repetirla es necesario grabar la corrida, en dispositivos de almacenamiento externos (tape, cd, etc.).

Procedimientos que generan números pseudos aleatorios

Son pseudos aleatorios porque se generan mediante funciones, y se usan como semillas para generar valores de variables aleatorias, entre sus ventajas son rápidas y de bajo costo, pero son de periodo finito.

En general, la secuencia de números generados debe contemplar:

- Distribución uniforme.
- Algoritmos rápidos.
- No correlación serial.
- Secuencias largas, sin saltos.

Una vez obtenida la semilla, procedemos a generar la secuencia, dado que es mucho más difícil obtener una secuencia de números aleatorios puros, generamos cadenas de números pseudos aleatorios. "Generador de números pseudos aleatorios es un dispositivo que recibe como entrada una semilla S , que supone aleatoria, y da como salida una cadena de bits de longitud n ".

Existen varias técnicas para probar la aleatoriedad de un generador de números pseudos aleatorios, estas verifican que las cadenas pseudos aleatorias tengan un número "igual" de ceros y unos. Finalmente se acepta que la diferencia es insignificante si la muestra cae dentro del intervalo de aceptación.

1.5 Sorteo de variables aleatorias discretas y continuas.

Entre las fuentes de información para cuantificar la incertidumbre en variables aleatorias tenemos:

1. **Serie de datos**
2. **Opinión de expertos** (disponible por el jefe de planta)

Definimos como distribución de probabilidad la que describe el rango de valores que puede tomar una variable aleatoria y la probabilidad asignada a cada valor o rango de valores.

Discreta

Una variable aleatoria representada mediante una distribución discreta de probabilidad puede tomar un valor de entre un conjunto de valores, cada uno de los cuales tiene asignada una determinada probabilidad de ocurrencia.

Ejemplos: Binomial, Geométrica, Poisson, Discreta.

Sorteo de una v.a. X,

- 1.- Se toma un valor $u = U(0,1)$.
- 2.- Tomamos el punto $y = u$.
- 3.- Si este punto aparece en el intervalo correspondiente al número i aceptamos que $X = x_i$ en este sorteo.

La probabilidad de que u pertenezca a uno de los intervalos es igual a la longitud del mismo.

$$P(0 < u < p_1) = p_1$$

$$P(p_1 < u < p_1 + p_2) = p_2$$

$$P(1 - p_n < u < 1) = p_n$$

Continua

Una variable aleatoria representada mediante una distribución continua de probabilidad puede tomar cualquier valor dentro de un rango determinado.

Ejemplos: Normal, Exponencial, Uniforme, Triangular, etc.

Se quiere generar una v.a. X continua: $F_X(x) = \int_{-\infty}^x f_X(t) dt$

Sorteamos una v.a. U de distribución U(0,1),

$$U = F_X(X) \quad X = F_X^{-1}(U).$$

Generamos X a partir de valores de U(0,1) con los métodos de sorteo de número aleatorios o pseudos aleatorios.

$$U = F_X(X) \quad X = F^{-1}(U)$$

El problema se reduce a encontrar una expresión analítica de la función inversa.

Problema: aún cuando se encuentre la expresión analítica de la inversa de la distribución, su cálculo puede recurrir métodos específicos numéricos.

Transformación inversa

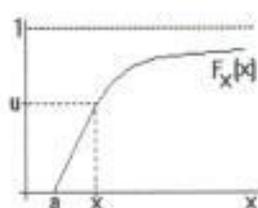


Fig. 1.9 INTERPRETACIÓN GEOMÉTRICA

Posteriormente, en el capítulo IV, sección 4.1 Definición de escenarios, analizaremos cada una de las funciones anteriormente señaladas

CAPÍTULO II

OPTIMIZACIÓN COMBINATORIAL

II. OPTIMIZACIÓN COMBINATORIAL

Existen problemas que solo tratan de satisfacer restricciones y otros la optimización de los procesos relacionados, nuestro problema en particular se basa en analizar ambas consideraciones definiendo prioridades entre la optimización y la satisfacción de restricciones.

“Se dice que un determinado problema es de optimización combinatoria si consiste en encontrar entre una cantidad finita de subconjuntos de un conjunto dado, aquel que cumpliendo con ciertas restricciones maximice o minimice una cierta función objetivo. Una forma de resolver esta clase de problemas consiste en evaluar la función objetivo para todas las soluciones válidas (enumeración o fuerza bruta). Lamentablemente, este método, si bien en teoría parece sencillo, es tan costoso de realizar (en términos de tiempo) que aún con las computadoras con mayor poder de cálculo tomaría años o siglos resolver problemas de mediana complejidad.” [53]

En la mayoría de los casos, el tiempo que necesita para resolver una instancia, crece de tal forma, que en el peor de los escenarios aumenta de manera exponencial con respecto al tamaño inicial.

Por lo cual, muchas veces se tiene que solucionar usando métodos de solución aproximados, que regresan soluciones aceptables (inclusive óptimas) a bajo costo computacional.

La heurística se puede usar para resolver problemas de optimización. Muchas de las estrategias de búsqueda podrían implementarse, aunque algunas pueden resultar de procesamiento costoso o quedar apresadas en mínimos locales, como lo analizaremos

2.1 Fundamentos y Aplicación.

Como es conocida la evaluación y la búsqueda es el objetivo principal en la Inteligencia Artificial, "Las heurísticas son criterios, métodos, o principios para decidir cual de varias alternativas de acción de búsqueda promete ser la más efectiva para cumplir con un objetivo". Por lo cual concluimos que toda búsqueda esta relacionada con la heurística, en problemas múltiples de optimización, la heurística se vuelve imprescindible para reducir el costo computacional. Entre los problemas mas estudiados tenemos:

- ▶ Localización
- ▶ Problema del agente viajero
- ▶ Problema del cartero
- ▶ Secuenciamiento
- ▶ Ruteamiento
- ▶ Distribución
- ▶ Redes de camino

Adicional, a esto podemos definir un "criterio de aceptación o tolerancia", dado que existe una gran probabilidad, se tiene el problema de optimización aproximada, por lo cual se debe establecer un equilibrio entre la calidad de la solución y el costo de hallarla. Como valor agregado contemplamos en nuestra solución la opción de un algoritmo con parámetros que ajusten la calidad y el costo de la solución.

Algunos problemas tratan de procesos de optimización, mientras en otros analizan satisfacer restricciones, La mayoría de los problemas son de optimización parcial, estableciendo un balance razonable entre calidad de la solución y costo de solución.

"En problemas complejos, las heurísticas juegan un papel fundamental para reducir el número de evaluaciones y para obtener soluciones dentro de restricciones de tiempo razonables. Las heurísticas no garantizan la acción más efectiva, pero muchas veces lo hacen." [54]

2.2 Programación Lineal.

Sea $f(x)$, optimizar la función con las restricciones:

$$\begin{cases} h_i(x) \leq b_i, i = 1..l \\ h_i(x) \geq b_i, i = l+1..m \\ h_i(x) = b_i, i = m+1..n \end{cases}$$

Como observamos es una función lineal, para la cual existe un método exacto para su resolución, el anterior es un problema donde se puede considerar una resolución del método simplex. Las técnicas existentes se pueden clasificar básicamente en algoritmos exactos o aproximados. Los algoritmos exactos intentan encontrar una solución óptima y demostrar que la solución obtenida es de hecho la óptima global; éstos incluyen técnicas como, por ejemplo: procesos de vuelta atrás (*backtracking*), Ramificación y poda (*branch and bound*), programación dinámica, etc. [84, 13].

Métodos de Programación Lineal

- ▶ **Métodos de Resolución** en caso que todas las funciones (la función objetivo y las restricciones) sean lineales de la forma $h_i(x) = a_j x_j$ con $a_j x_j \in \mathbb{R}$
- ▶ **Algoritmo Simplex** encuentra la respuesta, solamente que se demora demasiado en su ejecución

Debido a que los métodos exactos muestran un rendimiento pobre en la resolución de muchos problemas, por lo cual sean desarrollado múltiples tipos de algoritmos aproximados que generan soluciones de alta calidad para problemas combinatorios (aunque no necesariamente la óptima) en un costo de tiempo corto. Los algoritmos aproximados se pueden clasificar en dos tipos principales: *algoritmos constructivos* y *algoritmos de búsqueda*

local. Los primeros se basan en generar soluciones desde cero añadiendo componentes a cada solución paso a paso. Un ejemplo, son las *heurísticas de construcción voraz* o heurísticas *greedy* [13]. Su gran ventaja es la velocidad; normalmente son muy rápidos y, además, a menudo devuelven soluciones buenas. Sin embargo, no puede garantizarse que dichas soluciones sean óptimas con respecto a pequeños cambios. En consecuencia, una mejora típica es depurar la solución obtenida por la heurística voraz utilizando una búsqueda local.

2.3 Algoritmos Meta-heurística.

En los problemas combinatorial, existe siempre el procedimiento elemental, para determinar la solución óptima requerida. Entre los cuales tenemos el método de explosión exhaustiva (genera todas las soluciones posibles)

"Heurística (Definición) Son procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no sea la óptima) a problemas difíciles, de un modo fácil y rápido..." [54]

A continuación enunciaremos los diferentes tipos de algoritmos heurísticas:

- **Métodos Constructivos:** Consiste en ir agregando componentes individuales a la solución hasta que se obtiene una solución viable.
- **Métodos de Descomposición:** Se trata de dividir el problema, para que la salida de una parte de solución sea el ingreso de la otra parte del problema
- **Métodos de Reducción:** Tratan de identificar algunas características que presumiblemente deba poseer la solución óptima y de ese modo simplificar el problema.
- **Manipulación del Modelo:** Modifican las estructuras del modelo con el fin de hacerlo mas sencillo de resolver, deduciendo, a partir de su solución, la solución del problema original. Por ejemplo: Linealizar el modelo.

- **Métodos de Búsqueda de Entorno:** Parten de una solución inicial (obtenida mediante otra heurística) y mediante iteraciones de esta solución obtiene otras soluciones de forma iterativa, hasta que se obtenga el cumplimiento del criterio de parada. Por ejemplo: El método de aproximación de Newton

Entre los algoritmos meta-heuristicos más estudiados tenemos:

- Redes Neuronales
 - *Grasp*
 - *Ant Colony*
 - *Comportamiento Emergente*
 - *Scatter Search Estático (ES)*
 - *Scatter Search Dinámico (DI)*
 - Variante de *Scatter Search Dinámico (VD)*
 - *Algoritmo Memético (ME)*
 - *Algoritmo de propagación hacia atrás usado en Neural Works Profession II/Plus.*
Sexton y otros, (1998 BP)
 - *Algoritmo Búsqueda Tabú* propuesto en Sexton y otros (1999 TS)
 - *Algoritmo Temple Simulado* propuesto en Sexton y otros (1999 SA)
 - *Algoritmo Genético* propuesto en Sexton y otros (1999 GA)
 - Adaptación de *Scatter Search* propuesta por Laguna y Marti (2002 SS)
-

La meta-heurística incorpora nociones de muchos y diversos campos como Matemáticas, Física, Genética, Biología, e inteligencia artificial, entre otras

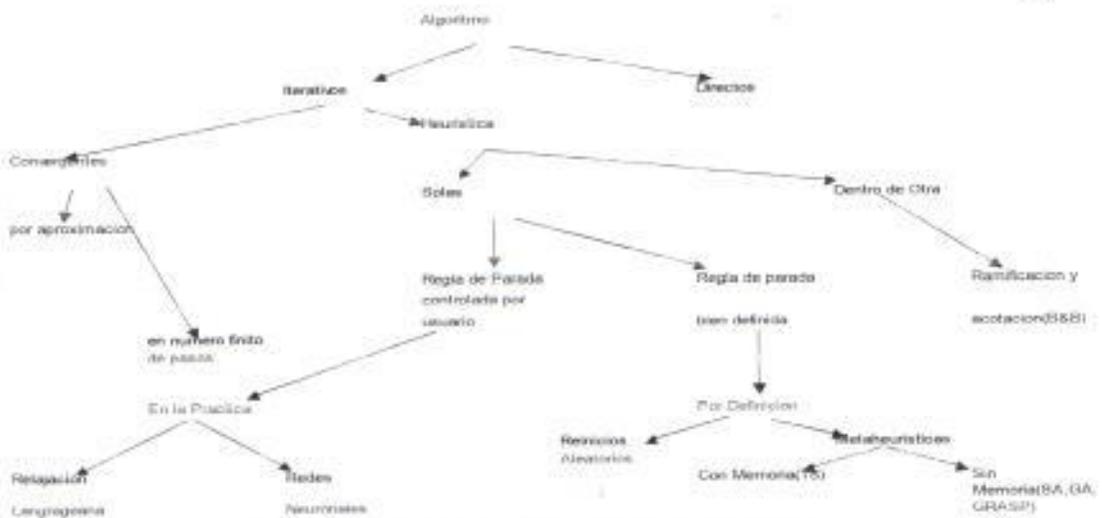


Fig. 2.1 *CLASIFICACIÓN DE ALGORITMOS

2.4 Factores de utilización.

En la actualidad, existe una gran variedad de estrategias para modelar la incertidumbre, como observamos entre los objetivos de esta tesis es aplicar en el diseño del modelo el uso de escenarios, para lo cual existen dos técnicas más usadas en modelos de incertidumbre las cuales son:

- *Optimización Estocástica*
- *Optimización Robusta.*

En la primera la idea principal es la habilidad de tomar acciones correctivas, después que eventos aleatorios ocurren, es decir nos ayuda en problemas de optimización que involucren incertidumbre en las restricciones del modelo a simular o en su función de salida, *"la falla más importante en metodología estocástica y de cualquier metodología que evalúe las decisiones utilizando solamente un escenario de datos, sea el esperado o el más probable está generalmente destinado a fallar"*. [43]

De lo anterior se concluye, que no se desea la solución óptima para un escenario en particular (ni siquiera para el escenario más probable), sino una decisión que se maneje bien a través de todos los escenarios potencialmente realizables (peor y mejor). A esto se le llama

Optimización robusta, como observamos la metodología estocástica no persigue este tipo de intereses. [52]

En esta tesis de optimización y simulación, aplicaremos la Optimización Robusta (una solución que mejor se tolere a través de todos los escenarios probables), adicionalmente en el manejo de algoritmos meta-heurísticos es necesario, que existan entornos en los escenarios a optimizar, entre las condiciones enunciaremos las siguientes:

- No se precisa de una solución óptima, sino eficiente.
- Existe limitaciones de tiempo, espacio y costo.
- No existe un método exacto de resolución
- La resolución requiere considerable tiempo de cálculo.
- Tenemos datos fiables.

El uso de modelos de simulación ha sido una técnica muy utilizada para aproximarnos a problemas muy complejos de tratar metódicamente. Sin embargo determinar los valores óptimos de los parámetros de una simulación puede pretender un tiempo de computación colosal, ya que el número de combinaciones posibles puede ser elevado. [52]

2.5 Convergencia: máximo y mínimos.

Los algoritmos aproximados se pueden clasificar en dos tipos principales: *algoritmos constructivos* y *algoritmos de búsqueda local*. Los primeros se basan en generar soluciones desde cero añadiendo componentes a cada solución paso a paso.

Los algoritmos de búsqueda local intentan repetidamente mejorar la solución actual con movimientos a soluciones vecinas (con la esperanza de que sean mejores). El caso más simple son los algoritmos de mejora iterativos: si en el vecindario de la solución actual s se

encuentra una solución mejor s' , ésta reemplaza la solución actual y se continúa la búsqueda a partir de s' ; si no se encuentra una solución mejor en el vecindario, el algoritmo termina en un óptimo local.

Uno de los principales inconvenientes en los algoritmos meta-heurístico es que estos puedan caer en óptimos locales y debido al comportamiento de la función a optimizar encallen en estos, por lo cual; el algoritmo no busca otras soluciones, esto daría un error en la obtención de la solución, dado que dicha solución no sería una aproximación de la **solución óptima**.

Como observamos en la gráfica el algoritmo encalla en un mínimo local (solución no óptima), actualmente la mayoría de los algoritmos meta-heurísticos, tienen implementado variantes, para comprobar que la solución no es un óptimo local. Uno de los métodos más usados, consiste en obtenida la solución óptima iniciar nuevamente el algoritmo, en otra vecindad totalmente diferente a la solución obtenida, si después de esto la “nueva” solución es igual a la solución inicial entonces nos encontramos con el **óptimo de la función**.

Como es lógico pensar la función a optimizar puede tener máximos o mínimos los cuales serán seleccionado de acuerdo a las condiciones de para del algoritmo, es decir si el algoritmo requiere maximizar (en nuestro caso minimizar el tiempo de producción) la condición de fin estará orientada a obtener el punto máximo de la función.



Fig. 2.2 ÓPTIMO LOCAL

CAPÍTULO III

META-HEURÍSTICA

III Meta-heurística

En general existe una gran cantidad de problemas que se consideran difíciles de resolver. La dificultad de los problemas está caracterizada por la complejidad computacional. La *meta-heurística* propone métodos, determinísticos o estocásticos para salir de mínimos locales y métodos de optimización robusta para la selección de una solución que mejor se comporte entre todos los posibles escenarios.

Algunas de las propiedades deseables para algoritmos meta-heurísticos son:

- ▶ **Eficiencia:** Tomar un tiempo razonable.
- ▶ **Amigable:** Fácil de entender y usar, con la menor cantidad de parámetros posibles.
- ▶ **Coherencia:** Se debe seguir en forma natural los principios de la meta-heurística.
- ▶ **Precisión:** Formulada en términos matemáticos precisos.
- ▶ **Simplicidad:** Basada en un principio claro que pueda ser aplicable en general.
- ▶ **Efectividad:** Encontrar soluciones óptimas de problemas donde se conoce su solución.
- ▶ **Robustez:** Ser consistente en una amplia variedad de instancias.
- ▶ **Innovación:** Poder atacar nuevos tipos de objetivos.

Los algoritmos meta-heurísticos cumplen con algunas de las propiedades arriba referidas, en general dependen de la naturaleza del problema y el propósito analizado. Es decir, los algoritmos meta-heurísticos están basados en métodos de construcción o en búsqueda local. Los basados en construcción trabajan con soluciones parciales que tratan de completar, mientras los de búsqueda local se mueven en el espacio de soluciones completas. Por lo general los algoritmos que van encadenando soluciones son los más eficientes, pero sus análisis no siempre las más eficaces.

3.1 Selección de algoritmo meta-heurístico.

Entre las distintas técnicas meta-heurísticas de resolución de problemas de optimización combinatorial, surge la necesidad de innovar "inteligencia" en los algoritmos de búsqueda local [Glover et al., 1985]. Tabú Search en contraste, con otros algoritmos establecidos en técnicas aleatorias de búsqueda de soluciones en entorno, se precisa porque este utiliza un modelo basado en el uso de *estructuras de memoria dinámicas*, la cual nos asiste en abandonar los óptimos locales en los que puede converger al desplazarse de una solución a otra, dentro de la vecindad.

Por lo anterior, se almacenan los últimos movimientos realizados; en estructuras denominadas "memorias", y que son utilizadas en llevar presente aquellos movimientos que hacen al caer de nuevo en soluciones ya analizadas. Estas estructuras servirían para impedir el posterior desarrollo de esas soluciones previamente analizadas.

3.2 Algoritmo Tabú Search.

Tabú Search pueden situarse en diversos trabajos publicados hace alrededor de 20 años. Pero, la metodología fue desarrollada posteriormente por Fred Glover (1989). Numerosas aplicaciones han aparecido, así como artículos, para difundir el conocimiento teórico del procedimiento (Glover y Laguna). Esta es una técnica o procedimiento meta-heurístico usado para resolver problemas de optimización combinatorial, utilizado para guiar cualquier procedimiento de búsqueda local para explorar el espacio de soluciones más allá de la simple optimización local.

Tiene sus antecedentes en métodos diseñados para cruzar fronteras de factibilidad local tratadas como barreras en métodos clásicos para permitir la exploración de regiones no consideradas en otro caso. En esencia es un meta-heurístico que puede ser manejado para

guiar cualquier procedimiento de búsqueda local en una "búsqueda agresiva del óptimo general". La metodología consiste en evitar que la búsqueda quede encallada en un óptimo local que no sea el óptimo general de la función simulada y optimizada. [52]

Por tal efecto, el algoritmo toma el concepto de memoria y lo implementa mediante estructuras (matrices) con el objetivo de dirigir la búsqueda teniendo en cuenta la historia de la solución, este concepto es tomado de la inteligencia artificial. Es decir, el procedimiento trata de extraer información de lo sucedido e interactúa con el resultado. Búsqueda Tabú comienza de la misma forma que cualquier metodología de búsqueda, procediendo a interactuar de una solución x a otra, en el entorno de la primera $N(x)$. Sin necesidad de considerar todo el entorno de la solución, De lo anterior se desprende, que existe cierto aprendizaje y que la búsqueda es dirigida "inteligente". Por lo cual, la búsqueda Tabú en base a la información, su definición es:

"Es mejor una mala decisión basada en información que una buena decisión al azar, ya que, en un sistema que emplea memoria, una mala elección basada en una estrategia proporcionará claves útiles para continuar la búsqueda. Una buena elección fruto del azar no proporcionará ninguna información para posteriores acciones."

Búsqueda Tabú define el entorno reducido $N'(x)$ en aquellas soluciones viables del entorno de x . Así, se considera que a partir de x , sólo las soluciones del nuevo entorno son factibles. Para definir el entorno de una solución, veremos que consiste en marcar como tabú las soluciones previamente visitadas con anterioridad. A esto llamamos memoria a corto plazo y está formada almacenando en una *lista tabú* T las soluciones analizadas últimamente. Por lo cual en una iteración determinada, el entorno reducido se obtendría como el entorno normal eliminando las soluciones marcadas como tabú. El principal objetivo de marcar las soluciones visitadas como tabú es el de evitar que se formen ciclos en el procedimiento

búsqueda. Por ello se considera que después de un cierto número de iteraciones la búsqueda local está en una región distinta y puede librarse del estado tabú a las soluciones anteriores. De esta manera se mejora el esfuerzo computacional en encontrar el entorno reducido (se reduce el tiempo) en cada iteración. Inicialmente la meta-heurística Tabú Search sugerían listas Tabú de tamaño pequeño, en la actualidad se considera que las listas pueden crecer o reducirse dinámicamente ajustándose de acuerdo a la estrategia que se esté implementando.

```

generar la solución inicial  $s$ 
mejorSol =  $s$ 
Mientras NO se cumpla la condición de parada, Hacer
  generar el conjunto  $V^*$  de soluciones  $s_i = s \oplus m_i$ 
    tal que  $o m_i \notin LT$  o bien  $s_i$  satisface el criterio de aspiración.
  elegir la mejor solución  $s' \in V^*$  (respecto de la función  $f$  a optimizar)
  actualizar la LT con  $m_i$ 
   $s = s'$ 
  Si  $f(s') < f(\text{mejorSol})$ , entonces  $\text{mejorSol} = s'$ 
  Cada "cierto tiempo", intensificar y/o diversificar
Fin Mientras

```

Fig. 3.1 ALGORITMO TABÚ SEARCH

En ciertos casos, la memoria a corto plazo de Tabú Search está basada en atributos en lugar de ser explícita; esto es, en lugar de almacenar las soluciones completas (como ocurre en procedimientos de búsqueda exhaustiva) se almacenan únicamente algunas características de éstas con la finalidad de reducir al mínimo el esfuerzo computacional de las operaciones que se realizan a menudo (requieren de la exploración de un gran número de soluciones en poco tiempo).

La *memoria mediante atributos* produce un efecto beneficioso en la búsqueda, ya que los atributo pueden identificar un conjunto de soluciones. Así, un atributo que fue marcado como tabú por pertenecer a una solución visitada anteriormente, puede impedir en la iteración actual, alcanzar una solución por contenerlo. Esto permite, a largo plazo, que se identifiquen y mantengan aquellos atributos que inducen una cierta estructura beneficiosa en las soluciones visitadas.

La memoria a largo plazo almacena las frecuencias u ocurrencias de atributos en las soluciones visitadas tratando de identificar o diferenciar regiones. La memoria a largo plazo tiene dos estrategias asociadas: Intensificar y Diversificar la búsqueda. La intensificación consiste en regresar a regiones ya exploradas para estudiarlas más a fondo, para ello se favorece la aparición de aquellos atributos asociados a buenas soluciones encontradas. La diversificación consiste en visitar nuevas áreas no exploradas del espacio de soluciones, con lo que se modifican las reglas de elección para incorporar a las soluciones atributos que no han sido usados frecuentemente.

Memoria	Atributos	Estrategias	Ámbito
Corto Plazo	Reciente	Intensificar	Local
Largo Plazo	Frecuente	Diversificar	Global

Fig. 3.2 *ESTRATEGIAS DEL ALGORITMO TABÚ SEARCH

Tabú Search está basado en la interacción entre la memoria a corto y la memoria a largo plazo. Ambas memorias llevan asociadas sus propias estrategias y atributos, y actúan en escenarios diferentes. En ocasiones se utilizan estrategias de listas de candidatos para restringir el número de soluciones examinadas en una iteración dada o para mantener un carácter agresivo en la búsqueda.

Para evitar “perdersé” de soluciones buenas, es frecuente que en ciertas circunstancias, se permita violar las restricciones impuestas por la lista tabú, es decir permitir realizar un movimiento tabú si la solución que se obtiene con este, es mejor que todas las encontradas hasta ese momento, lo anterior se define *nivel de aspiración*. Su implementación consiste en permitir alcanzar una solución siempre que mejore a la mejor almacenada, aunque esté marcada como tabú. De esta forma se introduce cierta flexibilidad en la búsqueda y se mantiene su carácter agresivo.

Con los elementos anteriormente descritos se puede diseñarse un algoritmo tabú Search para un problema de optimización. Sin embargo, existen más componentes, que serán descritos posteriormente, por lo general dichos elementos han sido ignorados en el desarrollo de aplicaciones debido que la introducción de estos, constituyen un reto.

3.3. Análisis del algoritmo.

“El éxito de esta metodología en diversos problemas se debe a sus estructuras de memoria y al uso de estrategias de intensificación y diversificación. Las estructuras de memoria evitan retornar a soluciones visitadas anteriormente, permiten guardar atributos de buenas soluciones contribuyendo a identificar regiones de interés y más generalmente guiar la exploración del espacio de solución. Las estrategias de intensificación y diversificación permiten avanzar a una solución vecina que es peor que la solución actual, pero que proporciona la posibilidad de penetrar a un espacio del conjunto de soluciones factibles que de otro modo no habría sido visitado y que podría contener una solución óptima global al problema”.[53]

Si se parte de una solución actual x_0 , que resulta ser óptimo local, y se procede a buscar la mejor solución en una vecindad de x_0 , se obtendrá una solución x_1 que pasará a ser la nueva solución, aunque tenga mayor costo. Al continuar con x_1 buscando en el espacio de soluciones dentro de su vecindad, puede que sea x_0 la mejor de todas las soluciones. Es lógico pensar que se regrese a la solución anterior, entrando así en un lazo que impide salir de esa región y explorar otras regiones a las ya analizadas.

Este algoritmo considera ciertos movimientos tabú, de forma que no puedan ser aceptados los óptimos locales durante un cierto tiempo o un número determinado de iteraciones. En el caso anterior, podría considerarse tabú el movimiento, solución x_1 a partir del entorno x_0 de tal forma que no se aceptaría nuevamente x_0 como nueva solución, sino otra para evitar caer en un ciclo.

A partir de aquí proviene la Lista Tabú. Cada vez que un movimiento es ejecutado, se introduce su inverso (FIFO) de forma que los elementos de la lista están castigados durante

cierto tiempo. Por tanto, no será aceptado siempre y cuando permanezca en la lista tabú, ésta estrategia constituye la estructura de memoria de la meta-heurística tabú search (lista tabú). Posteriormente, analizaremos el criterio mediante el cual un movimiento tabú pueda ser aceptado, *nivel de aspiración*, las técnicas mencionadas aquí son implementadas en el algoritmo exceptuando aquellas incluidas como mejoras

- ▶ **Entorno:** Saltar hacia otras soluciones, significaría que el algoritmo de búsqueda tabú selecciona éstas en un entorno de la solución actual. Lo natural sería la selección completa, evaluando cada una de las soluciones y quedándose con la mejor que no sea tabú. Para realizar una búsqueda completa, es deseable que el tamaño no sea elevado. Si se considera un entorno de tamaño grande, con objeto de reducir el tiempo de computación, se puede realizar la búsqueda en un subconjunto tomado aleatoriamente, o bien realizar la búsqueda hasta que se mejora el costo de la solución encontrada.

- ▶ **Solución inicial:** Un factor importante a tener en cuenta es la posible influencia que tenga comenzar la búsqueda tabú con una solución inicial. Esta solución dependerá del algoritmo específico que la genera. Con una solución inicial buena, de bajo costo, se puede pensar que es posible evolucionar, a corto plazo, hacia soluciones mejores, pero podría suponer un gran perjuicio computacional si realmente evoluciona la búsqueda hacia regiones de soluciones menos favorables, en cualquier caso, siempre será posible generar una solución aleatoria.

- ▶ **Memoria a corto plazo:** Los movimientos rechazados del algoritmo de búsqueda tabú en su forma más simple se denomina de **memoria a corto plazo**, debido a que la búsqueda que se realiza es local, utilizando esta memoria en los movimientos prohibidos de la lista tabú.

- ▶ **Memoria a medio plazo:** Tiene como objetivo registrar los atributos más comunes de un subconjunto de soluciones generadas durante un período de la búsqueda que con más probabilidad lleven hacia mejores zonas para explorar. Estos atributos sirven para intensificar la búsqueda de mejores soluciones.
- ▶ **Memoria a largo plazo:** Renueva la búsqueda sobre regiones poco exploradas. Una forma de diversificación consiste en reiniciar periódicamente la búsqueda desde puntos elegidos aleatoriamente. El método consiste en registrar los atributos de los movimientos más utilizados en anteriores iteraciones, penalizándolos a través de una lista tabú a largo plazo.
- ▶ **Tamaño de lista tabú:** Actualmente, se toman valores dependientes del tamaño del problema. En cualquier caso, constituye un parámetro importante cuya influencia habría que analizar y del cual dependerá la evolución del algoritmo.
- ▶ **Atributos de lista tabú:** Recopilar la descripción completa de las soluciones exploradas y comprobar si cada movimiento se encuentra en la lista tabú puede ocupar mucho tiempo. Como alternativa se puede almacenar un atributo representativo del movimiento, los que se consideren, así como la forma de almacenarlos dependerá la resolución del problema a resolver.
- ▶ **Aspersión:** Si todos los movimientos de la lista tabú se prohíben, se evita entrar en ciclos, pero se puede perder movimientos que acerque a mejores soluciones. El nivel de aspiración refiere a un criterio de aceptación de las soluciones que están incluidas en la lista tabú. Por lo general se define una función de aspiración, de forma que un movimiento tabú podría ser elegido si la nueva solución tiene un costo menor que la

solución actual. También se puede considerar como tal nivel el coste de la mejor solución alcanzada hasta el momento. Puede suceder que un movimiento tabú diese lugar a una solución cuyo costo fuese menor que dicho nivel. En tal caso, parece no tener sentido rechazarla.

- ▶ **Criterio de Parada:** Se puede establecer un número máximo de iteraciones, o un número máximo de pasos sin mejorar la diferencia del costo.

Existen otros elementos de Tabú Search que han dado buenos resultados en algunos problemas de optimización los cuales son:

- ▶ **Oscilación:** La oscilación estratégica opera orientando los movimientos en relación a una cierta frontera en donde el método se detendría normalmente. Sin embargo, en vez de detenerse, las reglas para la elección de los movimientos se modifican para permitir que la región al otro lado de la frontera sea alcanzada. Posteriormente se fuerza al procedimiento a regresar a la zona inicial. El proceso de aproximarse, traspasar y volver sobre una determinada frontera crea un patrón de oscilación que da nombre a esta técnica. Una implementación sencilla consiste en considerar la barrera de la factibilidad / infactibilidad de un problema dado. Implementaciones más complejas pueden crearse identificando determinadas estructuras de soluciones que no son visitadas por el algoritmo y considerando procesos de construcción / destrucción asociados a éstas. La oscilación estratégica proporciona un medio adicional para lograr una interacción muy efectiva entre intensificación y diversificación.
- ▶ **Influencia:** Son aquellos movimientos que producen un cambio importante en la estructura de las soluciones. Usualmente, en un procedimiento de búsqueda local, la

búsqueda es dirigida mediante la evaluación de la función objetivo. Sin embargo, puede ser muy útil el encontrar o diseñar otros evaluadores que guíen a ésta en determinadas ocasiones. Los movimientos de influencia proporcionan una evaluación alternativa de la bondad de los movimientos al margen de la función objetivo. Su utilidad principal es la determinación de estructuras subyacentes en las soluciones. Esto permite que sean la base para procesos de Intensificación y Diversificación a largo plazo.

- ▶ **Elecciones Probabilísticas:** Normalmente TS se basa en reglas sistemáticas en lugar de decisiones al azar. Sin embargo, en ocasiones se recomienda el aleatorizar algunos procesos para facilitar la elección de buenos candidatos o cuando no está clara la estrategia a seguir (quizá por tener criterios de selección enfrentados). La selección aleatoria puede ser uniforme o seguir una distribución de probabilidad construida empíricamente a partir de la evaluación asociada a cada movimiento.
- ▶ **Umbrales Tabú:** El procedimiento conocido como Tabú Thresholding se propone para aunar ideas que provienen de la Oscilación Estratégica y de las Estrategias de Listas de Candidatos en un marco que facilite su implementación. El uso de la memoria es implícito en el sentido que no hay una lista tabú en donde anotar el status de los movimientos, pero la estrategia de elección de los mismos previene el ciclado. Esta técnica utiliza elecciones probabilísticas y umbrales en las listas de candidatos para implementar los principios.

Tabú con Umbral

El algoritmo Tabú con Umbral [Glover, 1991] surge como una versión reducida del algoritmo general de búsqueda tabú, en el que se prescinde del uso de lista tabú. Este introduce criterios probabilísticas para la elección de movimientos sin hacer referencia

explícita a ninguna de las memorias. De aquí se evita la exploración repetida de soluciones, por lo que, el algoritmo tabú con umbral pretende conseguir un procedimiento de búsqueda que permita una exploración de soluciones de manera que se evite caer en aquéllas que ya han sido analizadas con anterioridad.

El algoritmo tabú con umbral consiste en la ejecución de dos fases, intensificación y diversificación. La primera consiste en una exploración donde sólo se aceptan movimientos que mejoran el valor de la función a obtener o resultado, de manera que al final se obtiene un óptimo local, en la fase de mezcla o dispersión se aceptan movimientos tanto que mejoran como que empeoran el valor de la función resultado para, de esta forma escapar del óptimo local y diversificar la búsqueda hacia otras vecindades.

Una vez definido el conjunto de movimientos posibles, de forma que abarca todo el espacio de soluciones como una combinación de éstos, el algoritmo propone dividir en subconjuntos de movimientos que pueden ser utilizados tanto en la primera como en la segunda etapa. En cada una, el subconjunto de movimientos a realizar se selecciona de manera probabilística, sin embargo no es el mismo procedimiento de selección y asignación de probabilidades el que se utiliza en ambas etapas. Tabú con umbral sustituye la lista tabú por estrategias que combinan la aleatoriedad en la selección del movimiento con la que se usa para examinar los posibles movimientos a efectuar.

Procedimiento para umbral tabú:

- 1.- En la etapa de intensificación se explora el conjunto de movimientos según la estrategia de selección aleatoria:

“Se divide el conjunto en bloques. En cada iteración de esta fase se realizan movimientos aleatorios dentro de cada bloque. Se elige el mejor movimiento si mejora la función objetivo. La

división del conjunto en bloques y la elección aleatoria de los elementos en dichos bloques emulan el comportamiento de una lista tabú."

2.- En la siguiente se explora el conjunto M mediante el procedimiento de selección aleatoria completa:

"Se toma un subconjunto de forma aleatoria. Se selecciona el mejor movimiento de dicho subconjunto y se acepta, tanto si mejora como si empeora la función objetivo. Este subconjunto no vuelve a ser examinado en esta fase."

Tanto en una etapa como en otra, cada vez que un elemento es seleccionado, se busca el movimiento que conduzca a la solución óptima. La búsqueda se realiza mediante el uso de listas de candidatos, que constituyen un procedimiento de búsqueda que permite reducir el número de exploraciones.

- ▶ **Encadenamiento:** Este método trata de volver a unir dos buenas soluciones mediante un nuevo camino. Así, si en el proceso de búsqueda hemos encontrado dos soluciones x_1 y x_2 con un buen valor de la función objetivo, podemos considerar el tomar x_1 como solución inicial y x_2 como solución final e iniciar un nuevo camino desde x_1 hasta x_2 . Para seleccionar los movimientos no consideraremos la función objetivo o el criterio que hayamos estado utilizando sino que iremos incorporando a x_1 los atributos de x_2 hasta llegar a ésta última. Por eso esperamos que alguna de las soluciones intermedias que se visitan en este proceso de Entorno Constructivo sea mejor. Actualmente se considera el explorar el entorno de las soluciones intermedias para dar más posibilidad al descubrimiento de buenas soluciones.

3.4 Aplicación del algoritmo

La simulación de procesos es una de las más grandes herramientas aplicadas actualmente en la administración de áreas productivas, la cual se utiliza para representar un

proceso mediante otro mucho más simple y entendible. La necesidad de su aplicación depende del grado de complejidad de los procesos dados por la misma organización, puesto que, en el caso del modelamiento de un proceso medianamente sencillo, la simulación podría generar pérdidas de tiempo y utilización de capacidades ociosas frente a eventos de mayor importancia en las empresas.

Con el desarrollo de las diferentes modalidades de industria que se han venido dando en los últimos tiempos, se hace implícita la relación entre manufactura y producción eficiente con aplicación efectiva del conocimiento. Desde que se ha tenido una operatividad lógica sobre los procesos implementados en determinada empresa, se ha hablado de la era industrial, en la cual el factor estratégico de la ventaja competitiva lo constituía la tierra; en este caso, el sector agrícola representaba la utilidad industrial del momento. Siguió luego la primera revolución industrial, la cual se delimitó por la maquinaria y la utilización minoritaria de mano de obra. Este factor de delimitación es el mismo factor estratégico de ventaja competitiva frente al resto de empresas. A diferencia de la segunda revolución industrial, la estrategia se enfocó en la utilización de tecnología como generadora de valor agregado. Por último, se llega a la época en la que el *"conocimiento define el funcionamiento de las empresas según el nivel de aplicación y el capital intelectual"* la cual garantiza la permanencia de las mismas dentro del mercado escogido.

La empresa como factor único frente al entorno y en los procesos implementados por ésta frente a dicho entorno y al resto del mundo. Es aquí en donde se rescata el concepto de tendencia (simulación y optimización combinatorial), puesto que es lo que la empresa debe aplicar según el interés que tenga, el momento que esté viviendo y los recursos con los que cuente para realizar los objetivos trazados desde un principio. De acuerdo con el desarrollo de

nuevos procedimientos en la gestión organizacional, las empresas que han decidido invertir en mejores recursos y optimizar los ya implementados mediante la administración adecuada de factores internos en búsqueda de un elevado desempeño en el mercado. Ante desafío de organizar la producción y aplicar los conocimientos en entornos cada vez más industrializados, la *gestión tecnológica* surge como una de las capacidades con las que puede contar la organización actual para administrar la tecnología que se implemente dentro de las diferentes áreas.*

Uno de los factores importantes para analizar este tipo de meta-heurística (búsqueda tabú) es su implementación en problemas de permutación, los cuales son una clase importante de problemas en optimización, y adicionalmente ofrecen un modo útil para demostrar algunas de las consideraciones que deben ser tratadas en la forma combinatoria. Como base para el análisis, consideremos la combinación en la elaboración de una serie de ítems fabricado por un número determinado de equipos. El orden según el cual se planifican estos ítems determina el valor del rendimiento total en la elaboración resultante (costos en términos de mínimo tiempo de elaboración).

El problema consiste en encontrar el orden de los ítems que maximiza el valor de rendimiento total de elaboración. Supongamos que se consideran 7 ítems para una serie en particular, y que evaluar el costo de elaboración total de una producción en particular es un procedimiento costoso. Deseamos un método capaz de encontrar una solución óptima o cercana a la óptima explorando sólo un pequeño subconjunto de todas las combinaciones posibles.

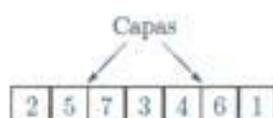


Fig. 3.3 COMBINACIÓN INICIAL

Nos centramos en el problema de evaluar el costo de elaboración total de una producción (en función de su tiempo de producción), para introducir e ilustrar los mecanismos básicos de la búsqueda tabú. Primero construimos una solución inicial de este problema. Asumamos que la solución inicial de nuestro problema es igual a la figura 1.

La ordenación en la Fig. 1 especifica que el ítem 2 se sitúa en la primera posición, seguida por el ítem 5, etc. La producción resultante tiene un valor de costo de producción (tiempo de elaboración de 10 unidades), así; los métodos búsqueda tabú operan bajo el supuesto de que se puede construir un entorno para identificar "soluciones adyacentes" que puedan ser alcanzadas desde la solución actual. Los intercambios por pares son usados para definir entornos en problemas de permutaciones, identificando movimientos que conducen una solución a la siguiente. En nuestro problema, un intercambio cambia la posición de dos ítems como se ilustra en la figura 2.



Fig. 3.4 INTERCAMBIO DE ÍTEMES 5 Y 6

Por tanto, el entorno completo de una solución está constituido por 21 soluciones adyacentes que pueden ser obtenidas a partir de estas permutaciones. Como observamos los métodos de búsqueda tabú (*Path Relinking*) [44], [54] emplean procedimientos sistemáticos. Mientras los algoritmos genéticos y meméticos [31], [61], [65] [60] emplean principalmente procedimientos aleatorios

"El tipo de meta-heurística más importante es el de las meta-heurísticas de búsqueda, que establecen estrategias para recorrer el espacio de soluciones del problema transformando de forma iterativa soluciones de partida."

3.5 Herramientas: Simulación - Optimización

A continuación detallamos una lista de los programas de simulación más utilizados en la actualidad y las áreas para las cuales fueron diseñados, cabe indicar que aún no se ha generalizado el uso de herramientas, que integren la simulación de sistemas a través de procesos estocásticos, con la optimización de los mismos a través de métodos meta-heurísticos, como el que actualmente se ha propuesto en esta tesis.*

PRODUCTO Distribuidor Fabricante	Áreas de aplicación	Aplicaciones realizadas	Lenguajes
GI - 7 Sincromat NDCSuecia	Simulación para el análisis de líneas de montaje flexi-bles, almacenes automáticos y sistemas de transporte	Líneas de montaje de los sectores automóvil, línea blanca. Grandes almacenes , logística y transporte.	-
HYSYS Rexroth Mannesman Rexroth (Alemania)	Simulación de sistemas hidráulicos. Obtención de gráficos de velocidad , posición, presión, etc.	Sistema hidráulico completo de banco de prueba de impactos de automóviles.	-
MODSIM III SICO Software CACI (USA)	Propósito general	Creación de librerías de objetos para otros entornos como el Simfactory.	Propio
SIMAC Gomend, S.A. PROSYST(Francia)	Simulación de sistemas reales para poner a punto automáticas.	Simulación de máquinas de conformado (automóvil). Simulación de máquinas transfer.	Propio
SIMANIV Sincromat System Models Corp. (USA)	Simulación para el análisis de líneas de montaje flexibles, almacenes automáticos y sistemas de transporte-AGV	Líneas de montaje de los sectores automóvil, línea blanca. Grandes almacenes , logística y transporte.	Propio
SIMFACTORY SICO Software CACI (USA)	Simulación para análisis de sistemas de producción.	Simulación de sistemas de producción y fabricación.	Modsim
SIMVOX Fisher- Rosemount, S. A. MUNGER(USA)	Simulación de controladores e instrumentación para sistemas de fabricación PROVOX.	Monitorizar simuladamente los inputs y outputs de sistemas basados en controladores industriales PROVOX	Propio, Fortran,C
TAYLOR II Aries Ing y Sistemas F&H Simulations BV (Holanda)	Propósito general	Simulación de grandes almacenes, líneas de montaje sector automoción, líneas de pistones.	Propio, Pascal
WITNESS AT&T ISTEEL (USA)	Simulación de sistemas de fabricación y logística.	Simulación de líneas de producción en el sector automoción, aplicaciones para grandes superficies	Propio
FACTOR/AIM PRITSKER Corp. (USA)	Simulación de sistemas de fabricación y logística.	Simulación de sistemas de producción y logística	C++

CAPÍTULO IV

APLICACIÓN: ANÁLISIS Y DISEÑO

IV APLICACIÓN: ANÁLISIS Y DISEÑO

El siguiente análisis nace de la necesidad de obtener un sistema que optimice la producción de ítems involucrados en una cadena de procesos, como sabemos no es fácil determinar el comportamiento de un determinado equipo aunque si bien es cierto los mismos funcionan bajo estándares de producción, estos estándares luego de cierto lapso de tiempo y deterioros propio del uso de los equipos se degradan, por lo cual es complicado controlar el tiempo de funcionamiento de los equipos, y si ha esto añadimos que ellos producen ítems a determinada razón en el tiempo. El compromiso de optimizar los tiempo de los recursos disponible para la producción, y de esta forma aumentar el número de ítems producidos, se complica en gran medida, y si a esto añadimos el hecho de verificar cuan eficiente resulta un cambio en los procesos productivos sin afectar el proceso productivo o peor aún sin llegar a generar pérdidas económicas, resulta altamente riesgoso:

- a.- Comprobar en sistemas en funcionamientos , sin un previo análisis.
- b.- Pérdidas económicas y tiempo de producción serian elevadas.
- c.- El tiempo en obtener una solución óptima y analizar este tipo de procesos a partir de sistemas reales (sin considerar las pérdidas económicas) nos tomaría un par de vidas llevar a cabo tal objetivo, como lo comprobamos en secciones posteriores.

Por lo anterior, y en la actualidad la eficiencia de una empresa no solo se mide en función de la tecnología que posee, si no también y de mucho más importancia, es el conocimiento de la tecnología y su aplicación en los diferentes procesos que a esta le concierne, siendo la optimización combinatorial (meta-heurística: Tabú Search) y la simulación de procesos (procedimientos estocásticos) herramientas de gran utilidad en el estudios de problemas de producción que involucran criterios de incertidumbre y optimización de la producción a partir de diversos escenarios o lapsos de tiempo.

4.1 Modelo de Análisis

Sea el sistema con las siguientes condiciones iniciales, obtener una secuencia de producción óptima, la cual se maneje bien a lo largo del tiempo y adicionalmente se proteja contra el peor de los escenarios, es decir hallar una solución, la cual no debe ser óptima para un escenario en particular, ni siquiera para el más probable, sino aquella que mejor se comporte a lo largo del tiempo y pueda considerarse como independiente del comportamiento de las variables de entrada (funciones de distribución)

Modelos Estáticos

Modelo de Análisis del Sistema

Sea el sistema, con las siguientes condiciones iniciales:

- Ítems producidos al final del proceso: 6
- Máquinas requeridas para todo proceso: 7

Cada proceso de producción tiene su función de distribución (tiempo en producir un ítem por un equipo en particular) respectiva, la cual detallamos más adelante. Adicionalmente detallamos los parámetros necesarios para el análisis de la simulación y la forma del ingreso de los parámetros al sistema

Parámetros de la Optimización y Simulación

Números de escenarios:	20,000	Simulación del Sistema
Número de iteraciones:	7	Optimización Combinatorial

PARÁMETROS DE SIMULACIÓN	
CANT. DE PRODUCTO:	6
CANT. DE MÁQUINAS	7
NUM. DE ESCENARIOS:	20,000
NUM. DE ITERACIONES:	7

CONDICIONES INICIALES: FUNCIONES DE DISTRIBUCION DE LAS VARIABLES DE ENTRADA

TIEMPO DE PROCESAMIENTO EN LA PRODUCCIÓN DE UN ÍTEM DE ACUERDO POR LOS EQUIPOS REQUERIDOS

EQUIPO ÍTEM	1	2	3	4	5	6	7
1	*N(40.5; 30.8)	-	E(89.5)	-	D(34.8; 100.3; 2.9)	-	T(33.45; 56.4; 88.9)
2	-	*N(78.8; 56.9)	D(40.88; 60.57; 12.89)	E(45.9)	-	-	U(24.56; 48.9)
3	D(89.23; 10.42; 3.88)	E(77.1)	*N(78.2; 10.2)	U(22.8; 25.6)	T(30.8; 31.3; 99.8)	**N(39.7; 1.3)	E(40.12)
4	U(70.3; 100.5)	E(23.6)	-	-	*N(99.3; 78.4)	E(67.8)	D(67.23; 100.34)
5	-	U(31.8; 56.1)	T(34.9; 35.4; 90.3)	E(87.6)	E(19.7)	D(20.56; 44.22; 15.89; 4.67)	-
6	*N(71.4; 33.9)	D(100.7; 345.2; 88.23)	**N(80.3; 5.7)	-	U(43.7; 67.9)	T(26.5; 45.1; 55.5)	T(20.1; 29.3; 31.4)

TIEMPOS DE FUNCIONAMIENTO Y REPARACIÓN

EQUIPO	TIEMPOS	
	FUNCIONAMIENTO	REPARACIÓN
1	T(23.6; 67.8; 89.3)	D(34.8; 100.4; 190.5)
2	*N(45.7; 36.8)	U(78.9; 99.23)
3	E(12.7)	U(23.8; 26.7)
4	**N(25.4; 5.02)	U(56.7; 70.8)
5	E(70.8)	*N(20.3; 15.8)
6	E(49.38)	**N(45.8; 4.6)
7	D(102.3; 46.3; 200.5)	T(12.4; 24.5; 66.91)

ESTRUCTURA DE LAS FUNCIONES DE DISTRIBUCIÓN DE ACUERDO AL INGRESO EN EL SISTEMA

NOMENCLATURA	FUNCIÓN	PARÁMETROS		
E(a; b)	Uniforme	a: Límite inferior	b: Límite superior	
E(a)	Exponencial	a: Factor Beta		
*N(max; min)	Normal	max: Límite superior	min: Límite inferior	
**N(mean; dev)	Normal	mean	dev: desviación	
T(a; b; c)	Triangular	a: Límite inferior	b: modo	c: Límite superior
D(a; b; c; d)	Trapezoidal	a: Probabilidad relativa	b: Probabilidad relativa	c: Probabilidad relativa

Sea el sistema con las condiciones propuestas del cuadro anterior, el objetivo es encontrar una secuencia óptima, la cual se comporte bien en cualquier instante de tiempo a lo largo de la cadena de producción, y se proteja contra el peor de los escenarios. Para lo cual el parámetro a optimizar será el tiempo requerido en todo el proceso de producción, es decir la combinación máquinas - ítems cuyo costo de producción (función de penalización) será aquella que corresponda al menor tiempo de utilización requerido por los recursos disponibles en el proceso de producción.

Como hemos confirmado en las condiciones iniciales, nuestro proceso será implementar un sistema que representa las estructuras (funciones de distribución) requeridas por los recursos para definir el proceso de producción de determinados ítems, es decir modelar el sistema a través de procesos de simulación (procesos estocásticos) en los cuales podamos analizar el sistema a lo largo del tiempo, es decir definir un entorno o escenario para estudiar su comportamiento. Para definir el escenario, es necesario primero construir las funciones matemáticas que representen nuestros procesos de producción, es decir las funciones de distribución probabilística que definan la forma de la función de probabilidad, obtenida a partir de los datos históricos recolectados en los diferentes procesos. Finalmente debemos certificar que nuestra función de distribución, ahora modelo matemático, represente un sistema lo más cercano a la realidad es decir cuyas probabilidades (variables de entrada) sean:

- a.- Independiente entre sí (correlación igual a cero, condición necesaria dado que las distribuciones son generadas a partir de funciones normalmente distribuidas)
- b.- Sean generadas en forma aleatoria.
- c.- Correspondan a la función de distribución de la variable de entrada.

De acuerdo a lo anterior, debemos generar semillas (procedimiento para concebir números aleatorios) que nos garanticen las condiciones anteriormente descritas, para dicho efecto aplicaremos el método congruencial para la generación de los números pseudos-aleatorios, por lo cual el escenario será entonces la definición de un evento en el tiempo, es decir el tiempo requerido por cada variable de entrada para cada proceso y a partir del cual iniciaremos nuestro análisis de optimización, por ejemplo sean las condiciones iniciales definidas anteriormente:

♣ Ítems producidos al final del proceso: 6

♣ Máquinas requeridas para todo proceso: 7

La máquina 1, demora 4 horas en producir los ítems 1, 3 5 y 7

La máquina 2, demora 7 horas en producir los ítems 2, 3, 4 y 7 y así sucesivamente.

Con lo cual, estaría modelado el evento de producir de todos los ítems, es obvio pensar que en nuestro caso el tiempo de demora es generado a través de las semillas (método congruencial) e introducido a la función matemática que representa la función de distribución de la máquina 1 - ítems 1, 3, 5 y 7 (normal, exponencial, discreta y triangular respectivamente) a esto llamaremos escenario n, de aquí en adelante entiéndase por escenario a cualquier periodo de tiempo requerido por todas las variable aleatoria V.A. de entrada (función de distribución) para la producción total de los ítems requeridos.

Definida la metodología para obtener un escenario en cualquier periodo, a continuación procederemos a definir la técnica de análisis de ese escenario, es decir el proceso para obtener la solución óptima requerida para minimizar el tiempo empleado por los recursos en la producción total de la secuencia de ítems. Como hemos analizado, en los capítulos anteriores una de las técnicas alternativas para el exploración de este tipo de problema son los algoritmos meta-heuristicos, entre los cuales existen diferentes puntos de

vista, la selección del algoritmo Tabú Search, que se ha estudiado anteriormente, (recordando, los algoritmos de búsqueda son procedimiento sistemático bien difundidos en sistemas que utilizan inteligencia artificial, mientras que algoritmo genéticos, meméticos y otros utilizan procedimientos aleatorios, los cuales modificando la estructura de su solución en base a una vecindad, lo cual desde el punto de vista de muchos autores y en el mío en particular, puede caer en óptimos locales, adicionalmente su tiempo de procesamiento no es regular esto debido a los procedimientos aleatorios, lo cual podría generar cálculo complicados) se debe que es una de las técnicas de mayor análisis y desarrollo, aplicado en diversos procesos. En definitiva de acuerdo a su estructura (memorias) su aplicación es ideal para obtener soluciones óptima, en forma rápida o en el peor de los casos soluciones que podrían mejorarse debido que existe registros de la evolución de sus soluciones.

Parámetros de la Optimización y Simulación

Números de escenarios:	20,000	Simulación del Sistema
Número de iteraciones:	7	Optimización Combinatorial

De acuerdo a lo anterior, serán 20,000 eventos con su tiempo total de producción requerido por todas las variables de entradas, y en cada evento o escenario se procederá a obtener una solución óptima, es decir una secuencia de producción máquina ítems, que satisfaga las condiciones iniciales y adicionalmente cuyo costo de producción (función de penalización del algoritmo meta-heurístico: Tabú Search) sea el tiempo mínimo requerido para la producción total de los ítems solicitados en la cadena de producción. Prestemos atención al siguiente razonamiento ya que es la base de nuestro sistema de optimización robusta y del cual procederemos a realizar la comparación de las soluciones. Dado que existe una solución óptima para un escenario n , si obtenemos todas las soluciones "óptimas" de cada escenario, es decir 20,000 soluciones "óptimas" y de estas obtenemos su moda

generacional, (aquella solución que más veces se repite) obtendremos entonces una solución que mejor se comporta no para el escenario más probable, sino aquella que es más eficiente para todos los escenarios en conjunto (independencia de las funciones de distribución: variables de entrada) y por lo tanto aquella que se protege contra el peor de los escenarios.

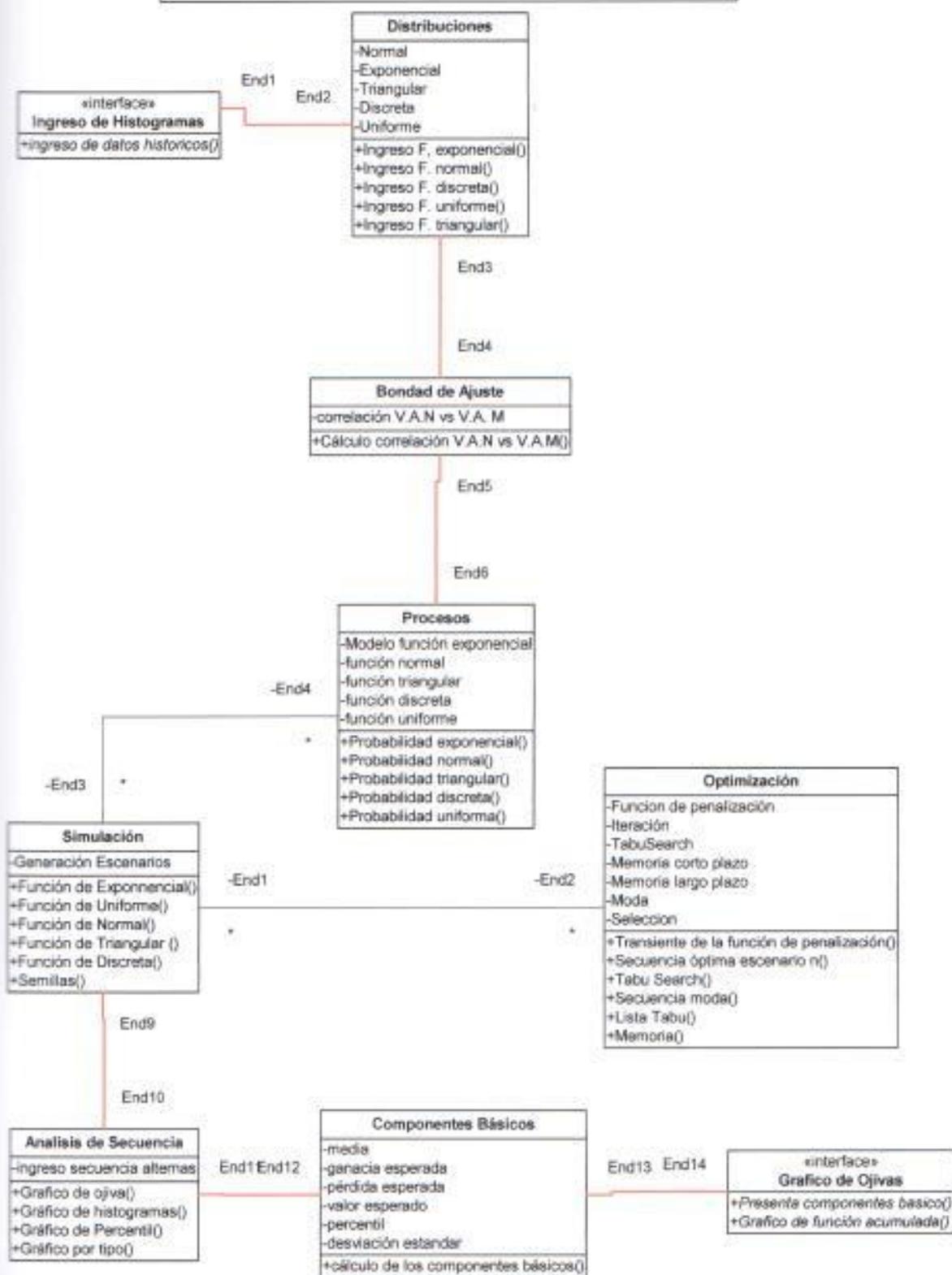
El análisis del algoritmo meta-heurístico Tabú Search, se complementa con la análisis de la moda generacional de la soluciones óptimas en los n escenarios, esto con la finalidad de obtener la solución más eficientes a lo largo de todos los eventos. Lo anterior podemos ratificar en el último paso del sistema; comprobar la solución óptima a través del análisis estadísticos, como último paso procederemos a simular nuevamente el sistema con las mismas condiciones iniciales, y almacenaremos los resultados obtenidos de la simulación de cada secuencia, es decir de la secuencia óptima y otras secuencias seleccionadas de manera al azar o secuencias donde el analista de procesos o tomador de decisiones del sistema tenga dudas. A través de estos resultados, calcularemos los componentes básicos y los gráficos de las funciones acumuladas u ojivas de todas las secuencias (incluida la secuencia óptima) y con los cuales evidenciaremos los resultados previamente esperados. Entre las características que deberían presentar los componentes básicos y el gráfico de la función acumulada (ojiva) de la secuencia óptima son:

- ◆ Media deberá ser menor que cualquier otra secuencia analizada (mínimo tiempo empleado en la producción de los cadena de ítems), es decir se debe concentrar la mayoría de los eventos en la parte izquierda del histograma.
- ◆ La intersección de las ojivas (porcentaje de riesgo en implementar el proyecto) para la secuencia óptima deberá representar una mayor ganancia en su implementación, es decir su intersección deberá ser en la parte superior de los gráficos (sobre el 50%) con relación a cualquier otra secuencia.

Con este último paso queda por terminado el proceso de optimización y simulación del proceso productivo antes propuesto, la aplicación combinada de estas metodologías es fuente de un amplio estudio por diferentes investigadores y su aplicación es cada vez más utilizada en proceso de programación no lineal, donde es necesario obtener soluciones eficientes en un corto periodo de tiempo, adicionalmente se puede inferir que el tiempo empleado en obtener la solución, es manejable dado que podemos afinar el número de escenarios requeridos para la simulación, de esta forma reducimos notablemente el tiempo de procesamiento de todo el sistema, claro esta sacrificando la eficiencia de la solución, esto es importante acotar cuando se trata de sistemas con una gran cantidad de procesos ejecutándose simultáneamente, entonces el analista y el tomador de decisión deberán medir entre obtener una solución lo más eficiente posible y el tiempo que se demoraría en obtener dicha solución. Esto en el caso, que se requiera que el sistema arroje en forma inmediata una solución, pero se deja constancia que dicho efecto degradaría el rendimiento de la solución en cuanto al tiempo de procesamiento empleado por todos los recursos para obtener la cadena de producción.

MODELO DE ANÁLISIS DE OBJETO

Modelo de Análisis de Objetos



Descripción de Clases

Nombre: Distribuciones

Descripción: El analista previo a la entrega de los datos históricos deberá obtener la forma de las funciones de distribución que intervienen en el proceso (variables de entrada), aquí se ingresará las funciones que intervienen en todo el proceso.

Atributos: Parámetros de cada distribución, condiciones iniciales de la optimización

Operaciones: Ingreso de funciones de distribución de los distintos procesos relacionados con la simulación

Nombre: Simulación

Descripción: Generación de n escenarios de acuerdo a la selección del usuario y la forma de la función matemática de cada distribución

Atributos: Selección de las semillas, normalmente independiente.

Operaciones: Generación de escenarios a través de las semillas previamente encontradas

Nombre: Optimización

Descripción: Análisis de los n escenarios, para encontrar la solución óptima en base del algoritmo meta-heurístico: Tabú Search

Atributos: Función de penalización y el número máximo de iteraciones definido por el usuario.

Operaciones: Búsqueda de la solución óptima, a través del algoritmo meta-heurístico Tabú Search.

Nombre: Procesos

Descripción: Definición de los modelos matemáticos que describen el comportamiento de las variables de entrada (funciones de distribución de las variables de entrada).

Atributos: Modelos matemáticos de las diversas distribuciones de probabilidad.

Operaciones: Generación de la probabilidad de acuerdo al comportamiento de la variable de entrada

Nombre: Análisis de Secuencia

Descripción: Comparación de las diferentes secuencia escogidas al azar versus la secuencia óptima encontrada

Atributos: Selección de secuencia alternas para su comparación

Operaciones: Comparación de las secuencias versus la secuencia óptima encontrada TS, gráfico de ojiva

Nombre: Componentes básicos

Descripción: A través de los gráficos de las funciones acumuladas podemos comparar el índice de riesgo en la implementación de la solución, en esta adicionalmente podemos verificar sus componentes básicos y realizar la comprobación en forma analítica además de los gráficos.

Atributos: Media, ganancia esperada, valor esperado, pérdida esperada, percentil, desviación estándar

Operaciones: Cálculo de los componentes básico estadísticos

Nombre: Bondad de ajuste

Descripción: Para confirmar que las semillas almacenadas, las cuales posteriormente serán las encargadas de generen los números pseudo-aleatorios para las funciones de distribución, estas sean independientes (condición indispensable en el algoritmo variable de entrada independientes entre si), se realiza el calculo de la correlación el mismo que deberá ser próximo a cero en la parte positiva o negativa.

Atributos: Correlación de las diferentes variables de entrada.

Operaciones: Cálculo de la correlación entre las variables de entrada.

Modelos Dinámicos

Descripción de Actores

Actores Primarios

- ⚡ Analista de procesos
- ⚡ Jefe de planta

Nombre: Analista de proceso

Descripción: Es la persona responsable de realizar las pruebas en el sistema antes de implementar la solución óptima en el sistema real

Notas:

El Analista, generar los diversos escenario (ingreso de parámetros de las funciones de distribución) y verifica la solución obtenida en la optimización.

La verificación lo realiza a través de la simulación de lo procesos, aplicando la secuencia óptima hallada por el algoritmo meta-heurístico: Tabú Search

Nombre: Jefe de Planta

Descripción: Es la persona responsable primero de entregar los datos históricos de las funciones de distribución de cada variable de entrada, es decir entrega la información relacionada con respecto al funcionamiento general de todos los procesos. Dicha información posteriormente es procesada por el analista de proceso

Segundo y lo mas importante, verifica el comportamiento del sistema real en base a la secuencia óptima provista por el sistema, si el sistema no mejora entonces se ejecuta nuevamente el algoritmo afinando los parámetros de entrada (funciones de distribución)

Actores Secundarios

👤 Gerente de la empresa

Nombre: Gerente de la empresa

Descripción: Es el encargado de analizar los índices de riesgo proporcionados por los gráficos de las funciones acumuladas (ojivas), es decir compara los índices de a través de los componentes básicos (media, ganancia esperada, pérdida esperada, etc.) de las diferentes secuencia analizadas en la simulación versus la secuencia óptima encontrada por el algoritmo Tabú Search; adicionalmente, confirma el incremento de la producción de los ítems, esto debido a la reducción del tiempo de procesamiento de todo el sistema real

Listado de Casos de Uso

1. Ingresar condiciones iniciales
2. Ingresar parámetros iniciales
3. Ingresar funciones de distribución para cada proceso
4. Ingresar funciones de distribución para tiempo de funcionamiento y reparación de cada equipo
5. Ejecución Tabú Search
6. Ver secuencia óptima
7. Ingreso secuencias a comparar
8. Ejecución simulación
9. Prueba de bondad de ajuste
10. Ver gráfico de secuencia

Caso de Uso 1: Ingresar condiciones iniciales

Descripción: Ingreso de condiciones iniciales: Productos y máquinas

Nota:

Se ingresa la cantidad de ítems que son producidos por todos los equipos, y la cantidad de máquina que requiere todo el proceso, e la elaboración de todos los ítems del proceso

Caso de Uso 2: Ingresar parámetros iniciales

Descripción: Ingreso de parámetros iniciales: Escenarios e iteraciones.

Nota:

Para el proceso de optimización del tiempo total empleado por los recursos del sistema, se requiere el ingreso del número de iteraciones, que al igual que la función de penalización, ayuda a determinar la convergencia del algoritmo, así como también su criterio de parada.

Caso de Uso 3: Ingresar funciones de distribución para cada proceso

Descripción: Ingresar Funciones de distribución de los procesos de producción

Nota:

A continuación procederemos a realizar el ingreso de todas las distribuciones que afecta el proceso de producción (proceso estocástico), funciones que están relacionadas al proceso de elaboración del ítem, como por ejemplo: Función de distribución Discreta, Exponencial, Uniforme, Normal, Triangular.

Caso de Uso 4: Ingresar funciones de distribución para tiempo de funcionamiento y reparación de cada equipo

Descripción: Ingreso de Funciones de distribución del tiempo de funcionamiento y reparación de cada equipo.

Nota:

En el siguiente evento, procederemos a ingresar las funciones de distribución tiempo de funcionamiento y reparación de los equipos que interviene en el proceso productivo, se ingresa el comportamiento del equipo frente a los procesos.

Caso de Uso 5: Ejecución Tabú Search**Descripción:** Procesamiento de la secuencia óptima**Nota:**

El proceso de mayor importancia es el encontrar las secuencias óptimas a lo largo de los n eventos, en este instante procederemos a buscar la secuencia para cada escenario y posteriormente determinar la secuencia que más se repite la cual será la secuencia óptima y en la mayoría de los escenarios, protegiéndose contra el peor de ellos.

Caso de Uso 6: Ver secuencia óptima**Descripción:** Mostrar secuencia óptima**Nota:**

Luego de hallar la secuencia, procederemos a mostrarla por pantalla, posterior a lo cual seleccionaremos secuencias alternas.

Caso de Uso 7: Ingreso secuencias a comparar**Descripción:** Selección de secuencia alternas, provista por el analista de proceso.**Nota:**

A continuación ingresamos secuencias alternas las mismas que las compararemos contra la secuencia óptima y su análisis estadístico deberá confirmar lo determinado por el algoritmo Tabú Search

Caso de Uso 8: Ejecución simulación**Descripción:** Ejecución Simulación

Nota: Como anteriormente indicamos luego de hallar la solución óptima el analista deberá comprobar si la solución arrojada por el algoritmo meta-heurístico es la correcta y la única forma de realizarlo es a través de la simulación estadística de los procesos que intervienen, la misma que sus componentes básicos deberán tener mayor ventaja sobre cualquier solución previamente ingresada (análisis Estadístico)

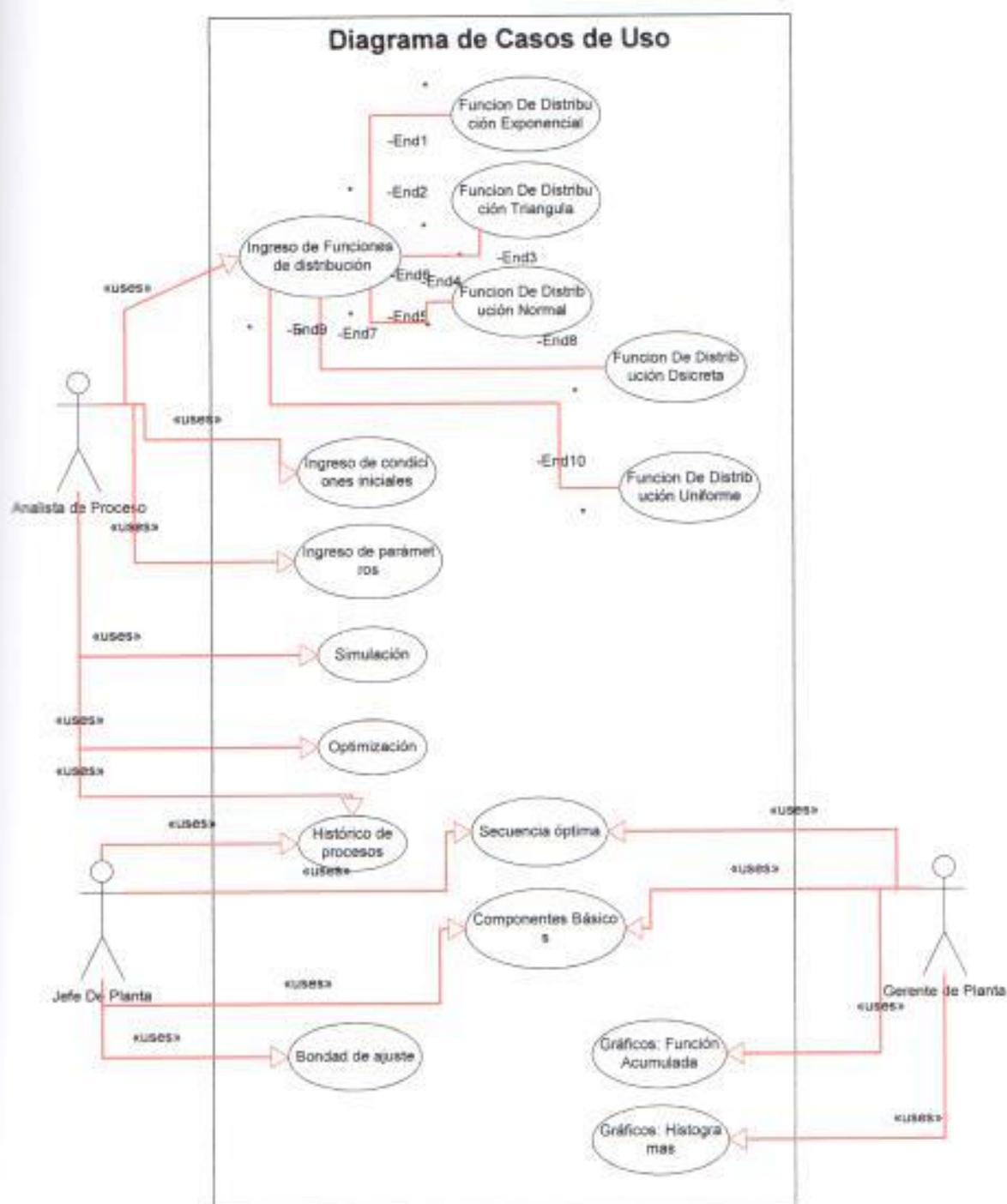
Caso de Uso 9: Prueba de bondad de ajuste**Descripción:** Prueba de Bondad de Ajuste**Nota:**

Si bien es cierto previamente al análisis del proceso se ha definido las semillas que generan las funciones de distribución de las variables de entrada, es decir se ha tomado por cierto que cada variable de entrada es independiente entre si, lo cual es fácilmente comprobable calculando la correlación de una variable con respecto a la otra, donde el valor de la correlación deberá ser igual a cero o aproximadamente cero esto ultimo es lo que se cumple en la realidad

Caso de Uso 10: Ver gráfico de secuencia**Descripción:** Gráficos: Histogramas, Percentil, Función Acumulada.**Nota:**

Como explicamos en el caso de uso anterior se puede comprobar el resultado arrojado por el algoritmo meta-heurístico, esto no solo se confirma mediante métodos analíticos, sino también mediante el análisis de los gráficos de las funciones acumuladas donde se analiza el porcentaje de riesgo del proyecto, como observaremos posteriormente el porcentaje de la secuencia óptima será el mayor, es decir existe una mayor ganancia, con la implementación de esta solución

DIAGRAMA DE CASOS DE USOS



Escenarios

- Caso de Uso 1: Ingreso de Parámetros
 - Escenario 1.1: Ingreso de Parámetros Exitoso
 - Escenario 1.2: Falta Ingresar Parámetros
- Caso de Uso 2: Ingresar condiciones iniciales
 - Escenario 2.1: Ingreso de condiciones iniciales Exitoso
 - Escenario 2.2: Falta Ingresar condiciones iniciales
- Caso de Uso 3: Ingresar funciones de distribución
 - Escenario 3.1: Ingreso parámetros función de distribución exponencial
 - Escenario 3.2: Falta Ingresar parámetros función de distribución exponencial
 - Escenario 3.3: Ingreso parámetros función de distribución uniforme
 - Escenario 3.4: Falta Ingresar parámetros función de distribución uniforme
 - Escenario 3.5: Ingreso parámetros función de distribución discreta
 - Escenario 3.6: Cálculo de Frecuencia Relativa
 - Escenario 3.7: Falta Ingresar parámetros función de distribución discreta
 - Escenario 3.8: Ingreso parámetros función de distribución triangular
 - Escenario 3.9: Falta Ingresar parámetros función de distribución triangular
 - Escenario 3.10: Ingreso parámetros función de distribución normal
 - Escenario 3.11: Falta Ingresar parámetros función de distribución normal
- Caso de Uso 4: Simulación
 - Escenario 4.1: Simulación Exitoso
 - Escenario 4.2: Simulación no procesada
- Caso de Uso 5: Optimización
 - Escenario 5.1: Optimización Exitoso
 - Escenario 5.2: Optimización no procesada
- Caso de Uso 6: Histórico de procesos
 - Escenario 6.1: Procesamiento de los datos históricos
 - Escenario 6.2: Opinión del experto
- Caso de Uso 7: Bondad de ajuste
 - Escenario 7.1: Cálculo bondad de ajuste
 - Escenario 7.2: Cálculo bondad de ajuste: forma de la función de distribución no corresponde
- Caso de Uso 8: Obtención de la secuencia óptima
 - Escenario 8.1: Secuencia óptima exitosa
 - Escenario 8.2: Procesamiento incompleto del algoritmo Tabú Search
 - Escenario 8.3: Parámetros de condicional iniciales mayor al número de semillas implementadas
- Caso de Uso 9: Cálculo de componentes básicos
 - Escenario 9.1: Cálculo de componentes básicos exitosa
 - Escenario 9.2: Procesamiento incompleto del cálculo de componentes básicos
- Caso de Uso 10: Gráficos: Histogramas, Percentil, Ojiva
 - Escenario 10.1: Presentación de gráficos exitosa
 - Escenario 10.2: Presentación de gráficos no satisfactoria

Escenario 1.1: Ingreso de Parámetros Exitoso**Actores que participan:**

Analista de procesos

Descripción: Ingreso Exitoso de los parámetros

Flujo de Eventos: El analista de procesos ingresa al sistema y ejecuta el algoritmo, posterior al cual debe ingresar la siguiente información:

Cantidad de máquinas.

Cantidad de productos.

Como se menciona en el desarrollo de la presente tesis, el presente trabajo es a nivel académico, lo cual tiene un limitante en la cantidad de máquinas y equipos, máximo 7 y 6 respectivamente o viceversa, esta restricción es fácil mejorarla (incremento de semillas), pero se lo deja, como parte de futuros mejores

Escenario 1.2: Falta Ingresar Parámetros**Actores que participan:**

Analista de procesos

Descripción: Falta parámetros por ingresar

Flujo de Eventos: El analista de procesos ingresa al sistema y ejecuta el algoritmo, posterior al cual debe ingresar la siguiente información:

Cantidad de máquinas.

Cantidad de productos.

Si el analista o persona que ejecuta la simulación, no ingresan todos los datos, el sistema presenta un mensaje de error.

Escenario 2.1: Ingreso de condiciones iniciales Exitoso**Actores que participan:**

Analista de procesos

Descripción: Ingreso de condiciones iniciales: Simulación y Optimización

Flujo de Eventos: El analista de procesos, ingresa las condiciones de la Simulación: número de escenarios y Optimización: Iteraciones de convergencia del algoritmo, cabe señalar que estos parámetros son de vital importancia, dados que ambos regulan el tiempo de ejecución del algoritmo, se puede con experiencia en el manejo de sistemas de simulación tener predeterminado un máximo número de escenarios y de esta forma el algoritmo, se ejecuta más eficientemente.

Escenario 2.2: Falta Ingresar condiciones iniciales**Actores que participan:**

Analista de procesos

Descripción: Ingreso de condiciones iniciales de la simulación y optimización

Flujo de Eventos: El analista ingresa al sistema, va al menú herramientas y selecciona la opción ingresar parámetros, posterior a ello ingresa el número de escenarios e iteraciones de convergencia, al no ingresar alguno de los parámetros, antes mencionado el algoritmo, muestra un mensaje de error: Faltan Parámetros por ingresar.

Escenario 3.1: Ingreso parámetros función de distribución exponencial**Actores que participan:**

Analista de proceso

Descripción: Ingreso de parámetros de distribución exponencial.

Flujo de Eventos: El analista selecciona la opción herramientas y escoge la opción Ingresar secuencia producto máquina, a continuación nos presenta una forma con los siguientes datos requeridos:

Producto, Máquina, Distribución.

Distribución: Exponencial

Ingresada esta información, seleccionamos el botón ingresar, ubicado en la parte superior, a continuación se refleja la información ingresada en la parte inferior de la pantalla. En esta sección existen cinco opciones y tres de las cuales, son importantes:

Parámetros, Eliminar último y Borrar secuencia.

- **Parámetros:** se presenta otra pantalla, en la cual para este caso, corresponde al valor de beta, información requerida en la distribución exponencial
- **Eliminar último:** Con esta opción podemos eliminar la última función de distribución ingresada.
- **Borrar secuencia:** Con esta opción el analista puede borrar todas las funciones de distribución ingresadas e iniciar nuevamente.

Escenario 3.2: Falta Ingresar parámetros función de distribución exponencial**Actores que participan:**

Analista de proceso

Descripción: Ingreso de parámetros función exponencial

Flujo de Eventos: Después de ingresar el tipo de la función de probabilidad se ingresa el parámetro, en el presente caso es beta, el cual de acuerdo al presente escenario el análisis obviado el ingreso de esta, con lo que el sistema arrojara un mensaje de error.

Falta ingresar parámetro Beta para la presente distribución.

Escenario 3.3: Ingreso parámetros función de distribución uniforme**Actores que participan:**

Analista de proceso

Descripción: Ingreso parámetros función uniforme

Flujo de Eventos: El analista selecciona la opción herramientas y escoge la opción Ingresar secuencia producto máquina, a continuación nos presenta una forma con los siguientes datos requeridos:

Producto, Máquina, Distribución.

Distribución: Uniforme

Ingresada esta información, seleccionamos el botón ingresar, ubicado en la parte superior, y a continuación se refleja la información ingresada en la parte inferior de la pantalla. En esta sección existen cinco opciones y tres de las cuales, son importantes:

Parámetros, Eliminar último y Borrar secuencia.

- **Parámetros:** se presenta otra pantalla, en la cual para este caso, corresponde al valor de límite superior y límite superior, información relacionada con la distribución uniforme.
- **Eliminar último:** Con esta opción podemos eliminar la última función de distribución ingresada.
- **Borrar secuencia:** Con esta opción el analista puede borrar todas las funciones de distribución ingresadas e iniciar nuevamente.

Escenario 3.4: Falta Ingresar parámetros función de distribución uniforme**Actores que participan:**

Analista de proceso

Descripción: Ingreso de parámetros función de distribución uniforme

Flujo de Eventos: Después de ingresar el tipo de la función de probabilidad se ingresan los parámetros, en el presente caso son límite superior y límite inferior, mismos que el análisis a obviado el ingreso de uno de estos, con lo que el sistema arrojará un mensaje de error.

Salida: Falta ingresar parámetros límite superior y/o límite inferior para la presente distribución.

Escenario 3.5: Ingreso parámetros función de distribución discreta**Actores que participan:**

Analista de proceso

Descripción: Ingreso parámetros función discreta.

Flujo de Eventos: El analista selecciona la opción herramientas y escoge la opción Ingresar secuencia producto máquina, a continuación nos presenta una forma con los siguientes datos requeridos:

Producto, Máquina, Distribución.

Distribución: Discreta

Ingresada esta información, seleccionamos el botón ingresar, ubicado en la parte superior, y a continuación se refleja la información ingresada en la parte inferior de la pantalla. En esta sección existen cinco opciones y tres de las cuales, son importantes:

Parámetros, Eliminar último y Borrar secuencia.

- **Parámetros:** se presenta otra pantalla, en la cual para esta caso, corresponde al valor de probabilidades, en esta opción una de las limitantes en el número de probabilidades a ingresar máximo 10 eventos discretos los cuales creemos cubrir con la mayoría de las necesidades. En esta opción nos presentan una pantalla con la siguiente información a ingresar Evento, Descripción del Evento, Probabilidad (el total de esta no necesariamente deberá ser el 100%), después de aceptar esta información, se presenta la siguiente pantalla de Cálculo de frecuencia relativa.
- **Eliminar último:** Con esta opción podemos eliminar la última función de distribución ingresada.
- **Borrar secuencia:** Con esta opción el analista puede borrar todas las funciones de distribución ingresadas e iniciar nuevamente.

Escenario 3.6: Cálculo de frecuencia relativa.**Actores que participan:**

Analista de proceso

Descripción: Cálculo de frecuencia relativa, de los eventos discretos ingresados

Flujo de Eventos: Posterior al ingreso de las probabilidades de todos los eventos de la distribución discreta, se realiza el cálculo de la frecuencia relativa es decir como sabemos las probabilidades de los eventos discretos no necesariamente sumarán el 100%, por lo que es necesario crear una subrutina que calcule la frecuencia relativa en base de la probabilidades ingresadas

Salida: Cálculo de la frecuencia relativa es satisfactoria.

Escenario 3.7: Falta Ingresar parámetros función de distribución discreta**Actores que participan:**

Analista de proceso

Descripción: Ingreso incorrecto de los parámetros de la distribución discreta.

Flujo de Eventos: El analista selecciona la opción herramientas y escoge la opción Ingresar secuencia producto máquina, a continuación nos presenta una forma con los siguientes datos requeridos:

Producto, Máquina, Distribución.

Distribución: Discreta

La información proporcionada por el analista es incorrecta o faltan datos relevantes a la función de distribución discreta. Ha seleccionado la opción parámetros de la distribución discreta y luego de ingresarlos, el sistema proporciona un mensaje de error.

Salida: Parámetros de la distribución discreta fueron ingresados incorrectamente.

Escenario 3.8: Ingreso parámetros función de distribución triangular**Actores que participan:**

Analista de procesos

Descripción: Ingreso parámetros función de distribución triangular

Flujo de Eventos: El analista selecciona la opción herramientas y escoge la opción Ingresar secuencia producto máquina, a continuación nos presenta una forma con los siguientes datos requeridos:

Producto, Máquina, Distribución.

Distribución: Triangular

Ingresada esta información, seleccionamos el botón ingresar, ubicado en la parte superior, y a continuación se refleja la información ingresada en la parte inferior de la pantalla. En esta sección existen cinco opciones y tres de las cuales, son importantes:

Parámetros, Eliminar último y Borrar secuencia.

➤ **Parámetros:** se presenta otra pantalla, en la cual para este caso, corresponde a las probabilidades de límite inferior, límite superior, límite medio.

➤ **Eliminar último:** Con esta opción podemos eliminar la última función de distribución ingresada.

➤ **Borrar secuencia:** Con esta opción el analista puede borrar todas las funciones de distribución ingresadas e iniciar nuevamente.

Escenario 3.9: Falta Ingresar parámetros función de distribución triangular**Actores que participan:**

Analista de Procesos

Descripción: Falta ingresar parámetros de la distribución Triangular

Flujo de Eventos: La información proporcionada por el analista es incorrecta, o falta por ingresar las probabilidades superior, inferior y media.

Salida: Se presenta una pantalla con la siguiente leyenda Falta por ingresar las probabilidades de la función de distribución triangular.

Escenario 3.10: Ingreso parámetros función de distribución normal**Actores que participan:**

Analista de Proceso

Descripción: Ingreso parámetros función de distribución normal

Flujo de Eventos: El analista selecciona la opción herramientas y escoge la opción Ingresar secuencia producto máquina, a continuación nos presenta una forma con los siguientes datos requeridos:

Producto, Máquina, Distribución.

Distribución: Normal

Registrada esta información, seleccionamos el botón ingresar, ubicado en la parte superior, y a continuación se refleja la información ingresada, ubicada en la parte inferior de la pantalla. En esta sección existen cinco opciones y tres de las cuales, son importantes:

Parámetros, Eliminar último y Borrar secuencia.

➤ **Parámetros:** se presenta otra pantalla, en la misma existe dos alternativas para el ingreso de los parámetros de la distribución normal:

a.- Ingreso de la media y la desviación estándar.

b.- Ingreso del límite inferior y límite superior.

➤ **Eliminar último:** Con esta opción podemos eliminar la última función de distribución ingresada.

➤ **Borrar secuencia:** Con esta opción el analista puede borrar todas las funciones de distribución ingresadas e iniciar nuevamente.

Salidas: Ingreso correcto

Escenario 3.11: Falta Ingresar parámetros función de distribución normal**Actores que participan:**

Analista de Proceso

Descripción: Se ingresa incorrectamente o falta ingresar parámetros de la función de distribución normal

Flujo de Eventos: La información proporcionada por el analista es incorrecta, o falta por ingresar las probabilidades de acuerdo a las opciones:

a.- Ingreso de la media y la desviación estándar.

b.- Ingreso del límite inferior y límite superior.

Salida: Se presenta una pantalla con la siguiente leyenda "Falta por ingresar parámetros de la función de distribución normal".

Escenario 4.1: Optimización Exitoso**Actores que participan:**

Analista de Proceso

Descripción: Inicio de la optimización Tabú Search.

Flujo de Eventos: Posterior al ingreso de las condiciones iniciales y las funciones de distribución: combinación ítem – máquina, seleccionamos la opción ejecuciones, a continuación se presenta un menú colgante con las siguientes opciones:

a.- Tabú Search.

b.- Simulación.

c.- Prueba de bondad de ajuste.

Para este escenario, escogemos la opción Tabú Search, con lo cual iniciamos el proceso de Simulación y Optimización (obtención de la solución óptima).

Salidas: Pantalla de Inicio de simulación, en la cual presenta, tres opciones:

a.- Iniciar Tabú: Private Sub b_iniciar(Index As Integer)

b.- Ver Secuencia Óptima: Private Sub b_verSecuencia()

c.- Salir: Private Sub b_salir_eso(Index As Integer)

Posterior, a la elección de la segunda opción, nos presenta el siguiente mensaje: "Optimización Terminada"

Escenario 4.2: Optimización no procesada**Actores que participan:**

Analista de Proceso

Descripción: Luego de ingresar las funciones de distribución se ejecuta la simulación

Flujo de Eventos: Posterior al ingreso de las condiciones iniciales y los parámetros de las funciones de distribución: combinación ítem - máquina, seleccionamos la opción ejecuciones, a continuación se presenta un menú colgante con las siguientes opciones:

- a.- Tabú Search.
- b.- Simulación.
- c.- Prueba de bondad de ajuste.

Para este escenario, escogemos la opción Tabú Search, con lo cual iniciamos el proceso de Simulación y Optimización (obtención de la solución óptima).

Salidas: Pantalla de Inicio de simulación, en la cual presenta, tres opciones:

- a.- Iniciar Tabú: Private Sub b_iniciar(Index As Integer)
- b.- Ver Secuencia Óptima: Private Sub b_verSecuencia()
- c.- Salir: Private Sub b_salir_eso(Index As Integer)

Posterior, a la selección de la segunda opción, nos presenta el siguiente mensaje: "Optimización no Procesada", demasiados equipos o ítems ingresados.

Escenario 5.1: Simulación Exitoso**Actores que participan:**

Analista de Proceso

Descripción: Luego de obtener la solución óptima, procedemos a ejecutar la simulación.

Flujo de Eventos: Después de obtener la secuencia óptima, seleccionamos la opción ejecuciones, a continuación se presenta un menú colgante con las siguientes opciones:

- a.- Tabú Search.
- b.- Simulación.
- c.- Prueba de bondad de ajuste.

Para este escenario, escogemos la opción Simulación, con lo cual iniciamos el proceso de Simulación (comprobación de la solución óptima).

Salidas: Pantalla de Inicio de simulación, en la cual presenta, tres opciones:

- a.- Iniciar simulación: Private Sub b_iniciar(Index As Integer)
- b.- Graficar: Private Sub b_graficar()
- c.- Salir: Private Sub b_salir_eso(Index As Integer)

Posterior, a la selección de la segunda opción, nos presenta el siguiente mensaje: "Simulación Terminada"

Escenario 5.2: Simulación no procesada**Actores que participan:**

Analista de Proceso

Descripción: Luego de ingresar las funciones de distribución se ejecuta la simulación

Flujo de Eventos: Después de obtener la secuencia óptima, seleccionamos la opción ejecuciones, a continuación se presenta un menú colgante con las siguientes opciones:

- a.- Tabú Search.
- b.- Simulación.
- c.- Prueba de bondad de ajuste.

Para este escenario, escogemos la opción Simulación, con lo cual iniciamos el proceso de Simulación (comprobación de la solución óptima).

Salidas: Pantalla de Inicio de simulación, en la cual presenta, tres opciones:

- a.- Iniciar simulación: Private Sub b_iniciar(Index As Integer)
- b.- Graficar: Private Sub b_graficar()
- c.- Salir: Private Sub b_salir_eso(Index As Integer)

Posterior, a la selección de la segunda opción, nos presenta el siguiente mensaje: "Simulación no Procesada"

Escenario 6.1: Procesamiento de los datos históricos**Actores que participan:**

Analista de Proceso y Jefe de planta

Descripción: Definición de las funciones de distribución de las variables de entrada

Flujo de Eventos: Este es un proceso manual, con el cual se define el comportamiento de las funciones de distribución (variables de entrada del sistema), sin el cual no se puede ingresar los parámetros iniciales de la optimización.

Salidas: Es proceso es la base para el ingreso de las funciones de distribución

Escenario 6.2: Opinión del experto**Actores que participan:**

Analista de Proceso y Jefe de planta

Descripción: Definición de las función de distribución, opinión del experto

Flujo de Eventos: Existen algunos procesos en los cuales no existe documentación de los procesos, en estos no se podría definir la forma de la función de distribución, por lo cual es necesario en conjunto con el jefe de planta definir el comportamiento de estas variables de entrada.

Salidas: Obtener de acuerdo al criterio del experto la forma de la función de distribución la cual posterior deberá ser ingresada al sistema.

Escenario 7.1: Cálculo bondad de ajuste: correlación aproximadamente cero

Actores que participan:

Jefe de planta

Descripción: Determinar independencia de las variables de entrada

Flujo de Eventos: Este proceso puede ser realizado en cualquier instancia del proceso debido que es para confirmar la independencia de las variables de entrada, lo cual previamente ha sido comprobado, pero es necesario para satisfacción del analista confirmar dicha condición de independencia, adicionalmente confirma si el comportamiento de variable de entrada corresponde a lo que indica la función de distribución

Salidas: Seleccionamos la opción Ejecuciones: y nos despliega las siguientes opciones:

a.- Tabú Search.

b.- Simulación.

c.- Prueba de bondad de ajuste.

Seleccionando la opción: Prueba de bondad de ajuste

Private Sub bondadA()

Nos presenta una pantalla en la cual podemos ingresa la distribución (datos del comportamiento de la funciones a comparar) y procede a determinar a cual distribución más se asemeja el comportamiento de la variable de entrada.

Escenario 7.2: Cálculo bondad de ajuste: no corresponde a la forma de la función de distribución

Actores que participan:

Jefe de planta

Descripción: El comportamiento de la variable de entrada no corresponde a la forma función de distribución.

Flujo de Eventos: Este proceso puede ser realizado en cualquier instancia del proceso debido que es para confirmar la independencia de las variables de entrada, lo cual previamente ha sido comprobado, pero es necesario para satisfacción del analista confirmar dicha condición de independencia, adicionalmente confirma si el comportamiento de variable de entrada corresponde a lo que indica la función de distribución

Salidas: Seleccionamos la opción Ejecuciones: y nos despliega las siguientes opciones:

a.- Tabú Search.

b.- Simulación.

c.- Prueba de bondad de ajuste.

Seleccionando la opción: Prueba de bondad de ajuste

Private Sub bondadA()

Nos presenta una pantalla en la cual podemos ingresa la distribución (datos del comportamiento de la funciones a comparar) y procede a determinar a cual distribución más se asemeja el comportamiento de la variable de entrada, el cual no se asemeja a ninguna de las antes estudiadas y se presenta un mensaje en el indica vuelva a procesar, comportamiento de la variable de entrada no corresponde.

Escenario 8.1: Secuencia óptima exitosa**Actores que participan:**

Analista de Proceso y Jefe de planta

Descripción: Encontrada solución óptima

Flujo de Eventos: Luego de terminada la optimización procedemos a visualizar la secuencia óptima, hallada por el proceso de optimización

```
Private Sub mnuToolsEjectarTabu()
```

```
frmSolucion.Show vbModal, Me
```

```
Private Sub b_verSecuencia()
```

```
frmSecuenciaOptima.Show vbModal, Me
```

Salidas: Posterior a la finalización del algoritmo procedemos a seleccionar la opción Ver secuencia óptima, con lo cual nos se ejecuta el siguiente proceso:

```
Private Sub b_verSecuencia()
```

```
frmSecuenciaOptima.Show vbModal, Me
```

Para finalmente mostrarnos la secuencia óptima.

Escenario 8.2: Procesamiento incompleto del algoritmo Tabú Search**Actores que participan:**

Analista de Proceso y Jefe de planta

Descripción: Tabú Search no termina de procesar, no es posible hallar solución óptima.

Flujo de Eventos: Luego de terminada la optimización procedemos a visualizar la secuencia óptima, hallada por el proceso de optimización

```
Private Sub mnuToolsEjectarTabu()
```

```
frmSolucion.Show vbModal, Me
```

```
Private Sub b_verSecuencia()
```

```
frmSecuenciaOptima.Show vbModal, Me
```

Salidas: Debido al ingreso de demasiados escenarios, para el proceso de simulación el algoritmo no termina por falta de recursos del computador, acordarse que las operaciones se las realiza en memoria.

Mensaje de Error: Falta de recursos del sistema

Escenario 8.3: Parámetros de condicional iniciales mayor al número de semillas implementadas**Actores que participan:**

Analista de Proceso y Jefe de planta

Descripción: Parámetros de condiciones iniciales son incorrectos

Flujo de Eventos: Luego de terminada la optimización procedemos a visualizar la secuencia óptima, hallada por el proceso de optimización

```
Private Sub mnuToolsEjectarTabu()
```

```
frmSolucion.Show vbModal, Me
```

```
Private Sub b_verSecuencia()
```

```
frmSecuenciaOptima.Show vbModal, Me
```

Salidas: Después de seleccionar

Ver secuencia óptima,

```
Private Sub b_verSecuencia(),
```

```
frmSecuenciaOptima.Show vbModal, Me,
```

Nos presenta el mensaje de error, se ingreso demasiados productos o máquinas, el máximo a ingresar en 50 variables de entradas es decir una matriz de 7 * 7 (items, equipos)

Escenario 9.1: Cálculo de componentes básicos exitosa**Actores que participan:**

Gerente de planta y Jefe de planta

Descripción: Cálculo de los componentes básicos

Flujo de Eventos: Posterior a la ejecución de la optimización (hallar secuencia óptima) y la simulación (comprobación estadística de la secuencia óptima), en esta última realizamos la comprobación de la secuencia óptima no solo en forma gráfica sino que adicionalmente lo comprobamos de la forma analítica a través de sus componentes básicos: media, esperanza, desviación estándar, ganancia esperada, pérdida esperada, valor esperado.

Opciones:

Ver

Secuencia óptima.

Graficar secuencia.

Private Sub Grafica()

frmChart.Show

Private Sub Histograma()

Private Sub Ojiva()

Private Sub Cajas()

Salidas: Como indicamos obtenemos los componentes básicos, los cuales se mostrarán en la misma pantalla de los gráficos estadísticos: histogramas, percentil, ojivas.

Escenario 9.2: Procesamiento incompleto del cálculo de componentes básicos**Actores que participan:**

Gerente de planta y Jefe de planta

Descripción: Cálculo de los componentes básicos es incompleto.**Flujo de Eventos:****Opciones:**

Ver:

Secuencia óptima.

Graficar secuencia.

Private Sub Grafica()

frmChart.Show

Private Sub Histograma()

Private Sub Ojiva()

Private Sub Cajas()

Salidas: Debido al ingreso incorrecto de la secuencia alternas no se puede calcular los componentes básicos y con lo cual no se realiza el análisis gráfico, ni estadístico.

Escenario 10.1: Presentación de gráficos exitosa**Actores que participan:**

Gerente de planta y Jefe de planta

Descripción: Presentación de los gráficos es correcta**Flujo de Eventos:** A continuación seleccionamos el módulo de gráficos, con el cual procedemos a realizar las comparaciones entre la secuencia óptima y las secuencias escogidas al azar.

Opciones:

Ver:

Secuencia óptima.

Graficar secuencia.

Private Sub Grafica()

frmChart.Show

Private Sub Histograma()

Private Sub Ojiva()

Private Sub Cajas()

Salidas: Se presenta la pantalla de los gráficos estadísticos:

Private Sub chtSample_LostFocus(), con el cual procedemos a realizar el análisis estadístico y gráfico.

Escenario 10.2: Presentación de gráficos no satisfactoria**Actores que participan:**

Gerente de planta y Jefe de planta

Descripción:**Flujo de Eventos:** A continuación seleccionamos el módulo de gráficos, con el cual procedemos a realizar las comparaciones entre la secuencia óptima y las secuencias escogidas al azar

Opciones:

Ver:

Secuencia óptima.

Graficar secuencia.

Private Sub Grafica()

frmChart.Show

Private Sub Histograma()

Private Sub Ojiva()

Private Sub Cajas()

Salidas: Se presenta la pantalla de los gráficos estadísticos:

Private Sub chtSample_LostFocus(),

Por no terminar el proceso de simulación, los gráficos estadísticos no se presentan e igual sucede con los componentes básicos, esto debido, que se interrumpió la ejecución por el usuario, ya sea por corrección en las condiciones iniciales o mal ingreso de los funciones de distribución, comportamiento de las variables de entrada.

Flujo de Ventana y Layout

Pantalla de Presentación

Herramientas

- Ingreso de parámetros iniciales
- Ingresar secuencia producto máquina
 - Ingresar función de distribución uniforme
 - Ingresar función de distribución discreta
 - Ingresar función de distribución exponencial
 - Ingresar función de distribución triangular
 - Ingresar función de distribución normal
- Ingreso de tiempo máquina
 - Ingresar función de distribución uniforme
 - Ingresar función de distribución discreta
 - Ingresar función de distribución exponencial
 - Ingresar función de distribución triangular
 - Ingresar función de distribución normal

Ejecución

- Tabú Search

Herramientas

- Ingreso de secuencia a comparar

Ver

- Secuencia Óptima

Ejecución

- Simulación
- Prueba de Bondad de ajuste

Ver

- Gráficos: Función acumulada

Archivo

- Salir

Ayuda

- Acerca de

Pantalla de Presentación



Herramientas

Ingreso de parámetros iniciales

Parámetros Iniciales

Parámetros

Cantidad de Productos:

Cantidad de Máquinas:

Número de Escenarios:

Número Iteraciones:

Aceptar Salir

Ingresar secuencia producto máquina

Ingresar Distribución

Ingreso:

Producto: Ingreso

Máquina:

Distribución: Ingreso

Secuencia de Procesos

Seq.	Máquina	Distribución	Parámetros

Parámetros

Eliminar Línea

Borrar secuencia

Aceptar Salir

Ingreso de parámetros: Función de distribución Uniforme

Uniforme

Parámetros

$F(x) = 1/(b-a)$ entre (a,b)

a: b:

Aceptar Salir

Ingreso de parámetros: Función de distribución Discreta

The 'Discreta' window contains the following fields and controls:

- Ingreso de Eventos:**
 - Num. De Eventos: Evento 2 (dropdown)
 - Evento: Lluvia (text input)
 - Frecuencia: 34 (text input)
 - Ingresar (button)
- Table:**

No.	Evento	Frecuencia	Prob. Relativa
Evento 1	1	23	
Evento 2	Lluvia	34	
- Acceptar (button) and Salir (button) at the bottom.

Cálculo de probabilidad relativa

The 'OptimizadorProcesos' window displays the message: "Se ha realizado el calculo relativo de la frecuencia de los eventos".

Acceptar (button)

Función de distribución Exponencial

The 'Probabilidad Exponencial' window contains the following fields and controls:

- Funcion a evaluar:**

$$F(x) = (1/B) * e^{-(x/B)}$$
- B:** [text input field]
- Acceptar (button) and Salir (button) at the bottom.

Ingreso de parámetros: Función de distribución Triangular

The 'Triangular' window contains the following fields and controls:

- Parámetros requeridos:**

$$F(x)1 = (2/(c-a))*(b-a) * (x-a)$$

$$F(x)2 = - (2/(c-a))*(c-b) * (x-c)$$
- a:** [text input field] **b:** [text input field] **c:** [text input field]
- Acceptar (button) and Salir (button) at the bottom.

Ingreso de parámetros: Función de distribución Normal

Ingreso de Parámetros

$$Z1 = \text{SQRT}(-2 * \text{Ln}(\mu1)) * \text{Cos}(2 * \text{Pi} * \mu2)$$

Media y varianza
 Máximos y mínimos

μ : Min:
 Desv: Max:

Ingreso de tiempo máquina

Ingreso de tiempo de máquina

Ingreso de Tiempo:

Máquina: Ingresar

Tiempo:

Distribución: [Aceptar]

Tiempo:

Tiempo	Distribución	Parámetros
Funcionamiento		
Reparación		

Ejecución

Tabú Search

Ejecución de Selección Optima...

Herramientas

Ingreso de secuencia a comparar



Ver

Secuencia Óptima

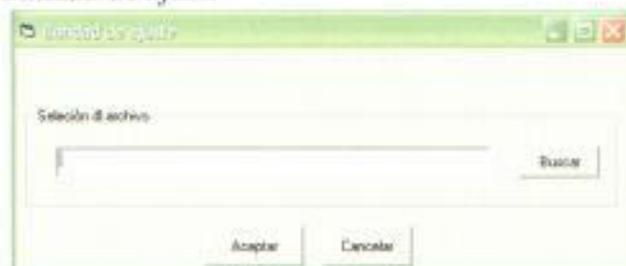


Ejecución

Simulación

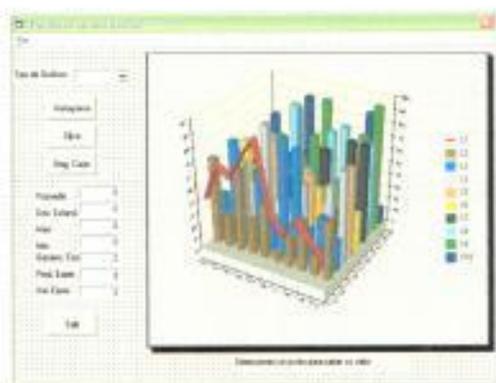
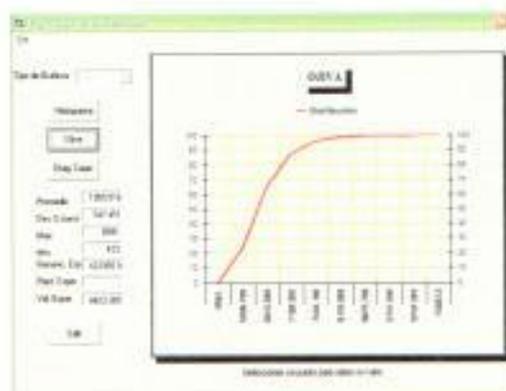
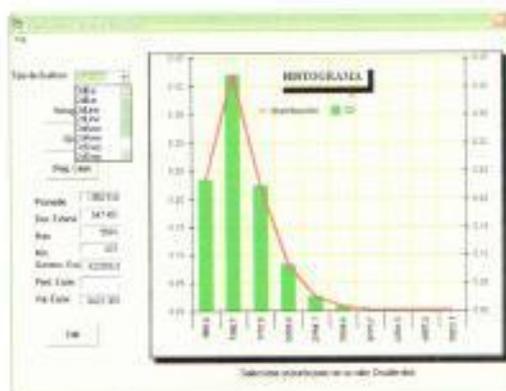


Prueba de Bondad de ajuste

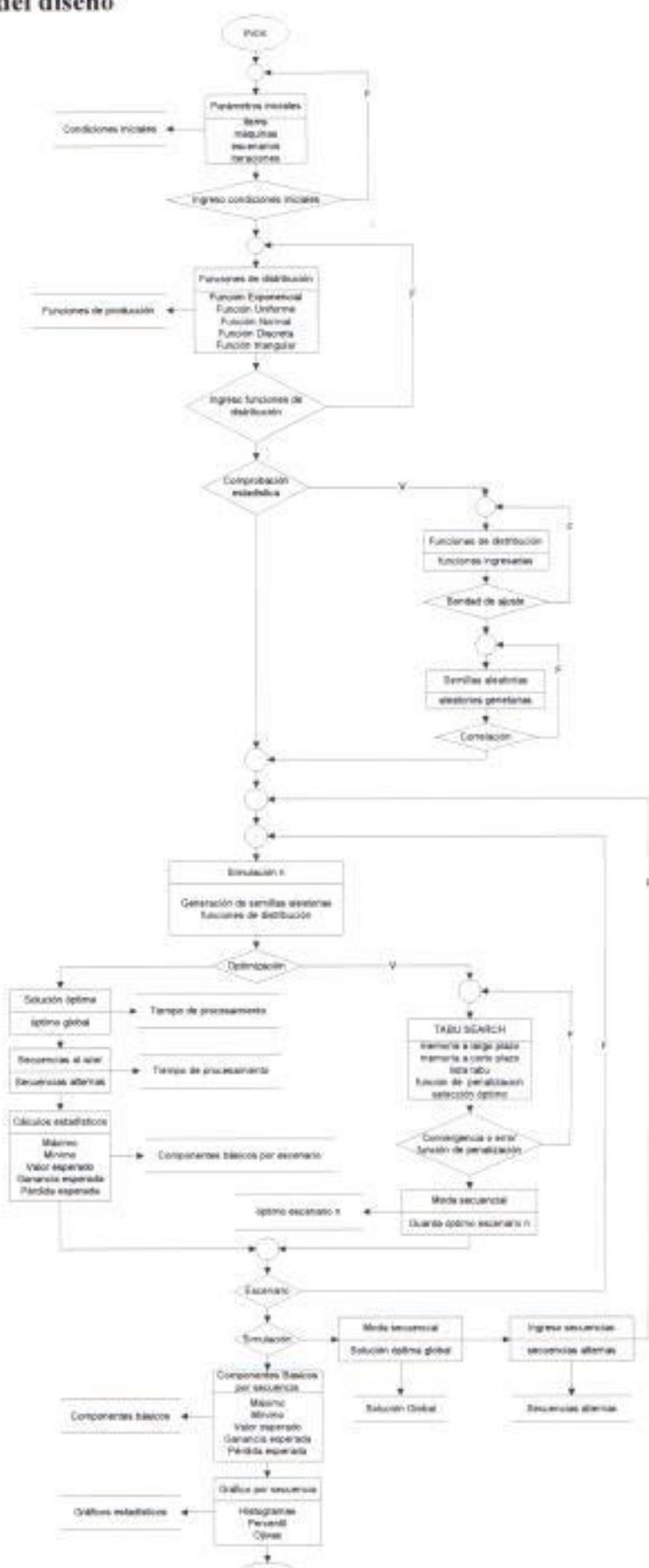


Ver

Gráficos:
 Histogramas
 Diagrama de cajas
 Función acumulada

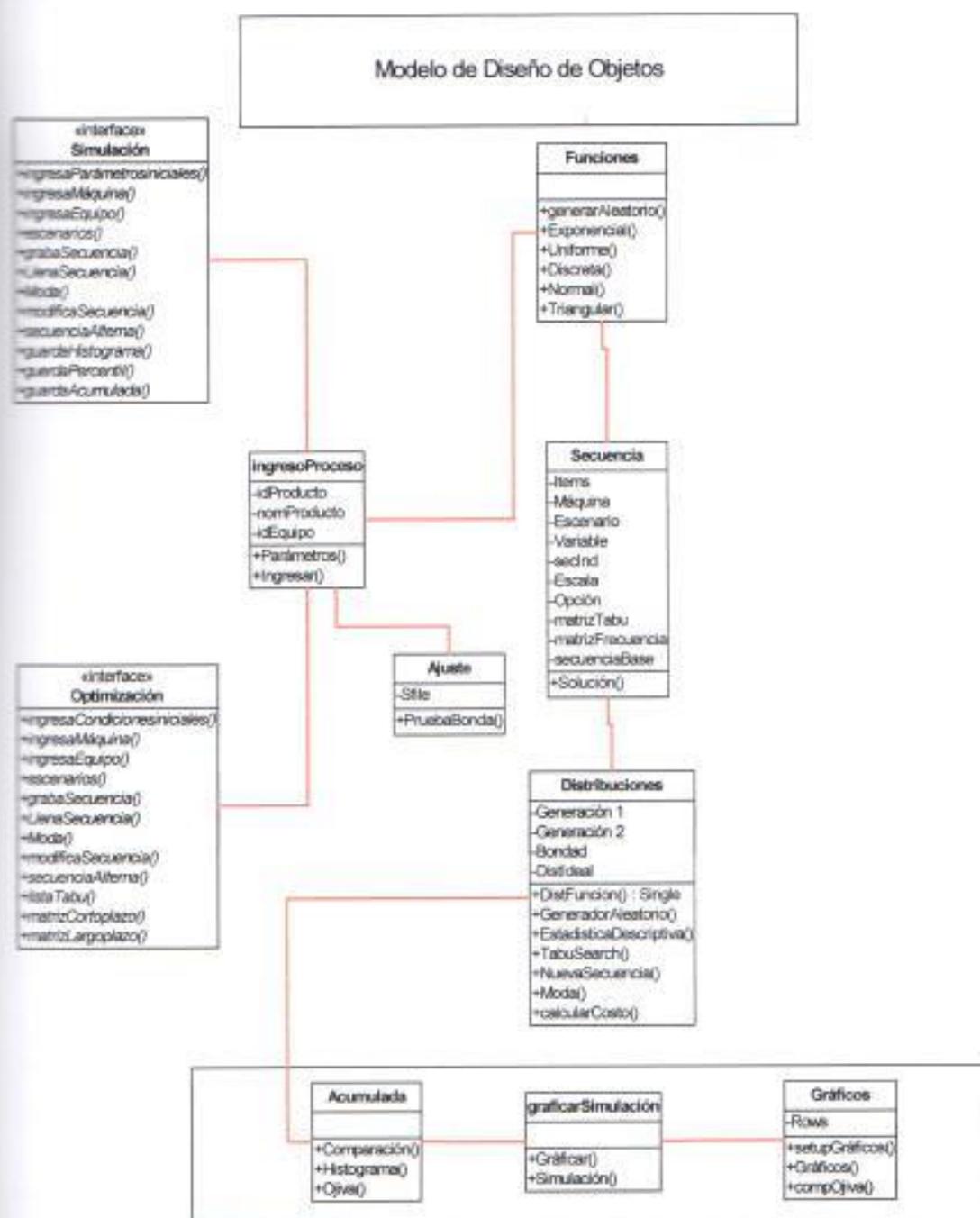


4.2 Modelo de Diseño Visión General del diseño



Modelos Estáticos

Modelo de Diseño de Objetos



Descripción de Clases

Distribución

Atributos

Public ale() As Double
 Public AleN() As Double
 Public nale As Long
 Public banGen As Integer
 Public Generacion1() As Single
 Public Generacion2() As Single
 Public Bondad() As Single
 Public pobx() As Single
 Public DistIdeal() As Single

Operaciones

Public Sub DistFuncion(ByVal beta As Single, ByVal lminF As Single, ByVal lmedF As Single, ByVal lmaxF As Single, ByVal i As Long, ByVal media As Single, ByVal desv As Single, ByVal cont As Integer)
 Public Sub GenerarAleatorio(opcion As Integer)
 Private Sub EstadistDescriptiva()
 Public Sub GanPerEsperada()
 Public Sub OrdCaj()
 Public Sub Poblacion()
 Private Sub Rango()
 Private Function TabuSearch(iter As Long)
 Public Function NuevaSecuencia(sequencia As Variant, Ind As Integer, MatrizTabu As Variant, MatrizFrecuencia As Variant)
 Public Function selectSecuenciaModa(matSoluciones As Variant) As Integer()
 Private Function costoTiempo(SecSeleccionada As Variant, iter As Long)
 Public Sub DesvStand()
 Public Function calcularCosto(solucionCandidata As Variant, sequencia As Variant, mTiempoMaquina As Variant, distRepMaquina As Variant) As Long
 Public Sub PrueBondad(ByVal sfile As String)
 Private Sub DistFuncionBJ(ByVal beta As Single, ByVal lminF As Single, ByVal lmedF As Single, ByVal lmaxF As Single, ByVal i As Long, ByVal media As Single, ByVal desv As Single, ByVal cont As Integer)
 Private Sub PoblacionBJI(ByVal liminf As Single, ByVal clas As Single, ByVal escala As Integer, ByVal i As Long, ByVal cont As Integer, ByVal media As Single, ByVal desv As Single)
 Public Sub PoblacionBJ(ByVal escala As Integer)
 Private Sub DifPrueba(ByVal escala As Integer)
 Private Sub OrdenaReal()
 Private Sub RangoBJ(ByVal escala As Integer)
 Private Function clase(ByVal beta As Single, ByVal lminF As Single, ByVal lmedF As Single, ByVal lmaxF As Single, ByVal escala As Integer, ByVal media As Single, ByVal desv As Single) As Single

Gráficos**Atributos**

```
Public intRows
Public ban As Integer
Public banhoc As Integer
```

Operaciones

```
Public Sub SetupChart()
Public Sub Chart(ByVal Rango As Variant, ByVal Datos As Variant, colV As Integer, chtTitle As String, colLabel As String)
Public Sub twoColChart(ByVal Rango As Variant, ByVal Datos As Variant, colV As Integer, chtTitle As String, colLabel As String)
Public Sub threeColChart(colV As Integer)
Public Sub CompOjiva()
Public Sub ChartCaj(ByVal col As Integer, chtTitle As String, colLabel As String)
```

Ajuste**Atributos**

```
Public sfile As String
```

Operaciones

```
Public Sub PrueBondad()
```

Discreta**Atributos**

```
Private valTot As Single
Private evIngresado As Integer
```

Operaciones

```
Private Sub imprimeCabecera()
Private Sub imprimeCabeceraDist()
Private Sub ingresaEvento()
Private Sub calculaRelativa()
Private Sub bIngresarEvento ()
Private Sub cNumEvento ()
Private Sub D_aceptar()
Private Sub GenerarAleatorio()
```

EditarMaquinas**Atributos**

```
Public filaSel As Integer
```

Operaciones

```
Private Sub llenaMaquinas()
Private Sub llenaDistribucion()
Private Sub bAceptar()
Private Sub bIngresar()
Private Sub bParametros()
```

```

Public Sub parIngresados()
Private Sub bSalir()
Private Sub cbMaquina()
Private Sub traeDatos()
Private Sub cabeceraMaquinas()
Public Sub ingresaTiempos()
Public Sub limpiatiempos()

```

Exponencial

Atributos

Operaciones

```

Private Sub GenerarAleatorio ()
Private Sub Imprime_Cabecera()
Private Sub E_Aceptar()
Private Sub DatosG ()

```

GráficoAcumulada

Atributos

```

Private cmdLastClicked As CommandButton
Private Const Help

```

Operaciones

```

Private Sub Cajas()
Private Sub chtSample_LostFocus()
Private Sub chtSample_PointActivated(Series As Integer, DataPoint As Integer,
MouseFlags As Integer, Cancel As Integer)
Private Sub chtSample_PointSelected(Series As Integer, DataPoint As Integer,
MouseFlags As Integer, Cancel As Integer)
Private Sub cmbType_LostFocus()
Private Sub Comparacion()
Private Sub cmbRows()
Private Sub cmbType()
Private Sub mnuContents()
Private Sub Histograma()
Private Sub mnuExit()
Private Sub Ojiva()
Private Sub Var1()
Private Sub Var10()
Private Sub Var2()
Private Sub Var3()
Private Sub Var4()
Private Sub Var5()
Private Sub Var6()
Private Sub Var7()
Private Sub Var8()
Private Sub Var9()
Private Sub Var11()

```

```

Private Sub Var12()
Private Sub Var13()
Private Sub Var14()
Private Sub Var15()
Private Sub Var16()
Private Sub Var17()
Private Sub Var18()
Private Sub Var19()
Private Sub Var20()
Private Sub menu()
Private Sub inicial()

```

IngresoProceso

Atributos

```

Public idProducto As String
Public nomProducto As String
Public productoActual As String
Public filaSel As Integer
Public id_Producto, sec As Integer

```

Operaciones

```

Private Sub bParametros()
Private Sub Edit()
Private Sub llenaProductos()
Private Sub llenaMaquinas()
Private Sub llenaDistribucion()
Private Sub imprimeCabecera()
Private Sub bIngresar()
Private Sub ingresaProceso()
Private Sub bEliminar()
Private Sub borrar_grid()
Public Sub parIngresados()
Public Function getIdProducto(producto As String) As Integer
Public Function getIdMaquina(maquina As String) As Integer
Public Function getProducto(producto As String) As String
Public Function getMaquina(maquina As String) As String
Private Sub bPrincipal()
Private Sub cbProducto()
Private Sub traeDatos()
Private Sub bModificar()
Private Sub bSecuencia()
Public Sub grabaMAtriz()
Public Sub llenasecuencia()
Public Sub obtProducto()
Public Sub guardaEnMAtriz()
Public Sub infoProducto()
Public Function getMTiempoMaquina() As String

```

InicioSimulacion**Atributos****Operaciones**

```
Private Sub b_graficar()
Private Sub b_iniciar(Index As Integer)
Private Sub b_salir_eso(Index As Integer)
```

Principal**Atributos****Operaciones**

```
Private Sub bondadA()
Private Sub Grafica()
Private Sub ingresoSecuencias()
Private Sub ingresoTiempos()
Private Sub mnuEjecutarSimulacion()
Private Sub mnuFileExit()
Private Sub mnuHelpAbout()
Private Sub mnuToolsEjectarTabu()
Private Sub mnuToolsParametrosIniciales()
Private Sub mnuViewGraficosSecuencias()
Private Sub mnuViewSecuenciaOptima()
```

ParametrosIniciales**Atributos****Operaciones**

```
Private Sub bAceptar()
Private Sub bIngreso()
Private Sub bSalir()
```

SecuenciaOptima**Atributos****Operaciones**

```
Private Sub bPrincipal(Index As Integer)
Public Sub Imprime_Cabecera()
Private Sub llenaSolucion(secuenciaPMaquina() As Integer)
```

Solucion**Atributos****Operaciones**

```
Private Sub b_iniciar(Index As Integer)
Private Sub b_salir(Index As Integer)
```

```
Private Sub b_verSecuencia()
```

GraficoBondad

Atributos

Operaciones

```
Private Sub Iniciar()
Private Sub Salir()
```

Normal

Atributos

Operaciones

```
Private Sub Imprime_Cabecera()
Private Sub GenerarAleatorio ()
Private Sub DatosG ()
Private Sub N_aceptar()
```

Resultados

Atributos

Operaciones

```
Private Sub Salir()
Private Sub cabecera()
Private Sub llenaGrid()
Private Sub Graficar()
```

Secuencias

Atributos

```
Public filaSel As Integer
```

Operaciones

```
Private Sub bAceptar()
Private Sub bSalir()
```

```
Private Sub bSelect()
Private Sub ingresaSecSelect ()
Private Sub imprimeCabeceras ()
Public Function llenaSecuencias(sec As Variant)
Private Sub ingresaSecSelect(secuencia As String)
Private Sub bEliminar()
Private Sub borrar_grid()
Public Sub llenaMatSelect()
```

Triangular**Atributos****Operaciones**

Private Sub Salir()
 Private Sub Imprime_Cabecera()
 Private Sub DatosG ()
 Private Sub T_aceptar()

Uniforme**Atributos**

Public filaSel As Integer

Operaciones

Private Sub Salir()
 Private Sub Imprime_Cabecera()
 Private Sub DatosG ()
 Private Sub U_aceptar()

Secuencia**Atributos**

Public numProductos As Integer
 Public numMaquinas As Integer
 Public numEscenarios As Long
 Public variables() As String
 Public cliente As String
 Public MatSecuencia() As Integer
 Public Solucion() As Integer
 Public DisProbOut() As String
 Public EstDistOut() As Single
 Public vectorEscojidos() As Integer
 Public DisProb() As String
 Public MaxIteraciones As Integer
 Public intRows1 As Long
 Public intR As Long
 Public SecInd(1 To 250, 1 To 4) As Integer
 Public Cajx() As Single
 Public MatPob() As Single
 Public MatRag() As String
 Public MatOji() As Single
 Public MatOjiRan() As String
 Public MatCaj() As Single
 Public MatPobBJ() As Single
 Public MatRagBJ() As Single
 Public DistProdMaquina() As Long
 Public DistFunReparMaquina() As Long
 Public MatPobBJI() As Single
 Public nEscogidos As Integer

Public selGraficar() As Integer
 Public escala As Integer
 Public opcion As Integer
 Public SolOptima() As Integer
 Public secuenciaNueva() As Integer
 Public MatrizTabu() As Integer
 Public MatrizFrecuencia() As Integer
 Public secBase() As Integer
 Public DBonj() As Single
 Public indiceT As Integer
 Public EscalaBG As Integer

Operaciones

Public Sub secuencia()

Modelos Dinámicos

Diagramas de Interacción de Objetos

- Caso de Uso 1: Ingreso de Parámetros
 - DIO 1.1: Ingreso de Parámetros Exitoso
 - DIO 1.2: Falta Ingresar Parámetros
- Caso de Uso 2: Ingresar condiciones iniciales
 - DIO 2.1: Ingreso de condiciones iniciales Exitoso
 - DIO 2.2: Falta Ingresar condiciones iniciales
- Caso de Uso 3: Ingresar funciones de distribución
 - DIO 3.1: Ingreso parámetros función de distribución exponencial
 - DIO 3.2: Falta Ingresar parámetros función de distribución exponencial
 - DIO 3.3: Ingreso parámetros función de distribución uniforme
 - DIO 3.4: Falta Ingresar parámetros función de distribución uniforme
 - DIO 3.5: Ingreso parámetros función de distribución discreta
 - DIO 3.6: Cálculo de Frecuencia Relativa
 - DIO 3.7: Falta Ingresar parámetros función de distribución discreta
 - DIO 3.8: Ingreso parámetros función de distribución triangular
 - DIO 3.9: Falta Ingresar parámetros función de distribución triangular
 - DIO 3.10: Ingreso parámetros función de distribución normal
 - DIO 3.11: Falta Ingresar parámetros función de distribución normal
- Caso de Uso 4: Simulación
 - DIO 4.1: Simulación Exitoso
 - DIO 4.2: Simulación no procesada
- Caso de Uso 5: Optimización
 - DIO 5.1: Optimización Exitoso
 - DIO 5.2: Optimización no procesada
- Caso de Uso 6: Histórico de procesos
 - DIO 6.1: Procesamiento de los datos históricos

- DIO 6.2: Opinión del experto
- Caso de Uso 7: Bondad de ajuste
 - DIO 7.1: Cálculo bondad de ajuste
 - DIO 7.2: Cálculo bondad de ajuste: forma de la función de distribución no corresponde
- Caso de Uso 8: Obtención de la secuencia óptima
 - DIO 8.1: Secuencia óptima exitosa
 - DIO 8.2: Procesamiento incompleto del algoritmo Tabú Search
 - DIO 8.3: Parámetros de condicional iniciales mayor al número de semillas implementadas
- Caso de Uso 9: Cálculo de componentes básicos
 - DIO 9.1: Cálculo de componentes básicos exitosa
 - DIO 9.2: Procesamiento incompleto del cálculo de componentes básicos
- Caso de Uso 10: Gráficos: Histogramas, Percentil, Ojiva
 - DIO 10.1: Presentación de gráficos exitosa
 - DIO 10.2: Presentación de gráficos no satisfactoria

Plan de pruebas

En esta etapa se califica el nuevo sistema para entrar en paralelo, de modo que se hace necesario hacer la entrega de la documentación con la aceptación del usuario final, de modo que este pueda asumir el control al momento del corte final del antiguo sistema.

Los siguientes son los documentos que deberán ser entregados durante este periodo.

- ✚ Documentación de los programas.
- ✚ Manual de usuario.
- ✚ Manual de operación
- ✚ Documentación de la prueba de aceptación.
- ✚ Plan del paralelo.

Para la entrega de los documentos en esta etapa, se debe cubrir los siguientes requerimientos:

- ✚ Documentación: programas, manual del usuario y manual de operación
- ✚ Producción deben integrarse simultáneamente
- ✚ Etapas para verificar el rendimiento de las pruebas
 - Prueba individual de programas

- Prueba del sistema global
- ✦ Aceptación de la documentación antes descrita, esto es obligatorio en el plan de prueba y representa que esta etapa ha sido completada satisfactoriamente

4.3 Implementación: Arquitectura

Dado la complejidad del algoritmo meta-heurístico Tabú Search, se hace necesario realizar todos los procesos que a este le concierne en la memoria principal del equipo, básicamente esto se debe a la estructura de matrices que maneja el algoritmo de búsqueda, matrices de corto y largo plazo con lo cual analiza la vecindad de una solución y si esta no presenta un óptimo, de acuerdo análisis de su estructura de memoria; se opta por una nueva vecindad, esta serie de operaciones sucesiva nos lleva a trabajar bajo esquemas de rápido acceso y obviamente dependiente de la configuración y características del equipo.

Adicionalmente, tenemos que considerar la metodología de los procesos estocásticos, los cuales son procesos aleatorios definidos por funciones matemáticas y van de acuerdo a las funciones de distribución de las variables de entrada, es decir para los eventos se genera un pool de variables de entrada las cuales nos sirven para analizar un escenario en particular para un instante de tiempo determinado, como hemos deducido en secciones posteriores el análisis de los escenarios en gran medida afecta el nivel de satisfacción de la solución; es decir entre más escenarios se procesen en la simulación la secuencia hallada tendrá un elevado grado de aceptación, pero al contrario en pocos escenarios analizados el tiempo de procesamiento se reducirá notablemente, y la solución encontrada no será eficiente, con lo cual es

preciso generar una gran cantidad de escenarios donde se analice el comportamiento del sistema, es por esto que la mejor opción para llevar a cabo tal cantidad de procesos de manera rápida y eficiente es a través de estructuras dinámicas que cambien de acuerdo al proceso analizado, estructuras que por el hecho; de tener que manejar procesos en forma acelerada y en gran cantidad, se recomienda realizarlo en dispositivos de rápido acceso, cuya limitante sea las características del equipo, es decir con gran capacidad de procesamiento (memorias y procesador) y rápido acceso. De esta forma, dada las características de las metodologías, obtenemos un sistema totalmente portable e independiente de estructura de almacenamiento (se recalca el hecho que lo anterior, esta relacionado a los procesos para obtener la solución óptima, más no tiene ninguna referencia con el almacenamiento de las secuencias o almacenamiento de las condiciones iniciales para una respuesta determinada, lo cual es considerado como mejora del sistema propuesto).

El siguiente recuadro representa el tiempo aproximado que tardaría en implementar el sistema de optimización y simulación propuesto, esto de acuerdo al análisis previamente realizado.

Descripción		ESTIMADOS	
		Horas	Días
1	Preparación		
1.1.	Capacitación a desarrolladores	40	5.00
2	Inicio		
2.1.	Levantamiento de requerimientos	35	4.38
2.2.	Especificación de requerimientos	25	3.13
2.3.	Validación de requerimientos	20	2.50
3	Elaboración		
3.1.	Diseño de alto nivel	15	1.88
3.2.	Diseño de bajo nivel		
3.2.A.	Diseño de objetos (Funciones de Distribución)	10	1.25
3.2.B.	Diseño de interfaz del usuario	20	2.50
3.2.C.	Modelo de Diseño de objetos	15	1.88
3.3.	Revisión del diseño y evaluación	10	1.25
4	Implementación		
4.1.A.	Implementación del sistema		
4.1.A.1.	Implementación: Funciones de Distribución	30	3.75
4.1.A.2.	Implementación: Simulación	30	3.75
4.1.A.3.	Implementación: Optimización	30	3.75
4.1.A.4.	Implementación: Análisis Estadístico	20	2.50
4.1.A.5.	Implementación: Gráficos: Histogramas, Ojivas, Percentil	25	3.13
4.1.A.6.	Integración de Componentes	10	1.25
4.1.B.	Documentación técnica	20	2.50
4.1.C.	Documentación para el usuario	15	1.88
4.1.D.	Pruebas		
4.1.D.1.	Planeación de pruebas	20	2.50
4.1.D.2.	Implementación de pruebas	35	4.38
4.1.D.3.	Pruebas de ejecución	5	0.63
4.2.	Resultados de las pruebas y evaluación	20	2.50
5	Transición: Ejecución en paralelo		
5.A.	Entrega de la primera versión	3	0.38
5.B.	Documentación: actualización manuales	25	3.13
6	Análisis de los Objetivos		
6.1.	Reporte: Conclusiones del proyecto	10	1.25
Total en horas/días		488	61.00

Tabla # 2 IMPLEMENTACIÓN DEL SISTEMA*

4.4 Manual Del Usuario

A continuación describiremos las opciones disponibles, no detallaremos las operaciones internas de las mismas, dado que su funcionamiento fue tratado con anterioridad. Cabe resaltar, la combinación ítem – equipo, no deberá sobrepasar los 20 ítems y 20 equipos, dicha limitante es factible mejorarla, almacenando más semillas para generar las funciones de distribución. Dado, que es una versión académica tiene 250 semillas almacenadas.

• Pantalla de presentación



• Opciones menú Herramientas



• Ingreso Parámetros: ítems, equipos, etc.



Como observamos el programa inicia con una presentación sencilla del objetivo del proyecto:

Pantalla inicial: OPTIMIZACIÓN Y SIMULACIÓN

Con la opción **Herramientas**, seleccionamos las siguientes alternativas:

1. *Parámetros iniciales*
2. *Ingreso Secuencias Producto Máquina*
3. *Ingreso de tiempo Máquina*
4. *Ingreso Secuencias a Comparar*

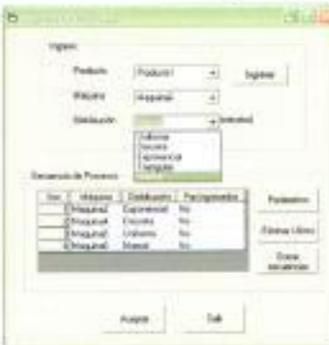
Seleccionando **Ingreso de Parámetros iniciales**, ingresamos:

1. *Cantidad de ítem*
2. *Cantidad de equipos*
3. *Número de escenarios*
4. *Número de iteraciones*

➤ **Ingreso Combinación: *Items - Máquina***



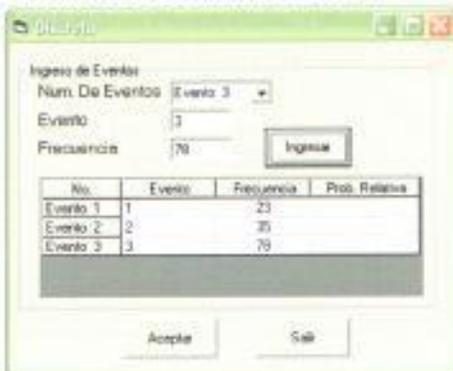
➤ **Ingreso De Distribución por Equipo**



➤ **Parámetros *Probabilidad Exponencial***



➤ **Parámetros *Probabilidad Discreta***



◊ Con esta opción (**Ingreso de procesos**) podemos seleccionar las máquinas que producirán un ítem en particular.

Es claro indicar, que el máximo número de equipos estará definido al inicio, en la opción Cantidad de equipos.

◊ Para definir el comportamiento de producción de un equipo para un ítem en particular, seleccionamos el tipo de distribución (exponencial, discreta, uniforme, triangular y normal) a través de la Opción **Distribución** de Ingreso de Procesos.

◊ La siguiente pantalla nos permite el ingreso del parámetro beta en una función de distribución exponencial. (Opción: **Probabilidad Exponencial**)

◊ En esta opción podemos ingresar los parámetros de la distribución discreta, los cuales son: Número de evento, descripción del evento y frecuencia. (**Probabilidad Discreta**). Dado, que trabajamos con probabilidades es necesario ingresar en la descripción del evento, números y no

• Cálculo Probabilidad Relativa

Ingreso de Eventos

Núm. De Eventos:

Evento:

Frecuencia: Ingresar

No.	Evento	Frecuencia	Prob. Relativa
Evento 1	1	23	16.911
Evento 2	2	35	25.736
Evento 3	3	79	57.352

Se ha realizado el cálculo relativo de la frecuencia de los eventos.

Aceptar

caracteres, la suma de la frecuencia ingresada no es necesariamente el 100%, por que existe el cálculo de la frecuencia relativa, la misma que se obtiene en base de las frecuencia absolutas y cuyo total es el 100%.

• Parámetros Probabilidad Uniforme

Uniforme

Parámetros

$F(x) = 1/(b-a)$ entre (a,b)

a: b:

Aceptar Salir

Esta opción nos permite el ingreso de los parámetros de la distribución uniforme (**Probabilidad Uniforme**) los cuales son:

Límite inferior y Límite Superior

• Parámetros Probabilidad Normal

Normal

Ingreso de Parámetros

$Z1 = \frac{X - \mu}{\sigma / \sqrt{n}}$

$Z2 = \frac{X - \mu}{\sigma / \sqrt{n}}$

Media y varianza

Máximo y mínimo

mu: sigma:

Aceptar Salir

El ingreso de los parámetros de la distribución normal, se lo puede realizar de dos formas: por su media y varianza; y por su máximo y mínimo (**Probabilidad Normal**), el ingreso que da a consideración del usuario.

• Parámetros Probabilidad Triangular

Triangular

Parámetros requeridos

$F(x)1 = \frac{2}{(c-a)} * (b-a) * (x-a)$

$F(x)2 = - \frac{2}{(c-a)} * (c-b) * (x-c)$

a: b: c:

Aceptar Salir

Otra de las distribuciones mayormente utilizadas, es la distribución triangular (**Probabilidad Triangular**), en esta los parámetros a ingresar son: **Límite inferior, medio y superior.**

➤ Ingreso tiempo equipo: *Funcionamiento*

The dialog box is titled 'Ejercicio de Simulacion'. It has a section 'Ingreso de Tiempo' with three dropdown menus: 'Máquina' (set to 'Máquina1'), 'Tiempo' (set to 'Funcionamiento'), and 'Distribución' (set to 'Exponencial'). There are 'Ingresar' and 'Cancelar' buttons. Below is a table with columns 'Tiempo', 'Distribución', and 'Parámetro(s)'. The 'Funcionamiento' row is highlighted. At the bottom are 'Aceptar' and 'Salir' buttons.

This screenshot shows the same dialog box as above, but the 'Distribución' dropdown menu is open, showing a list of options: 'Uniforme', 'Discreta', 'Exponencial', 'Triangular', and 'Normal'. The 'Funcionamiento' row in the table below is still highlighted.

➤ Ingreso tiempo equipo: *Reparación*

The dialog box is titled 'Ejercicio de Simulacion'. In the 'Ingreso de Tiempo' section, 'Máquina' is 'Máquina1', 'Tiempo' is 'Reparación', and 'Distribución' is 'Triangular'. There are 'Ingresar' and 'Cancelar' buttons. The table below shows 'Reparación' with a 'Uniforme' distribution and a checked 'Parámetro(s)' box. At the bottom are 'Aceptar' and 'Salir' buttons.

Posterior al ingreso de los parámetros de las distribuciones, de acuerdo al modelo de las máquinas que producen los ítems, es necesario el ingreso de las distribuciones de los tiempos de funcionamiento y reparación (**Tiempo de Máquina: Funcionamiento**), al igual que en la opción anterior, podemos ingresar diferentes tipos de distribuciones (uniforme, discreta, exponencial, triangular y normal), con sus respectivos parámetros, tal como observamos en la figura adjunta.

Lo importante de toda simulación es que cumpla con la mayoría de las condiciones que se dan en la realidad y entre ellas son el tiempo de reparación de cada equipo, el cual están regidas por diferentes distribuciones probabilísticas y parámetros de acuerdo a su distribución. (**Tiempo de máquinas: Reparación**).

➤ Opciones de Ejecución



➤ Ejecución del algoritmo



➤ Ver Secuencia Óptima

	Escenario1	Escenario2	Escenario3	Escenario4	Escenario5
Producto 1	1	3	0	0	0
Producto 2	1	0	0	0	0
Producto 3	2	4	0	0	0
Producto 4	1	0	0	0	0
Producto 5	1	2	0	0	0

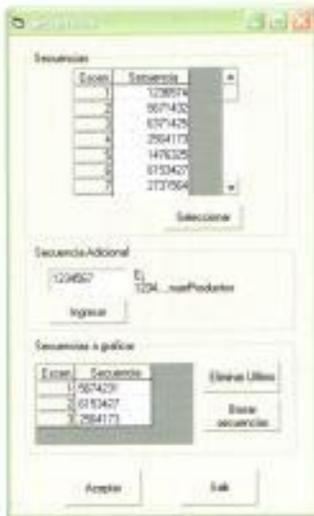
➤ Entre las **Opciones de ejecución** tenemos:

1. **Tabú Search**
2. **Simulación**
3. **Prueba de Bondad de Ajuste**

➤ Seleccionando la opción de Tabú Search, el inicio del algoritmo (posterior al ingreso de todos los parámetros requeridos) ejecuta los procedimientos de simulación para n escenarios y optimización meta-heurística (optimización para cada escenario)

➤ Después de encontrar la secuencia que mejor se comporta en todos los escenarios y a la vez se protege contra el peor de ellos, es necesario mostrar la solución calculada, esto de acuerdo al número de escenarios podrá tomar varios minutos en encontrar dicha solución.

➤ Ingreso de Secuencias



➤ Inicio Proceso de Simulación



➤ Opciones de Visualización



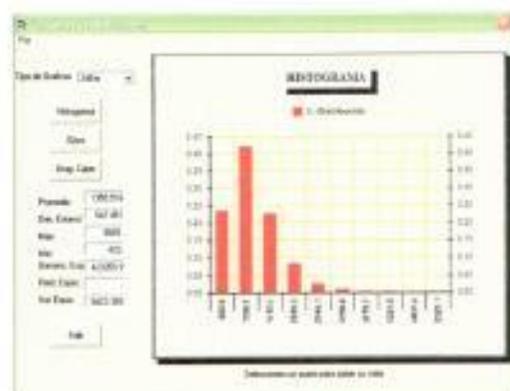
➤ Después de encontrar la solución óptima y con la finalidad de comprobar los resultados, seleccionamos secuencias diferentes a la hallada, posteriormente iniciamos nuevamente el proceso de simulación pero sin la meta-heurística, esto con la finalidad de poder comparar otra secuencias vs. la secuencia óptima.

➤ Seleccionamos la opción Proceso de Simulación, con esto iniciamos la simulación de los diferentes escenarios. (**Procesando Simulación**), en esta opción no se ejecuta la meta-heurística, solo se realiza la simulación.

➤ Por último, tenemos la opción **Ver**, ellas podemos escoger entre dos alternativas, de acuerdo a los requerimientos:

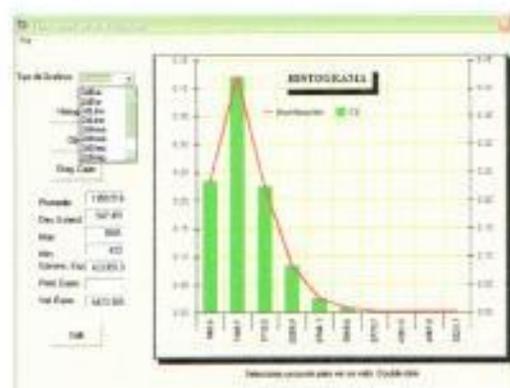
1. **Secuencia Óptima**
2. **Graficar Secuencia.**- Seleccionando esta opción obtenemos el análisis gráfico y estadístico de las diferentes secuencias seleccionadas con anterioridad

➤ Opción Graficar Secuencia



◊ Uno de los primeros gráficos mostrados por la **Opción Graficar Secuencia**, es el gráfico de histograma, acompañado de sus datos estadísticos (promedio, desviación estándar, máximo, mínimo, ganancia esperada, pérdida esperada y valor esperado)

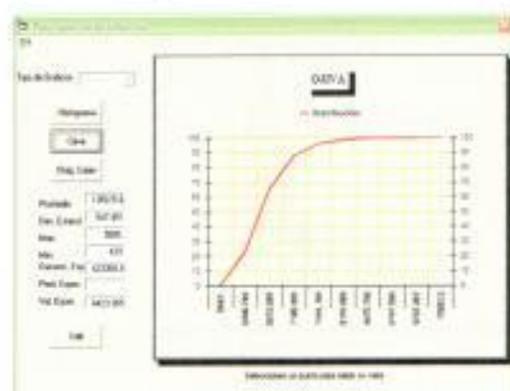
➤ Tipo De Gráfico: Suavizar



◊ Como observamos adicional, al gráfico tradicional de barras en 2D, se encuentra implementado una serie de gráficos de diferente tipo, como:

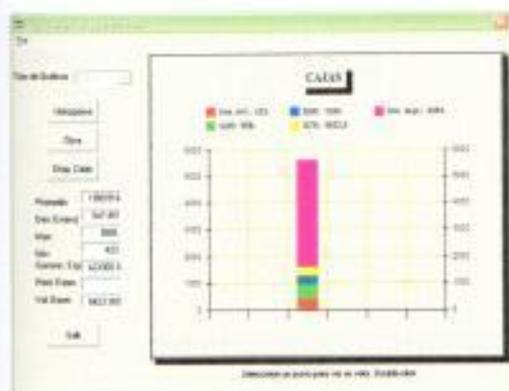
3dBar, 2dBar, Suavizar, etc.

➤ Gráfico: Ojiva



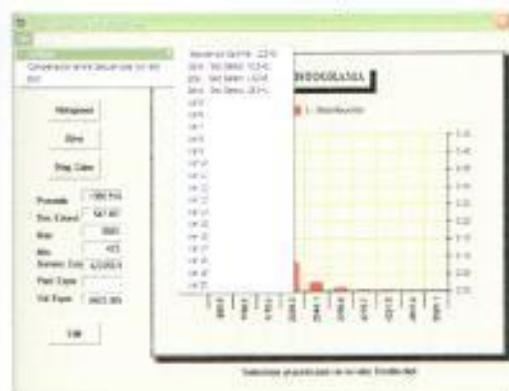
◊ Entre los gráficos de utilidad para el análisis de los diversos escenarios propuesto y encontrado (solución óptima), en el gráfico de la **Ojiva**, al igual que en todas presenta los datos estadísticos, de esta secuencia

➤ Gráfico: Diagramas de Cajas



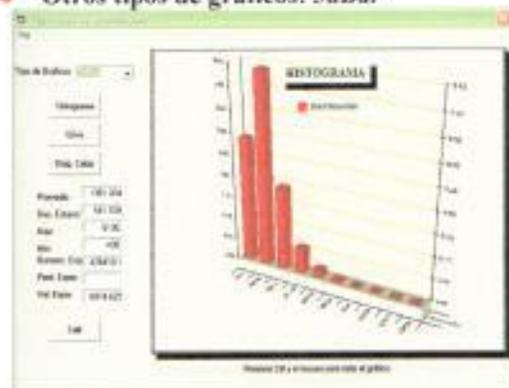
El **Diagrama de Cajas** es otra método por el cual podemos determinar la mejor secuencia, por lo cual, también se lo ha propuesto como gráfico base, en el estudio de la simulación de las secuencias. Estos son los gráficos base (histograma, ojiva y diagrama de cajas)

➤ Selección de Secuencias a graficar



Podemos seleccionar las diversas combinaciones, a través de las opciones del menú, lo cual nos facilita su apreciación gráfica, además; en forma independiente graficar su histograma, ojiva y diagrama de cajas. (**Opción Graficar Secuencia**)

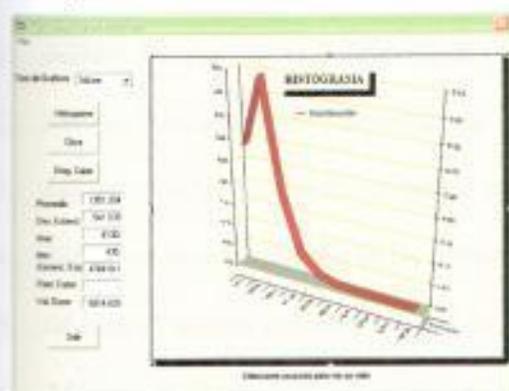
➤ Otros tipos de gráficos: 3dBar



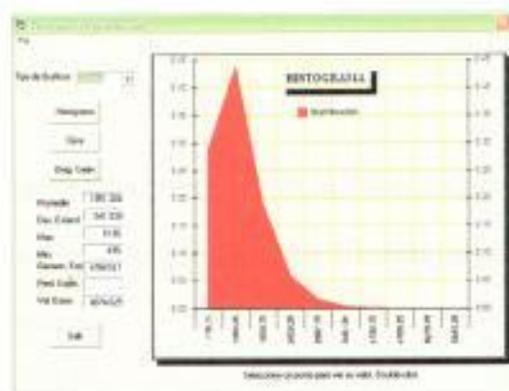
A continuación detallaremos todos los posibles gráficos que están a disposición en este programa:

- 3dBar
- 2dBar
- 3dLine
- 2dLine
- 3dÁrea
- 2dÁrea, etc.

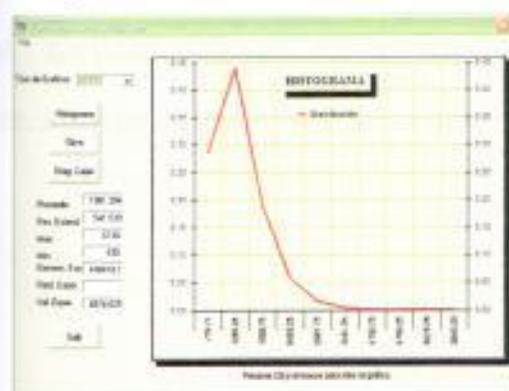
➤ Tipos De Gráficos: 3dLine



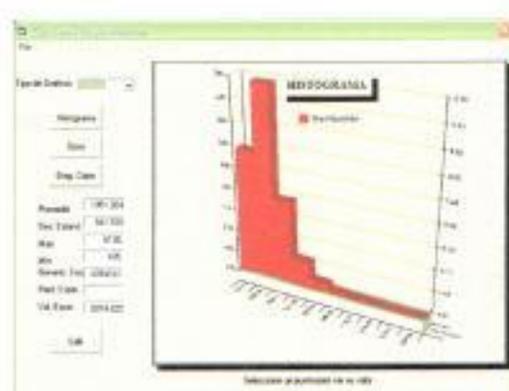
➤ Tipo De Gráfico: 2dÁrea



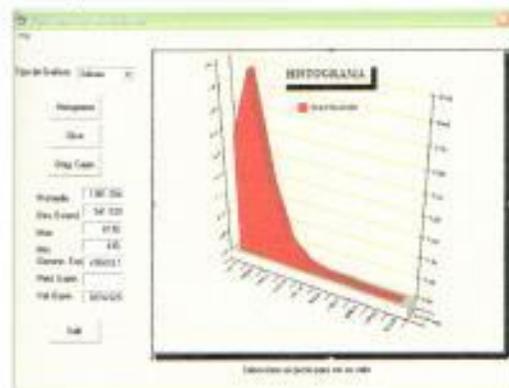
➤ Tipo De Gráfico: 2dLine



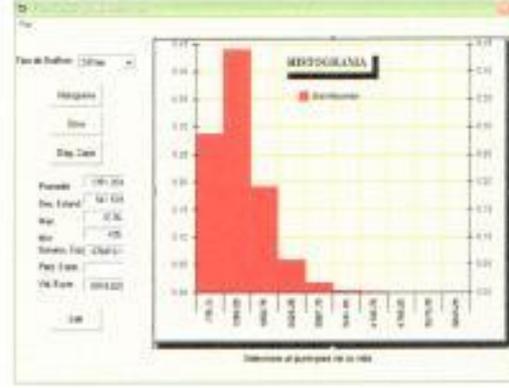
➤ Tipo De Gráfico: 3dStep



➤ Tipo De Gráfico: 3dÁrea



➤ Tipo De Gráfico: 2dStep



◇ Entre las opciones disponibles para graficar los resultados tenemos:

1. **3dBar**.- Histograma en 3 dimensiones
2. **2dBar**.- Histograma en 2 dimensiones
3. **3dLine**.- Une promedio de las barras de los histogramas, y lo presenta en 3 dimensiones
4. **2dLine**.- Une el promedio de las barras del histograma, y lo presenta en 2 dimensiones
5. **3dArea**.- Presenta en forma de área en 3 dimensiones
6. **2dArea**.- Presenta en forma de área en 2 dimensiones.
7. **3dStep**.-Presenta en pasos en 3 dimensiones
8. **2dStep**.- Presenta en pasos en 2 dimensiones
9. **3dCombinación**.- Combina los gráficos 3dBar y 3dLine
10. **Suavizar**.- Combina los gráficos 2dBar y 2dline

CAPÍTULO V

ESCENARIO

V ESCENARIO

Se dice que un sistema ha alcanzado las condiciones estables o ha llegado a un **"régimen permanente"** cuando las observaciones sucesivas del desempeño del sistema son estadísticamente independientes. Por lo cual, un sistema cuyo comportamiento no satisface las condiciones del régimen permanente esta en un **"estado transitorio"**. Los modelos pueden exhibir propiedades transitorias por una de dos razones:

- 1.- *"Si las condiciones usadas para inicializar el sistema no son típicas de las condiciones operativas, pero se espera que eventualmente el sistema exhiba las condiciones del estado estable, entonces hay un periodo transitorio hasta que los efectos de las condiciones iniciales desaparezcan o se vuelvan insignificantes. Un buen diseño experimental debe asegurar que los resultados durante tal fase de transición sean insignificantes o no sean tenidos en cuenta en el análisis."*
- 2.- *"Puede ocurrir un fenómeno transitorio en la situación que se esta simulando."*

Por lo anterior, el investigador debe ser capaz no solo de diseñar y ejecutar un programa de simulación, sino también de interpretar los resultados. Un estudio de simulación se plantea como una serie de corridas orientadas a comparar alternativas que debe analizarse en forma colectiva, determinando la solución que mejor se comporte en todos los escenarios e inclusive en el peor de ellos.

En este capítulo, haremos una breve introducción sobre las técnicas de generación de variables aleatorias, es decir nuestro entorno esta básicamente regido por dos técnicas: Transformación Inversa y Caracterización, finalmente se demostrará para cada método su **FUNCIÓN DE DISTRIBUCIÓN ACUMULADA**. Para, lo cual trabajaremos con las estructuras estadísticas más comúnmente usadas.

De esta forma queda definido el entorno sobre el cual se ejecutará la simulación (función de distribución acumulada, conjunto de escenarios) y la optimización combinatorial (Meta-heurística “Búsqueda Tabú”, selección de la mejor secuencia de acuerdo al escenario analizado)

5.1 Definición del escenario.

Como observamos en el siguiente extracto de programa, se detallan funciones matemáticas las cuales describen las distribuciones acumuladas inversas de: exponencial, triangular, discreta, uniforme y normal.

```
Public Sub DistFuncion(ByVal beta As Single, ByVal lminF As Single, ByVal lmedF As Single, ByVal lmaxF As Single,
ByVal i As Long, ByVal media As Single, ByVal desv As Single, ByVal cont As Integer)
Dim liminf As Single
Dim limsup As Single
Dim lmfun As Single
Select Case haoGen
Case 1
    DisProb(i, cont) = CLng(beta * Log(1 / (1 - ale(i))))
Case 2
    lmfun = (lmedF - lminF) / (lmaxF - lminF)
    If ale(i) <= lmfun Then
        DisProb(i, cont) = CLng(Sqr((lmaxF - lminF) * (lmedF - lminF) * ale(i) + lminF))
    Else
        DisProb(i, cont) = CLng(lmaxF - Sqr(((lmaxF - lmedF) ^ 2) - ((lmaxF - lminF) * (lmaxF - lmedF)) * (ale(i) - ((lmedF - lminF) / (lmaxF - lminF)))))
    End If
Case 3
    liminf = 0
    limsup = 0
    temp = 4
    For j = 1 To Val(variables(cont, 4))
        temp = temp + 3
        If j <= Val(variables(cont, 4)) Then limsup = limsup + Val(variables(cont, temp)) / 100
        If ale(i) > liminf And ale(i) <= limsup Then
            DisProb(i, cont) = CLng(variables(cont, temp - 2))
        End If
        If ale(i) = 0 Then DisProb(i, cont) = CLng(variables(cont, 5))
        liminf = limsup
    Next j
Case 4
    DisProb(i, cont) = CLng(ale(i) * (lmaxF - lminF) + lminF)
Case 5
    If ale(i) = 0 Then ale(i) = 1 / (10 ^ Len(CStr(nale - 1)))
    DisProb(i, cont) = CLng(media + desv * Sqr(-2 * Log(ale(i))) * Cos(2 * 3.141593 * AleN(i)))
End Select
End Sub
```

Existe una variedad de procedimientos para generar variables aleatorias y cada procedimiento se aplica solo a un subconjunto de distribuciones y para una distribución en particular un método puede ser más eficiente que otro. A continuación presentaremos un diagrama de flujo, que nos ayuda a decidir la mejor técnica a utilizar

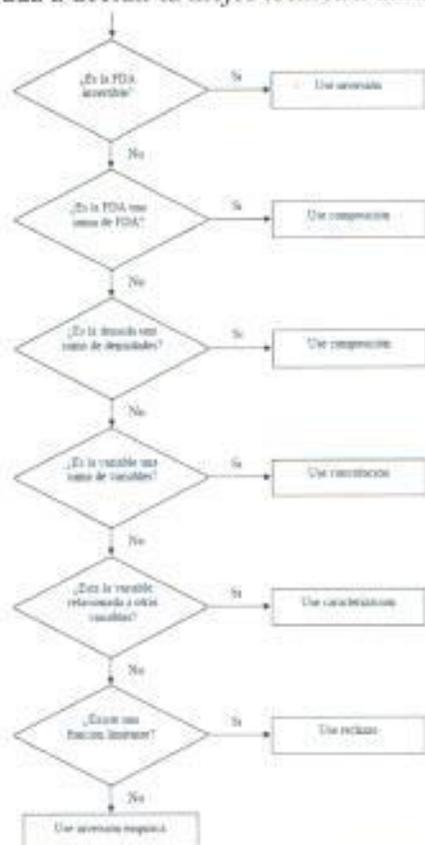


Fig. 5.1 TÉCNICAS DE GENERACIÓN DE VARIABLES ALEATORIAS

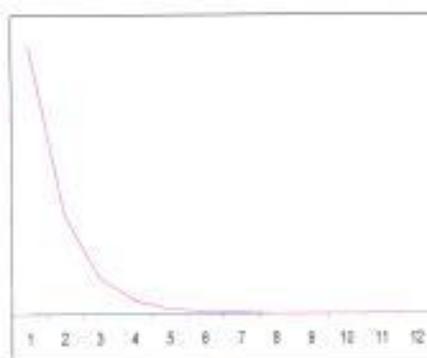
Si la variable aleatoria X tiene una FDA $F(x)$, entonces la variable $u = F(x)$ esta distribuida uniformemente entre 0 y 1. Por lo tanto, X se puede obtener generando números uniformes y calculando $x = F^{-1}(u)$.

Sea $u = g(x)$ tal que $x = g^{-1}(u)$: $F_U(u) = P(U \leq u) = P(X \leq g^{-1}(u)) = F_X(g^{-1}(u))$

Seleccionemos $g(\cdot)$ de forma que $g(x) = F_X(x)$, o $u = F_X(x)$, y que u sea una variable aleatoria entre 0 y 1 con distribución dada por $F_U(u) = F_X(g^{-1}(u)) = F_X(F_X^{-1}(u)) = u$

Y

$$f(u) = \frac{dF}{du} = 1$$

Fig. 5.2 $f(x) = \lambda e^{-\lambda x}$

Sea la función exponencial, de acuerdo al gráfico 4.2, donde su función de distribución acumulada es $F(x) = 1 - e^{-\lambda x}$ y además $x = F^{-1}(u)$ es igual a $x = -\frac{1}{\lambda} \ln(1-u)$. Donde u es una variable uniformemente distribuida entre 0 y 1. Por lo cual el siguiente paso para generar variables aleatorias exponenciales, sería generar u y después calcular la distribución a través de la función, anteriormente descritas.

Como observamos el cálculo de u , deberá ser aleatorio y uniformemente distribuido, por lo cual u tomará la siguiente forma, de acuerdo al método congruencial:

Sea $C_{i+1} = B * C_i + \mu$, donde C_0 es la semilla del algoritmo y siendo su exactitud o cantidad de números aleatorios determinada por el factor P , y es el cual; divide el resultado C_{i+1} , obteniendo de esta forma el número aleatorio y uniformemente distribuido. Si P es de precisión 2 entonces la cantidad de números aleatorios es de 100, si la precisión es tres, la cantidad de números aleatorios es 1000, y así sucesivamente. De acuerdo como consta en el siguiente extracto del algoritmo:

```
Public Sub GenerarAleatorio(opcion As Integer)
  C1=1-B*C1+mu
  nAle = numEscenarios
  m.variables(0, 1) = numProductos * numMaquinas + 2 * numMaquinas
  m.variables(0, 2) = numEscenarios
  If opcion > 1 Then
    ReDim EstDistOut(1 To nEscogidos + 1, 1 To 20)
    ReDim DisProbOut(1 To numEscenarios, 1 To nEscogidos + 1)
    ReDim SecSelecionada(1 To numProductos)
  End If
```

```

Call secuencia
escala = Val(m.variables(0, 3))
iter = 1
nrc = Len(CStr(nale - 1))
While iter <= nrc
  cont = 0
  While (cont < Val(m.variables(0, 1)))
    cont = cont + 1
  If iter = 1 Then
    Ind = 1
    indn = 1
    While (m.SecInd(Ind, 4) <> 0)
      Randomize
      Ind = Int((Rnd * 250) + 1)
    Wend
     $\beta(cont) = m.SecInd(Ind, 1)$ 
     $ci(1, cont) = m.SecInd(Ind, 2)$ 
     $ai(cont) = m.SecInd(Ind, 3)$ 
     $m.SecInd(Ind, 4) = cont$ 
    While (m.SecInd(indn, 4) <> 0)
      Randomize
      indn = Int((Rnd * 250) + 1)
    Wend
     $f_n(cont) = m.SecInd(indn, 1)$ 
     $cin(1, cont) = m.SecInd(indn, 2)$ 
     $an(cont) = m.SecInd(indn, 3)$ 
     $m.SecInd(indn, 4) = cont$ 
  End If
  If nale <= 9 Then  $\beta(cont) = 11$ 
   $ci(iter + 1, cont) = \beta(cont) * ci(iter, cont) + \mu(cont)$ 
   $cin(iter + 1, cont) = f_n(cont) * cin(iter, cont) + \mu_n(cont)$ 
   $ale(iter) = (ci(iter + 1, cont) / prec) - Fix(ci(iter + 1, cont) / prec)$ 
   $AleN(iter) = (cin(iter + 1, cont) / prec) - Fix(cin(iter + 1, cont) / prec)$ 
  beta = 0
  lminF = 0
  lmedF = 0
  lmaxF = 0
  media = 0
  desv = 0
  Select Case m.variables(cont, 2)
    Case "Exponencial"
      banGen = 1
      beta = Val(m.variables(cont, 4))
    Case "Triangular"
      banGen = 2
      lminF = Val(m.variables(cont, 4))
      lmedF = Val(m.variables(cont, 5))
      lmaxF = Val(m.variables(cont, 6))
    Case "Discreta"
      banGen = 3
    Case "Uniforme"
      banGen = 4
      lminF = Val(m.variables(cont, 4))
      lmaxF = Val(m.variables(cont, 5))
    Case "Normal"
      banGen = 5
      media = Val(m.variables(cont, 6))
      desv = Val(m.variables(cont, 7))
      If media = 0 Then
         $media = (Val(m.variables(cont, 4)) + Val(m.variables(cont, 5))) / 2$ 
         $desv = (Val(m.variables(cont, 4)) - Val(m.variables(cont, 5))) / 6$ 
         $m.variables(cont, 6) = media$ 
         $m.variables(cont, 7) = desv$ 
      End If
  End Select
  Call DistFuncion(beta, lminF, lmedF, lmaxF, iter, media, desv, cont)

```

```

banGen = 0
If ci(iter + 1, cont) > 10000 Then
    resultado = ci(iter + 1, cont) Mod 10
    If resultado = 0 Then ci(iter + 1, cont) = ci(iter + 1, cont) + 1
    ci(iter + 1, cont) = Val(Mid(CStr(ci(iter + 1, cont) / prec), InStr(1, CStr(ci(iter + 1, cont) / prec), ".", 1) + 1, sect))
    If resultado = 0 Then ci(iter + 1, cont) = ci(iter + 1, cont) - 1
End If
If cin(iter + 1, cont) > 10000 Then
    resultado = cin(iter + 1, cont) Mod 10
    If resultado = 0 Then cin(iter + 1, cont) = cin(iter + 1, cont) + 1
    cin(iter + 1, cont) = Val(Mid(CStr(cin(iter + 1, cont) / prec), InStr(1, CStr(cin(iter + 1, cont) / prec), ".", 1) + 1, sect))
    If resultado = 0 Then cin(iter + 1, cont) = cin(iter + 1, cont) - 1
End If
Wend
*nota: aquí debemos analizar un escenario y encontrar la solución óptima para este escenario,
* a continuación debemos guardar esta solución
* repetir este procedimiento n iteraciones(n=iter) y obtener la combinación que se repite mas veces.
* entrar solución mediante tabú search
Select Case opción
    Case 1 'encuentra solución óptima
        ReDim SolOptima(1 To numProductos)
        SolOptima = TabuSearch(iter)
        For l = 1 To numProductos
            solución(iter, l) = SolOptima(l)
        Next l
        Erase SolOptima
        frmSolucion.pb_solucion.Value = iter
    Case 2 'grafica solución óptima
        DisProbOut(iter, 1) = costoTiempo(SolOptima, iter)
        EstDistOut(1, 1) = Val(EstDistOut(1, 1)) + Val(Val(DisProbOut(iter, 1)) / numEscenarios)
    Case 3 'grafica n soluciones y las superpone a la solución óptima
        For i = 2 To nEscogidos + 1
            For n = 1 To numProductos
                SecSeleccionada(n) = seGraficar(i - 1, n)
            Next n
            DisProbOut(iter, i) = costoTiempo(SecSeleccionada, iter)
            EstDistOut(i, 1) = Val(EstDistOut(i, 1)) + Val(Val(DisProbOut(iter, i)) / numEscenarios)
        Next i
        frmInicioSimulacion.pb_solucion.Value = iter
End Select
iter = iter + 1
Wend
Select Case opción
    Case 1 'encuentra solución óptima
        ReDim SolOptima(1 To numProductos)
        SolOptima = selectSecuenciaModa(solución)
    Case 2 'grafica solución óptima
        Call EstadistDescriptiva
End Select
End Sub

```

Cabe indicar y como se explicó anteriormente este método, es factible de aplicar dado que en la generación aleatoria de las series son independiente entre sí, por que sus factores β y μ son diferentes.

Nota: la condición anterior es muy importante por lo cual podemos inferir que si la correlación entre las variables es cero entonces, estamos frente a funciones de entrada independientes entre sí.

Como veremos posteriormente dentro de los cálculos del programa realizaremos esta comprobación, la cual es de vital importancia en la simulación de sistemas (procesos estocásticos: independencia entre las variables de entrada)

Lo anteriormente analizado es referente a la función de probabilidad exponencial, pero adicionalmente se realizará la simulación del sistema con otros tipos de funciones las cuales se detallan a continuación:

✚ Función de Probabilidad Exponencial

$$DisProb(i, cont) = CLng(beta * Log(1 / (1 - ale(i)))) \quad \text{donde } beta = \frac{1}{\lambda}$$

✚ Función de Probabilidad Triangular

```
If ale(i) <= lmfun Then
DisProb(i, cont) = CLng((Sqr((lmaxF - lminF) * (lmedF - lminF) * ale(i)) + lminF))
Else
DisProb(i, cont) = CLng((lmaxF - Sqr(((lmaxF - lmedF) ^ 2) - ((lmaxF - lminF) * (lmaxF - lmedF)) * (ale(i) - ((lmedF - lminF) / (lmaxF - lminF))))))
End If
```

✚ Función de Probabilidad Discreta

$$DisProb(i, cont) = CLng(variables(cont, temp - 2))$$

✚ Función de Probabilidad Uniforme

$$DisProb(i, cont) = CLng(ale(i) * (lmaxF - lminF) + lminF)$$

✚ Función de Probabilidad Normal

$$DisProb(i, cont) = CLng((media + desv * Sqr(-2 * Log(ale(i)))) * Cos(2 * 3.141593 * AleN(i)))$$

En el caso de la distribución normal, donde no se puede obtener su función inversa a partir de la función acumulada, es necesario aplicar otro método. Dada sus características especiales (distribución normal) permite generar variables normales usando algoritmos ajustados para ella, la cual es una técnica llamada **caracterización**, a continuación se realizará la demostración de la distribución normal.

Considere dos distribuciones normales estándar Z_1 y Z_2 graficadas en el plano tal como se muestra en la figura:

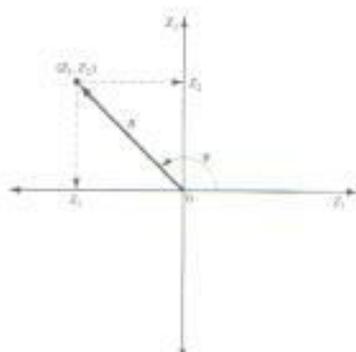


Fig. 5.3 GRAFICO EN PLANO POLAR

Siendo su representación en coordenadas polares la siguiente:

$$\begin{aligned} Z_1 &= B \cos \theta \\ Z_2 &= B \sin \theta \end{aligned}$$

Entonces, ya que $B^2 = Z_1^2 + Z_2^2$ tiene distribución Chi - cuadrado con 2 grados de libertad que es igual una distribución exponencial con media 2:

$$f(x) = \frac{x^{2-2}/2 e^{-x/2}}{2^{2-1} \Gamma(2/2)} = \frac{1}{2} e^{-x/2} \quad x \geq 0$$

Por lo cual el radio B lo podemos encontrar por la inversa usando

$$B^2 = -2 \ln(u) \quad \text{o} \quad B = \sqrt{-2 \ln(u)}$$

Es razonable suponer que el ángulo θ está distribuido uniformemente entre 0 y 2π .

Igualmente se puede considerar que el radio B y el ángulo θ son independientes. Con esto podemos generar dos variables aleatorias normales estándares independientes Z_1 y Z_2 a partir

$$\begin{aligned} Z_1 &= \sqrt{-2 \ln(u_1)} \cos(2\pi u_2) \\ Z_2 &= \sqrt{-2 \ln(u_1)} \sin(2\pi u_2) \end{aligned}$$

Para obtener normales con parámetros de media y varianza, usamos las transformaciones:

$$\begin{aligned} X_1 &= \mu_1 + \sigma_1 Z_1 \\ X_2 &= \mu_2 + \sigma_2 Z_2 \end{aligned}$$

5.2 Planteamiento del problema.

Los modelos de simulación se han incorporado a un sin número de aplicaciones entre los cuales tenemos los procesos de producción y de toma de decisión. En algunos casos es posible construir un modelo físico y probarlo, aunque esto resulta ser muy costoso o poco práctico.

La simulación de los sistemas resulta entonces ser una herramienta económica y de gran utilidad en el estudio del comportamiento de sistemas que involucran variables aleatorias, y a su vez combinada con algoritmos de optimización meta-heurística (como Tabú Search). Dando como resultando una herramienta poderosa, en el estudio de sistemas de procesos de producción.

Lo anterior, da la base para el estudio del siguiente sistema, sea una secuencia de procesos productivos la cual está gobernada por entidades exógenas y endógenas, y estas a su vez están generadas por funciones de distribución probabilística independientes (de acuerdo a lo revisado en la sección anterior). Definido nuestro entorno, procedemos a analizar las entidades que deberán ser optimizadas, primero mediante un muestreo de su comportamiento (con los datos históricos obtendremos su histograma y por consiguiente la forma de la función de distribución), definiendo de esta forma el tipo de distribución (distribución exponencial, triangular, uniforme, discreta y normal), ahora nos concentraremos básicamente en la secuencia del proceso ítems por equipo producido y los tiempos de funcionamiento y mantenimiento de cada uno, que se requieren en el proceso de producción, con el cual obtendremos una secuencia que nos garantice una utilidad óptima en cualquier escenario y a la vez se proteja contra el peor de ellos, como se evidencia nuestra función de penalización será el tiempo de recurso en cada equipo (**la utilidad máxima será inversamente**

proporcional, a la función de penalización "minimizar el tiempo de recurso, empleado por todos los equipos en la producción total", esto será aplicado en el algoritmo meta-heurístico)

Posteriormente, evidenciaremos nuestros resultados a través de un procedimiento manual, para el cual utilizaremos herramientas de análisis estadístico como son Arena y Risk Ease, con el primero modelaremos el sistema en forma gráfica, y después el análisis estadístico (valor esperado, desviación estándar, ganancia esperada, pérdida esperada, entre otros) de los resultados nos apoyaremos en Risk Ease, ambos excelentes herramientas para modelamiento de sistemas y análisis estadístico respectivamente. Cabe indicar que en ninguno de los programas antes mencionados provee módulos que ataquen los problemas de optimización combinatorial (selección de la solución óptima: a través de algoritmos meta-heurísticos, ni tampoco con otros métodos), en general el procedimiento de comprobación es manual y seleccionan diversas combinaciones aleatorias. Adicionalmente el mismo sistema nos proporcionará además de la secuencia óptima, un módulo de gráficos en cual se comparará los gráficos de las funciones acumuladas de las diferentes secuencias vs. el gráfico de la secuencia óptima.

5.3 Representación Meta-Heurística de la solución.

Los requerimientos en la representación de un problema de búsqueda (Tabú Search)

son:

- 1.- *"Estrategia de búsqueda o de control que decide que operación aplicar."*
- 2.- *"Estructura simbólica que contiene conjuntos de soluciones potenciales."*
- 3.- *"Reglas que nos producen entornos depurados con posibles soluciones."*

La representación debe de ser capaz de representar los objetos. Es preferible poder representar también subconjuntos de objetos o posibles potenciales, ya que nos permite eliminar grandes espacios de búsqueda.

El uso de heurísticas impone requerimientos adicionales en el formato de representación. La representación debe de identificar adecuadamente las posibles soluciones y a la vez facilitar las operaciones de la heurística. En general es deseable incluir dentro de la representación información adicional que nos defina el subproblema o estado que falta por resolver.

La importancia de representar los subproblemas o estados es que muchas veces estos:

- 1.- *"Se repiten y podemos aprovechar los resultados obtenidos."*
- 2.- *"Podemos seleccionar que opción es mejor si tienen el mismo espacio solución y una de ellas es mejor que la otra."*

Si la solución puede especificarse como un camino o un estado, la representación basada en estados es la más adecuada, por otro lado; si la solución es más conveniente representarla como un árbol, entonces una representación de reducción de problemas o grafo es más adecuada.

Dado las técnicas para resolver el problema (simulación "procesos estocásticos" y optimización combinatorial "meta-heurística: Búsqueda Tabú"), se hace necesario la

representación de la solución en caminos o estados. Estados que nos indican que subconjunto o posible solución hacemos referencia, esto nos ayudará a eliminar espacios o vecindades de soluciones (se elimina las grandes búsquedas).

Sea la combinación de 7 ítem, los cuales son producidos por 7 equipos, estos tienen diferente función de distribución para la combinación equipo-máquina, por lo cual analizaremos solo un escenario, pero entiéndase que cada escenario de acuerdo al proceso estocásticos tiene su propia secuencia óptima, en otras palabras en cada escenario se deberá ejecutar el método meta-heurístico y encontrar una solución óptima, al final debemos obtener una secuencia que más se repita en relación a las demás, es decir en el escenario tendrá una secuencia A, en el escenario 2 su secuencia óptima es la B, en el escenario 3, será la secuencia B, en el 4to. escenario será la secuencia C, y así sucesivamente, por lo cual nuestra secuencia óptima será la que más veces se repita, para confirmar esta conclusión la función acumulada (ojiva) de esta secuencia deberá ser la mejor en relación al resto (la solución es la óptima, y la que mejor se protege contra el peor de los escenarios), es importante recalcar que en cada escenario se ejecuta el algoritmo de búsqueda tabú, con lo cual obtenemos soluciones o secuencias diferentes para cada iteración, la siguiente es la representación de los estados de acuerdo algoritmo meta-heurístico: TABÚ SEARCH, aquí detallamos las matrices de corto plazo y la lista tabú, de acuerdo a las condiciones anteriormente detalladas.

The image shows a software window titled "Parámetros Iniciales" with a standard Windows-style title bar. Inside the window, there is a section labeled "Parámetros" containing four input fields, each with a numerical value: "Cantidad de Productos" (7), "Cantidad de Máquinas" (7), "Número de Escenarios" (1000), and "Número Iteraciones" (9). Below these fields are two buttons: "Aceptar" and "Salir".

Fig. 5.4 INGRESO DE CONDICIONES

Este ejercicio se ejecuta bajo las siguientes condiciones iniciales, ítems: 7, equipos: 7, número de escenarios: 1000 y número de iteraciones: 5.

Como podemos apreciar el número de escenarios nos indica la precisión del número aleatorio, es decir cuantos escenarios se realizarán en nuestra simulación, adicionalmente tenemos el número de iteraciones, esta es una condición adicional a la función penalización (minimizar el tiempo de recursos para la elaboración de todos los productos), con la cual podemos finalizar el algoritmo tabú search, es decir existe dos condiciones de para (aunque en definitiva solo es una):

- Número de iteraciones,
- Función de penalización: minimizar el tiempo de recursos para elaboración de ítems, de acuerdo a esta condiciones podemos encontrar la mejor combinación equipos – productos.

A continuación observamos un estado de la matriz a corto plazo, lista tabú, y valor de penalización o tiempo que se demora en un escenario particular la producción de la secuencia de ítems antes señaladas.

Solución actual	Estructura tabú (Recurso)							Primeros 5 candidatos			
	1	2	3	4	5	6	7	Valor			
1								1,4	3	3	T
3	1			3				2,4	-1	-0	
6	2							3,7	-3	-3	*
2	3	3				2		1,6	-2	-5	
7	4	1	5				1	6,5	-4	-0	
5	5		4	4							
4	6			1	2						Valor Penalizado
1	7	2			3						

Fig. 5.5 ESTRUCTURA TABÚ

5.4 Aplicación de la solución.

Como observamos el procedimiento busca encontrar una solución, tal que sea la mejor en todos los posibles escenarios y adicionalmente se proteja contra el peor de ellos por lo

cual se convierte en la solución óptima, en posteriores análisis se confirmará esta conclusión a través de la función acumulada de diferentes combinaciones, por el momento nos concentraremos en la solución óptima con las condiciones iniciales anteriormente señaladas.

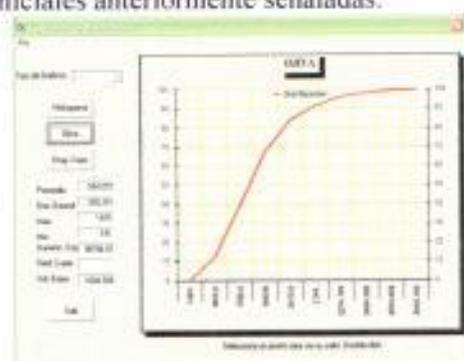
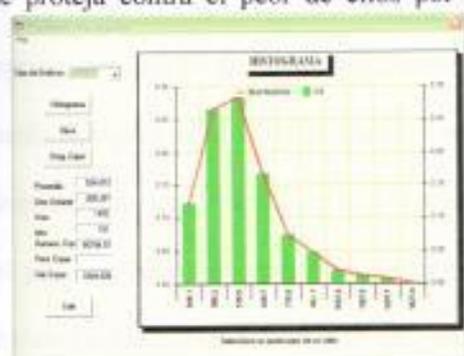


Fig. 5.6 SOLUCIÓN ÓPTIMA.

La combinación óptima reflejará los parámetros de promedio, desviación estándar, máximo, mínimos, generación esperada, pérdida esperada y valor esperado, con estos datos podemos comparar no solo gráficamente la solución sino analíticamente y confirmar que la mejor es aquella cuya ojiva señala una mejor ganancia (menor tiempo de ejecución).

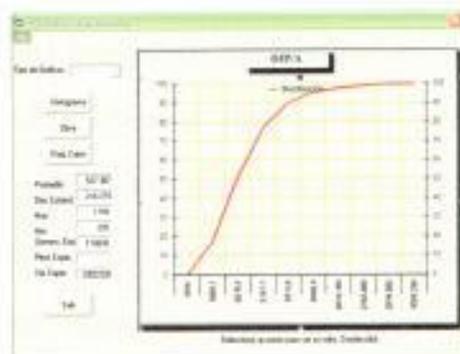


Fig. 5.7 SECUENCIA ALTERNA.

No así las demás cuya ganancia es mayor (mayor tiempo de ejecución), la solución actual tiene una ganancia esperada de 95794.57 unidades de tiempo, seleccionan arbitrariamente una secuencia alterna tenemos que esta presenta una ganancia de 116830 unidades de tiempo tal como se lo había previsto, con lo cual confirmamos la secuencia óptima señalada por la meta-heurística: TABÚ SEARCH.

5.5 Sistema de Planeación Avanzada.

Las compañías han crecido creando islas de información, es decir varios sistemas que operan o manejan diferentes segmentos del negocio. Algunas veces estos sistemas diferentes están integrados entre sí y algunas veces no. La mayoría de las compañías fracasan al implementar los sistemas de ERP porque esperan beneficios financieros diferentes a los que el paquete ofrece propiamente, en otras ocasiones se van con tendencias de la industria y tratan de implementar sistemas que aunque parecen muy innovadores no cumplen con sus necesidades.

Desde inicios de los años noventa las empresas comenzaron la implantación de sistemas ERP (enterprise resource planning), o conocidos como sistemas integrados. Por lo que, los noventa se caracterizó en la implantación de estos sistemas, que vinieron a resolver los problemas transaccionales y la integración de la compañía, posteriormente cambió la tendencia hacia la implantación de objetivos complementarios tales como aplicaciones de la cadena de suministros (APS: advanced planning and scheduling). Los principales proveedores de ERP (SAP o Peoplesoft) estuvieron implementando masivamente sistemas ERP para resolver el problema del Y2K, lo cual los dejó rezagado. Sin embargo, hoy en día, los proveedores de ERP han recuperado terreno en la carrera por el liderazgo de las funcionalidades complementarias. El ERP extendido, como es conocido por algunos, ha

llevado a los principales proveedores de ERP a invertir en el desarrollo de funcionalidades tales como CRM y e-procurement. La incursión en este mercado extendido ha generado el dilema de implantar CRM, e-procurement o APS vs. implantar dichas funcionalidades de sus sistemas ERP. Ericsson España decidió implantar la aplicación e-procurement de SAP para mantener una plataforma única de ERP-B2B. Sin embargo, analizando los datos de participación de mercado se puede observar que las empresas parecen estar prefiriendo mantener una plataforma única con el proveedor original del ERP para cubrir las diferentes funciones del negocio. SAP y Peoplesoft se han colocado rápidamente en el segundo y tercer lugar, respectivamente, de participación en el mercado de CRM detrás de Siebel. Igual, SAP y Oracle están muy cerca de Ariba en el mercado de e-procurement[*].

El mercado está favoreciendo el concepto de plataforma única, lo cual podría llevar a una carrera de pocos proveedores (SAP, IBM, Oracle y Microsoft) donde las alianzas marcarán la pauta. Pocos proveedores pueden garantizar economías de escala y más estabilidad, lo cual implica mejores precios para las empresas donde se implementan las soluciones.

[*] Todos los datos de participación de mercado utilizados en este artículo son de finales del año 2004.
Profesor Oswaldo Lorenzá Operaciones y Tecnología, Madrid.

CAPÍTULO VI

ANÁLISIS DE LA SOLUCIÓN

VI ANÁLISIS DE SOLUCIÓN

Como hemos observado a lo largo del desarrollo de esta tesis, existe métodos meta-heurísticos, los cuales distan entre sí en la forma de resolver un problema en particular, es así; como los algoritmos genéticos [31], [65], meméticos [61], [60] y los de estimación de distribuciones [51] [47] emplean imprescindiblemente procedimientos aleatorios, mientras que, los de búsqueda o de re-encadenamiento de caminos (*Path Relinking*) [44], [54] emplean procedimientos metódicos que nos permite *“recorrer el espacio de soluciones del problema transformando de forma iterativa soluciones de partida...”*.

En general hemos comprobado como la combinación de dos metodologías, (*Simulación*: “procesos estocásticos”, *Optimización Combinatorial*: meta-heurística “Tabú Search”), proporcionan un marco referencial en la solución de nuevos problemas, en los cuales el analista no solo necesita conocer una solución óptima, para un escenario en particular, por ejemplo el más probable, si no una solución que mejor se comporte en todos los escenarios y adicionalmente se proteja contra el peor de ellos. Es decir una secuencia de ítems para el comportamiento de su producción (funciones de distribución probabilística), que no solo abarque el escenario más probable, si no que a través de función acumulada para los diferentes escenarios analizados, sea aquella que este presente en mayor número de veces, originado así una solución, la cual es la que mejor se comparte independiente del escenario que se presente.

La Optimización Combinatorial, analizada en la presente tesis no solo es la única temática (optimización en los procesos productivos) donde se utiliza, actualmente existen otros campos donde se aplicado, naturalmente combinando *Optimización Combinatorial* y *Optimización Robusta*, entre ellas tenemos:

- Planeación y optimización de la producción [63],
- Sistemas logísticos de gran escala (inventarios) [64],
- Ubicación de instalaciones [40],
- Ordenamiento de máquinas en paralelo con perturbaciones estocásticas [46],
- Ingeniería química [10],
- Reubicación de géneros de baja fluctuación en su desarrollo demográfico [29], etc.

En gran parte de los procedimientos heurístico tiene parámetros dentro de la metodología que deben ser calibrados para su correcta ejecución en el procesamiento de los datos, en nuestro caso los parámetros calibrados son los parámetros de entrada los cuales deberán ser en extremo cuidadosamente generados a partir de los datos históricos o en su defecto del muestreo de los eventos en un lapso de tiempo, en este problema en particular como unidad de tiempo será tomado el minuto aunque esto es relevante para los propósitos antes mencionados, es decir se podrá cambiar la unidad de tiempo siempre y cuando todas las variables de entrada soporten dicho cambio

Por último, la única cuestión relevante es conocer si el beneficio en el rendimiento, proporcionado por la inclusión de tales herramientas (Simulación y Optimización), compensa al esfuerzo de su implementación y al incremento de la complejidad del código resultante, lo cual se ira analizando a lo largo de este capítulo.

6.1 Análisis Estadísticos de los Resultados.

El objetivo de una simulación es efectuar cambios al modelo con el objeto de evaluar su efecto sobre las variables de salida que miden el desempeño de un sistema simulado. En los modelos de simulación estocástica estas guías son variables aleatorias por lo cual nos interesó conocer las características de distribución de cada uno de ellos (media, min., máx.,

etc.), es decir desde su generación (a través del método congruencial), hasta la forma de su distribución de acuerdo a sus histogramas y la representación de la misma a través de las funciones matemáticas antes deducidas.

Asiendo una breve reseña del procedimiento, antes de continuar con el análisis, reiteramos que el objetivo principal es obtener una secuencia óptima de ítems producidos de acuerdo a la distribución de sus funciones de densidad, no sin antes considerar que la secuencia de ítems estará definida por el resultado que nos provee el algoritmo meta-heurístico: Tabú Search, es decir que de acuerdo a las variable de entrada: ítems producidos por sus respectivos equipos, y además considerando la distribución del tiempo de funcionamiento y reparación de cada equipo (procesos estocásticos), se obtiene una secuencia óptima para cada escenario, entiéndase escenario como el entorno que genera el procedimiento pseudos aleatorio, de acuerdo a las funciones de entrada y la forma de la distribución en el tiempo, en otras palabras cada unidad de tiempo ha sido analizada por el algoritmo meta-heurístico Tabú Search, y defino su secuencia optima, paso siguiente se almaceno todas las secuencias "óptimas" generadas en los diferentes escenarios y se extrajo aquella que se repite más veces, con lo cual no aseguramos que es la solución que mejor se comporta en el tiempo, por último analizamos la secuencia que más veces se repite y la cual fue generada en los n escenario por el algoritmo de búsqueda, y la comparamos contra cualquier secuencia seleccionada en forma arbitraria, y como vemos a continuación, se compara los gráficos de ojiva de la secuencias optima. Como observamos la integración de ambas metodología es la clave para resolver este tipo de problemas es decir la lista de secuencias optimas arrojadas por el algoritmo de búsqueda analizada en cada instante de tiempo de acuerdo con los procesos estocásticos y su posterior simulación y análisis de la secuencia más repetidas vs. todas aquellas seleccionadas arbitrariamente.

PARÁMETROS INICIALES

CANT. DE PRODUCTO:	7
CANT. DE MÁQUINAS	6
NUM. DE ESCENARIOS:	30,000
NUM. DE ITER:	7

DISTRIBUCIONES DE PROBABILIDAD: EQUIPOS vs. ÍTEMS PRODUCIDO

EQUIPO ÍTEMS	1	2	3	4	5	6
1	-	U(34.6; 78.9)	E(89.5)	**N(46.8; 3)	-	T(3.67; 34.7; 36.4)
2	-	*N(78.8; 56.9)	D(40.88; 60.57; 12.89)	-	E(45.9)	-
3	E(47.1)	**N(70.9; 4.9)	-	T(89.7; 95.4; 100.2)	-	U(49.6; 88.1)
4	U(70.3; 100.5)	-	E(23.6)	-	*N(99.3; 78.4)	E(67.8)
5	-	D(20.56; 44.22; 15.89; 4.67)	T(34.9; 35.4; 90.3)	**N(80.3; 5.7)	E(19.7)	T(20.1; 29.3; 31.4)
6	*N(71.4; 33.9)	-	U(31.8; 56.1)	-	T(20.1; 29.3; 31.4)	D(100.7; 345.2; 88.23)
7	D(89.23; 10.42; 3.88)	E(77.1)	*N(78.2; 10.2)	U(22.8; 25.6)	T(30.8; 31.3; 99.8)	**N(39.7; 1.3)

TIEMPOS

EQUIPO	FUNCIONAMIENTO	REPARACIÓN
1	U(45.7; 67.9)	T(26.5; 45.1; 55.5)
2	E(59.8)	*N(88.1; 44.7)
3	**N(12.4; 3.1)	E(34.5)
4	D(112.3; 5.4; 76.9)	E(21.45)
5	U(55.5; 123.6)	U(59.8; 84.3)
6	E(3.9)	**N(12.56; 2.3)

NOMENCLATURA

FUNCIÓN

PARÁMETROS

U(a; b)	Uniforme	a: Limite inferior	b: Limite superior
E(a)	Exponencial	a: Factor Beta	
*N(máx.; min.)	Normal	máx.: Limite superior	min.: Limite inferior
**N(a; b)		a: media	b: desviación
T(a; b; c)	Triangular	a: Limite inferior	b: medio c: Limite superior

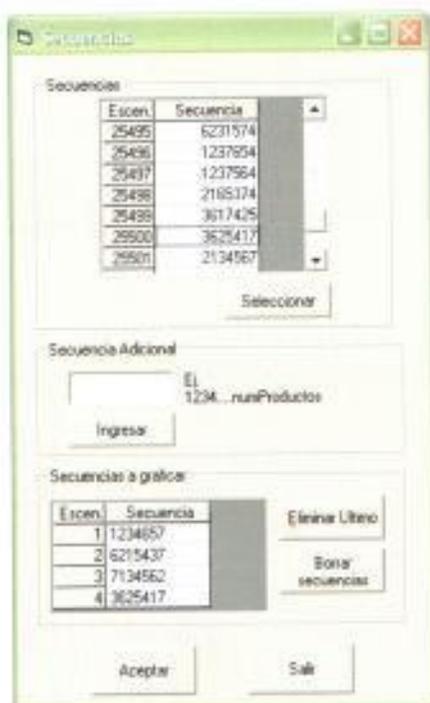
Este análisis no solo se lo puede realizar a través de los gráficos de sus ojivas sino también a través de sus componentes básico, como son la media, varianza, ganancia esperada, valor esperado, pérdida esperada, entre otros los cuales también se detallan en el cada gráficos que veremos a continuación

Sea el sistema, de acuerdo a los parámetros anteriormente descritos, donde su secuencia óptima (aquella que más veces se repite en la lista que almacenada la solución óptima de cada escenario) de acuerdo a la simulación y el algoritmo de búsqueda es:

Solución	Maquina1	Maquina2	Maquina3	Maquina4	Maquina5	Maquina6	Maquina7
Producto 1	2	3	4	6	0	0	0
Producto 2	2	3	5	0	0	0	0
Producto 3	1	2	4	6	0	0	0
Producto 4	1	3	5	6	0	0	0
Producto 5	2	3	4	5	0	0	0
Producto 6	1	3	5	6	0	0	0
Producto 7	1	2	3	4	5	6	6

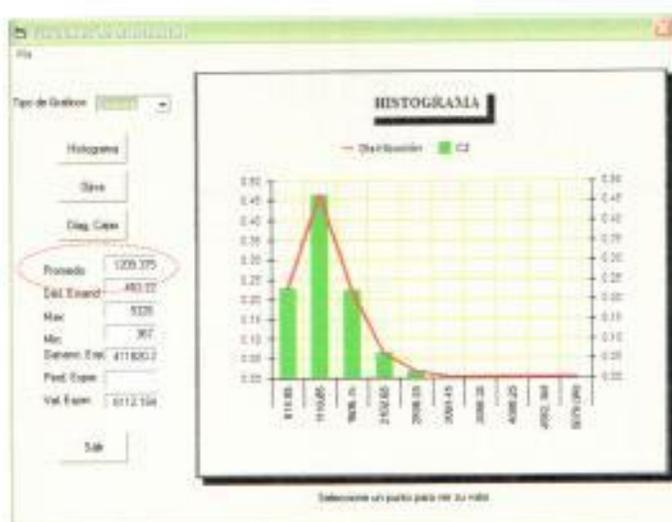
Salir

Luego de obtener la secuencia óptima de los ítems, procederemos a realizar nuevamente la simulación con la modificación que no se va ha buscar una secuencia, si no que se observara el comportamiento de la solución y compararla con otras alternativas consideradas por el analista del proceso, para su observación e inspección de su comportamiento en el tiempo, como vemos por ejemplo, las siguientes 4 secuencias seleccionadas al azar:

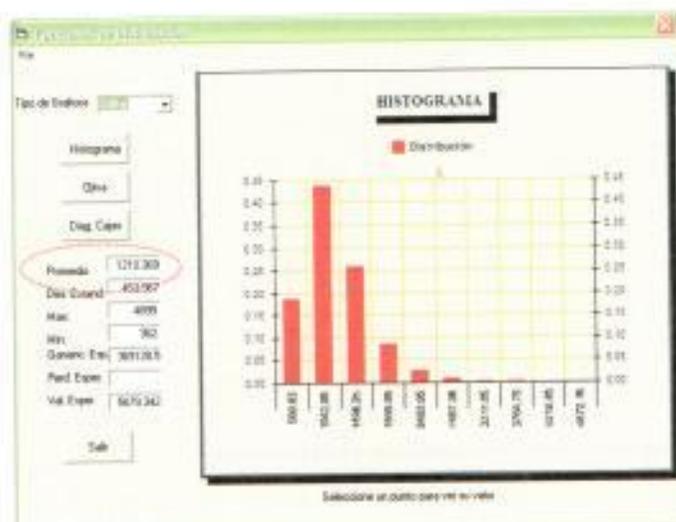


Ingresadas estas secuencias procederemos a generar la simulación de estas series y la cadena óptima del algoritmo de búsqueda, y los compararemos a través de sus gráficos o componentes básicos como vemos a continuación.

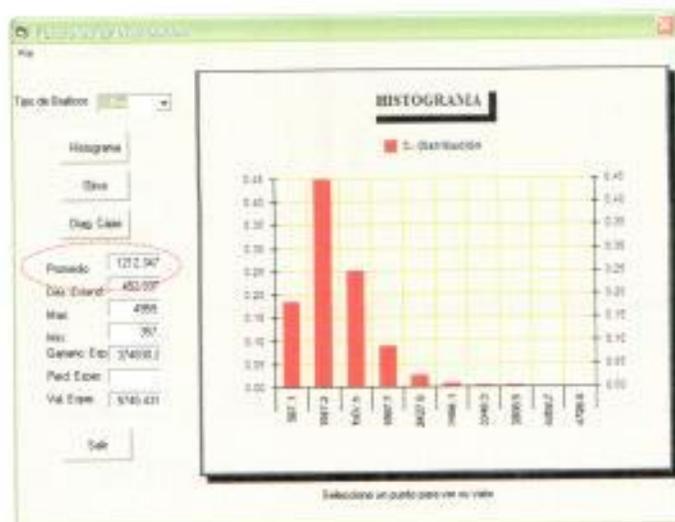
En el siguiente gráfico observamos el histograma de frecuencia de la solución óptima, con una variante, la suavización del mismo, entonces téngase en consideración los componentes básicos de la solución que se muestra en los gráficos y su comparación con las demás soluciones, con estos (promedio, máximo, mínimo, ganancia esperada, valor esperado, pérdida esperada, etc.) podemos analizar si realmente el algoritmo de búsqueda no arroja un solución que mejor se comporte a lo largo de todos los escenarios o en su defecto comparando los gráficos de las acumuladas de las soluciones entre si, como lo realizaremos en la sección siguiente.



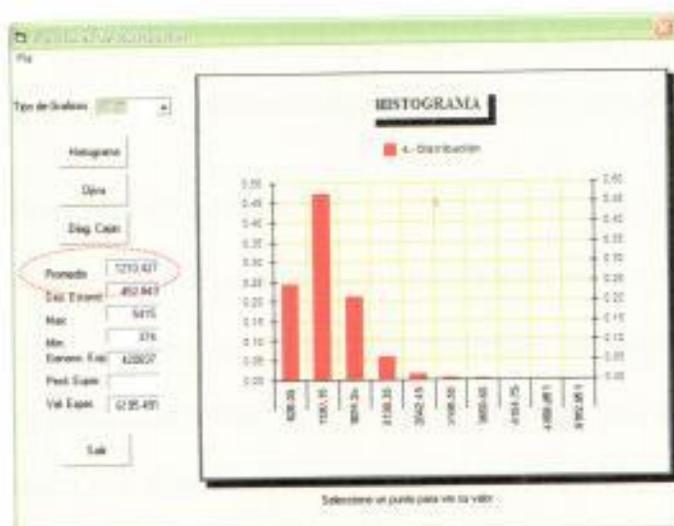
Histograma de la primera secuencia: 1;2;3;4;6;5;7
Media: 1210.309



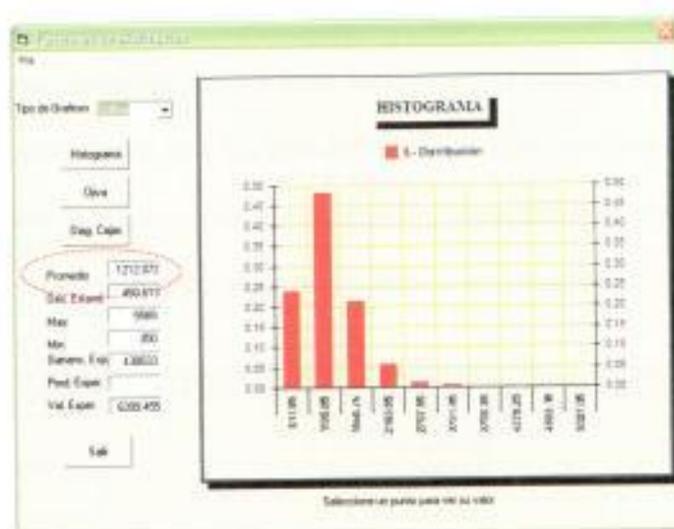
Histograma de la segunda secuencia: 6;2;1;5;4;3;7
Media: 1212.347



Histograma de la tercera secuencia: 7;1;3;4;5;6;2
 Media: 1210.427



Histograma de la cuarta secuencia: 3;6;2;5;4;1;7
 Media: 1212.072

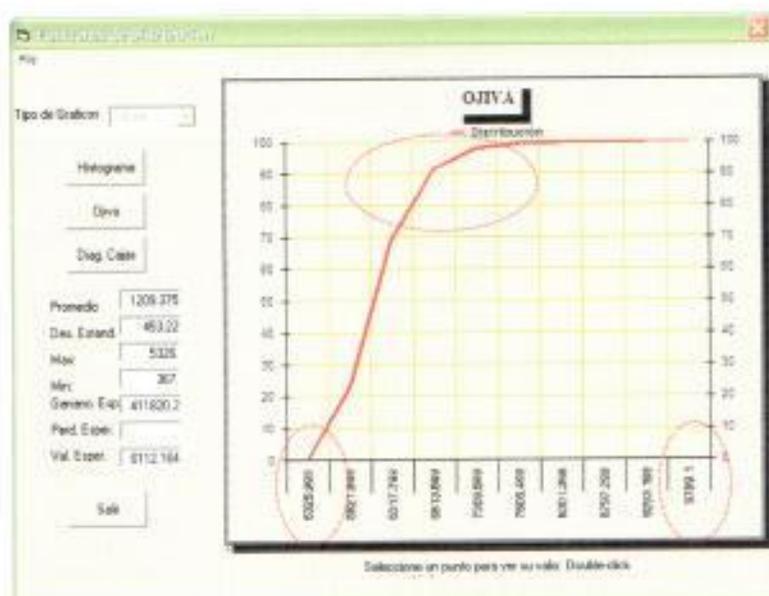


Como observamos, media de la secuencia óptima obtenida a partir del algoritmo "Tabú Search", es menor (1209.375) al resto de las secuencias comparadas, con lo cual comprobamos de la forma analítica que esta es la mejor combinación, para el proceso de producción. Es decir si consideramos que nuestro objetivo es hallar una secuencia tal que

minimice el tiempo total de procesos de la cadena de producción y dado que el histograma de la secuencia óptima deberá presentar una mayor concentración en su lado izquierdo (menos tiempo), que en su lado derecho (mayor tiempo), con lo cual se comprueba, que la secuencia óptima deberá tener un promedio menor a cualquier secuencia escogida al azar.

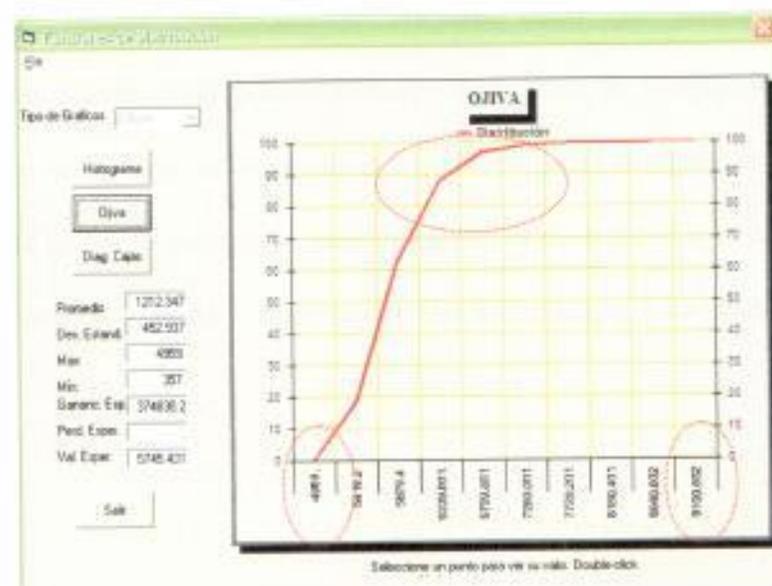
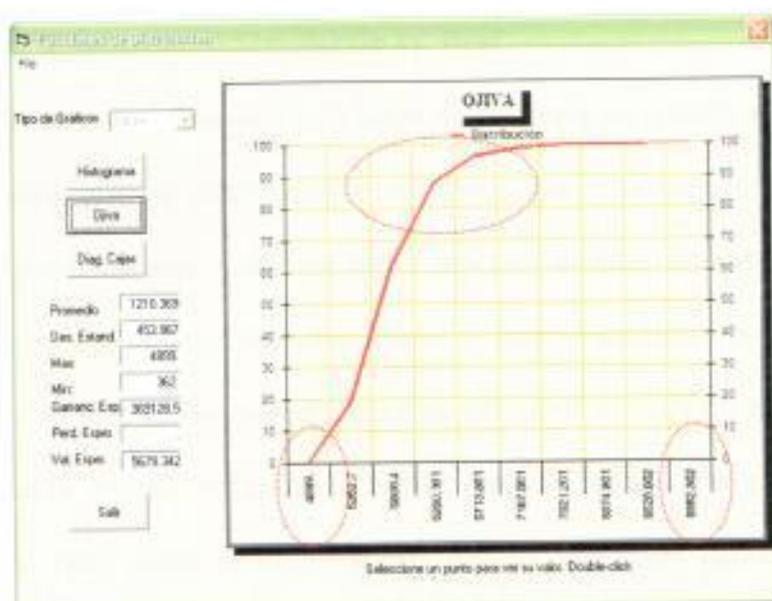
6.2 Análisis Función Acumulada.

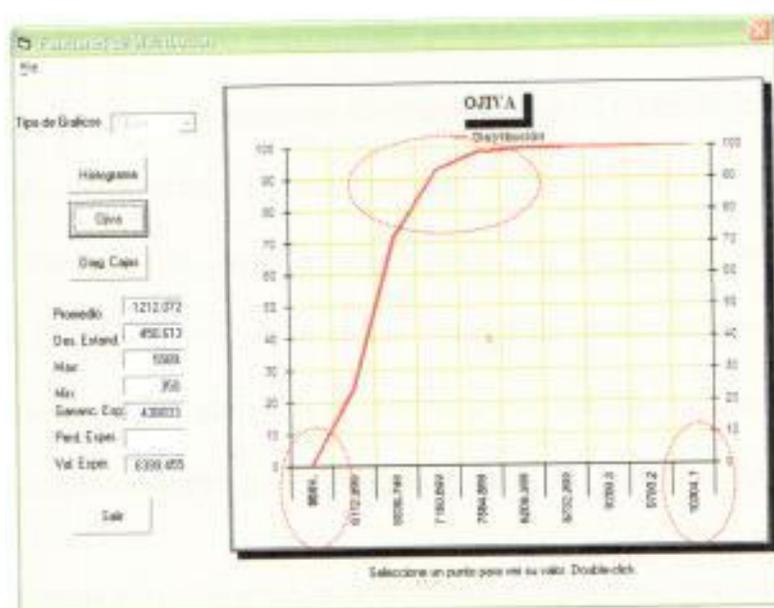
En esta sección analizaremos, la función acumulada de cada secuencia escogida a la azar, incluida la secuencia óptima, para lo cual deberemos tener en cuenta que nuestra Ojiva, deberá converger en forma desacelerada en relación a las demás, debido que como indicamos el objetivo es encontrar una solución cuyo tiempo se el menor posible, es decir al inicio del periodo de tiempo (tiempo mínimo y máximo), la mayor concentración de eventos para la secuencia óptima deberá ocurrir en el margen izquierdo como lo comprobamos en los histogramas de frecuencias, con lo cual inferimos que la curva de la función acumulada inicialmente será mas levantada que las otras secuencia y al finalizar la curva deberá crecer en forma menos desacelerada que las otra curvas, otra característica importante es que el punto de intersección de la función óptima acumulada, con respecto a las demás ojivas deberá ser siempre mayor al 50%, es decir su ganancia deberá ser siempre mayor con relación a los demás.



Como observamos se cumple que los gráficos de las funciones acumuladas de las secuencias elegidas al azar, cumplen:

- La convergencia del gráfico, no es rápida, en cambio en la secuencia óptima su convergencia es "acelerada"
- La media de los escenarios es mayor que la media de la secuencia óptima
- El punto de intersección de todas las funciones acumuladas con respecto a la secuencia óptima está por encima del 50%, esto es otra característica que identifica cual secuencia tiene una mejor ganancia o tiene el menor tiempo en procesar todos los items requeridos.





6.3 Costos Relacionados: Simulación vs. Real

Como hemos confirmado a lo largo de los diversos ejemplos propuesto, es más económico realizar sistemas que combinen las técnicas de simulación y optimización combinatorial, y de esta forma modelar entornos los más colindantes a la realidad, lo cual es muy complejo dado al comportamiento de las diferentes variables que se involucran en todo proceso, este tipo de análisis no solo resulta económico, dado que no es necesario probar las soluciones en el sistema real de producción, el cual; como es lógico podría generar cuantiosas pérdidas, sino que adicionalmente resulta tremendamente eficiente dado que probar cada posible solución en un sistema real podría demorar hasta años de verificación, lo cual es innecesario en un sistema de simulación cuyos resultados son generados y probados en cuestión de horas. El tiempo de procesamiento va de acuerdo a la complejidad del modelo matemático y al tipo de meta-heurística empleada, esto último nos llevo a

utilizar una de las heurísticas mayormente probadas y más eficientes dado que su metodología se basa en procedimiento sistemáticos y no en procedimientos aleatorios, generando soluciones dentro de una vecindad y desechándolas de acuerdo a sus técnicas de memoria de corto y largo plazo. Es importante recalcar que la convergencia del algoritmo se basa en las técnicas de memorias que utiliza, y a su rapidez en buscar óptimo que no sean locales, por lo anterior es una de las técnicas mayormente aceptadas dentro de la inteligencia artificial (Tabú Search).

En sistemas efectivos realizar una simulación generaría pérdidas económicas por concepto de materiales y tiempo invertido, como se demuestra posteriormente la simulación de los procesos productivos tomaría cuestión de hora en arrojar resultados fiables, mientras en un sistema real nos tardaría un par de vidas en realizar todo el proceso: optimización y simulación, esta es un punto importante la implementación de este tipo de sistemas estocásticos.

Por último, pero no menos importante es el costo financiero de producir este sistema, como sabemos existen diversas técnicas para costear un software, para nuestro sistema tomaremos como relevantes tres puntos característicos del mismo:

- ⬇ Valoración por código fuente. (Líneas de código multiplicado por un factor)
- ⬇ Por tecnología. Tendencia tecnología aplicada en la programación (lenguaje de desarrollo)
- ⬇ Por metodología. Métodos aplicados en la resolución del sistema
- ⬇ Por valor agregado. Funciones u objetivos adicionales que el sistema satisface.

A continuación procederemos a dar un valor estima del presente (tal como se encuentra estructura actualmente, las mejoras no se tomarán en cuenta para los efectos de valoración) sistema.

Rubro por número de líneas en código fuente y lenguaje de programación:

Luego de revisar cuidadosamente el programa y contar todas sus líneas de programación llegamos a la conclusión que existen entre comentarios y líneas de procesamiento 5.152 líneas de código fuente, si bien es cierto Visual Basic aun dista de ser un lenguaje totalmente orientado objeto, es muy eficiente para este tipo de procesos, y su implantación resulta relativamente viable

Por tecnología, este factor es muy importante dado que el desarrollo de la presente tesis se basa en metodologías, aun en desarrollo, e investigación, aun que la Meta-heurística (Tabú Search), ha sido desarrollado en todo su potencial desde 1999, con el resurgimiento de algoritmos genéticos, de búsqueda, híbridos, etc., estos aun en la actualidad están en procesos de desarrollo y como se demuestra a lo largo del presente trabajo es una herramienta de gran potencial, que en combinación con la simulación de procesos estocásticos, da soluciones garantizados y en un corto periodo de tiempo, ideal para un tomador de decisión.

Por valor agregado, la ejecución del sistema no solo se puede realizar para el estudio de escenarios durante un periodo de tiempo, sino que el mismo nos ayuda en el estudio del proceso cuando cambie el estados de sus variables de entrada (funciones distribución), es decir si el comportamiento del sistema cambia por desgaste propios de los equipos esta simulación se podrá volver a ejecutar las veces que sea necesario

y encontrar nuevas soluciones de acuerdo al nuevo entorno de trabajo o cambio de las variables de entrada. Algo parecido al estudio de un nuevo proceso con otras condiciones iniciales.

Rubros:

◆ Líneas de código: 5.152	900 US
◆ Lenguaje de programación: Visual Basic	300 US
◆ Metodologías	
Meta-Heurística: Tabú Search	1.700 US
Simulación de sistemas: Procesos estocásticos:	1.500 US
◆ Valor agregado: Nuevas condiciones iniciales	500 US

Costo del Sistema: Optimización - Simulación

Total sin incluir impuestos 4.900 US.

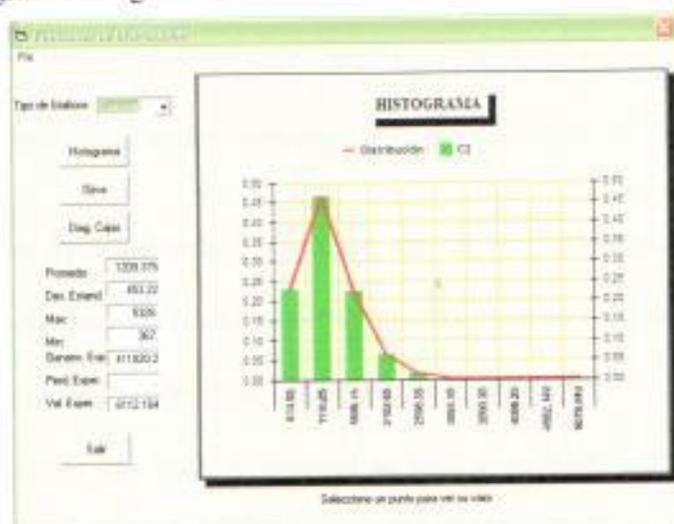
6.4 Pruebas y Resultados

Como se comprobó la secuencia óptima en esta simulación de acuerdo a los parámetros previamente descritos es:

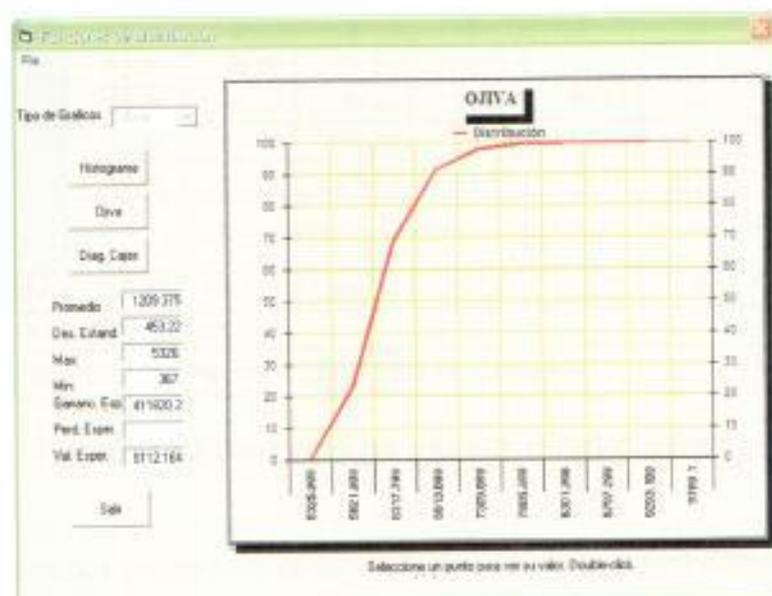
Solucion	Máquina1	Máquina2	Máquina3	Máquina4	Máquina5	Máquina6	Máquina7
Producto 1	2	3	4	5	0	0	0
Producto 2	2	3	5	0	0	0	0
Producto 3	1	2	4	5	0	0	0
Producto 4	1	3	5	6	0	0	0
Producto 5	2	3	4	5	0	0	0
Producto 6	1	3	5	6	0	0	0
Producto 7	1	2	3	4	5	6	7

Salir

Siendo su histograma el siguiente



Observamos, claramente que el promedio de 1209.375, el cual como se indico es menor que los promedios de las otras secuencias seleccionadas al azar.



Por ultimo, la gráfica función acumula de la secuencia óptima converge más rápidamente que los demás gráficos de funciones acumuladas, cabe indicar que el tiempo empleado para realizar esta simulación fue aproximadamente 25 minutos, lo

cual es eficiente y a la vez económico desde el punto de vista de un sistema real, debido que para poder realizar todo este tipo de análisis en un sistema real, tendríamos los siguientes inconvenientes:

a.- El tiempo en realizar esta simulación en un sistema real, con los mismos parámetros tardaría años en realizar solo el evento de obtener la secuencia óptima y siglos en realizar ambas la optimización y simulación.

Es necesario, recalcar que todo el proceso se realizo en base a 30.000 escenarios y cada escenario tiene un promedio de demora de acuerdo a los datos históricos de los eventos realizados de 32.26 horas, es decir para ejecutar los 30,000 eventos en el sistema real nos demoraríamos aproximadamente 967.660 horas.

MAQ. \ ÍTEMS	1	2	3	4	5	6	7	
1	78.90	78.80	47.10	100.50	44.22	71.40	89.23	
2	89.50	60.57	70.90	23.60	90.30	56.10	77.10	
3	46.80	45.90	100.20	99.30	80.30	31.40	78.20	
4	36.40		88.10	67.80	19.70	100.70	22.80	
5							99.80	
6							39.70	
TOTAL*ÍTEM	251.60	185.27	306.30	291.20	234.52	259.60	406.83	
TIEMPO TOTAL (MINUTOS)								1935.32
TIEMPO PROMEDIO DE PRODUCCIÓN TOTAL (HORAS)								32.26

Tabla # 4 **TIEMPO PROMEDIO DE PRODUCCIÓN (MINUTOS)**

Si consideramos que se trata de una industria donde se labora a tres turnos las 24 horas del día nos tardaríamos aproximadamente 110.46 años en realizar solo la opción de optimización, a esto debemos incrementar; el tiempo anterior por el número de secuencia a probar, es decir que nuestro caso son 3 secuencias alternas y una óptima (comprobación) el tiempo que demoraría el sistema real en realizar solo la

simulación es aproximadamente de 441.85 años, y si queremos realizar la optimización y la simulación es decir todo el proceso nos tardaríamos 552.32 años aproximadamente, lo cual nos tomaría varias vidas en realizar dicho experimento, en relación con los 25 minutos que se demora el sistema en realizar todo el proceso no representa mayor pérdida de tiempo y en su lugar mejora la producción. Entonces, económicamente resulta rentable realizar este tipo de sistemas en procesos de producción, donde intervienen factores de incertidumbre en las variables de entrada, ya que su periodo de prueba es corto y además resulta económicamente factible su implementación.

Otro de los puntos relevantes, de este tipo de sistemas es que estos no intervienen directamente sobre el sistema real, todo es realizado en base de datos históricos (generación de histograma) y su aplicación solo es realizada cuando tenemos la certeza que la secuencia óptima es la correcta, con lo cual disminuimos las pérdidas económicas por concepto de pruebas e implantación, ya que no se altera el ciclo de producción en ningún instante.

Conclusiones y Recomendaciones.

*“El tipo de meta-heurística más importante es el de las meta-heurísticas de búsqueda, que establecen estrategias para recorrer el espacio de soluciones del problema, transformando de forma iterativa soluciones de partida. Las búsquedas evolutivas se distinguen de éstas en que es un conjunto de soluciones, generalmente llamado población de búsqueda, el que evoluciona sobre el espacio de búsqueda.” **

Con el presente trabajo, se ha valorado el efecto del uso de la meta-heurística: *Tabú Search* en los procedimientos de optimización con métodos estocásticos proporcionando técnicas para escaparse de los óptimos locales de mala calidad, dado que el valor de tales óptimos locales frecuentemente difiere notablemente del valor del óptimo global. En general la metodología de búsqueda entregó soluciones robustas de buena calidad, y el uso menos agresivo de la meta-heurística rebaja los tiempos de procesamiento aunque supone un cierto deterioro en la calidad de las soluciones.

Como evidenciamos, en la experimentación a través del modelo de simulación y a manera de recomendación, tener en cuenta ciertos factores como son: la estimación de parámetros, estabilización de variables, validación del modelo y entre otros (adicionalmente a esto se ha involucrado el criterio de otros investigadores al presente trabajo)

- Como una posterior mejora al presente trabajo, se recomienda utilizar los datos históricos generados de cada análisis de la vecindad solución, es decir almacenar las respectivas “soluciones”, las cuales en futuras corridas y bajo las mismas condiciones iniciales, serán reutilizadas o en su defecto no se volverá analizar dichas vecindades, optimizando de esta forma tiempo en la resolución de la secuencia óptima.
- En la experimentación con un modelo de simulación y optimización combinatorial es conveniente examinar el efecto que puede tener en los resultados, las variaciones en los parámetros de entrada.

- ☑ El equilibrio de las variables se debe tomar en cualquier instante después del transiente.

Fig. 6.1*

- ☑ Se realizan muestreos de la producción para obtener los valores μ y s ; límite máximo y límite mínimo, coeficiente de correlación, y otros.
- ☑ Verificar que el modelo sea capaz de reproducir el sistema adecuadamente.
- ☑ Dado la independencia de las semillas con la que se generan las distribuciones estadísticas, se tiene la certeza de la independencia de las variables de entrada.
- ☑ Para obtener una buena solución se considera aquellas localizadas, después que se estabiliza la función de penalización.

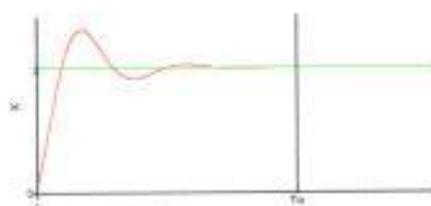


Fig. 6.1 FUNCIÓN DE ESTABILIZACIÓN

Dado que el marco de aplicación de la “*Optimización Combinatorial*” se ha acrecentado en la actualidad y que la mayoría de los científicos se han ladeado por el uso de esta técnica, ya que la meta-heurística ofrece más oportunidades para aplicar la intuición que la deducción: Esto último ha generado diversas directrices en las exploraciones sobre técnicas meta-heurísticas. “*unas tratan de mantener la pureza de los métodos y comprobar su efectividad, sin incorporar herramientas de otras meta-heurísticas, otras tratan de aprovechar los recursos proporcionados por cada una de ellas...*”. En la práctica a una mayor cantidad de problemas, no resulta apropiado utilizar procedimientos específicamente diseñados para cada modelo, esto por que; las meta-heurísticas proporcionan estrategias generales de diseño.

*“Los intentos por organizar este campo son numerosos, pero los conceptos principales son raramente definidos con precisión y hay todavía muy pocos teoremas significativos. Ninguna estructura ha conseguido una aceptación general.” **

Es por esto que cada grupo de investigadores de una meta-heurística tiene su punto de perspectiva diferente para explicar las heurísticas, así como para extraer ideas de todos los campos, por ejemplo los híbridos donde sus aportaciones de meta-heurísticas escasamente son objetos de un estudio experimental adecuadamente fundamentado.

Por lo general, se delinean nuevas heurísticas, con algo de originalidad, pero las causas de por qué se desempeñan bien permanecen aun desconocidas, algunas propuestas definidas para una mejor comprensión de estos aspectos es el estudio de la *“influencia de la topografía de los óptimos locales”*.

*“El propósito de estas investigaciones debe ser la comprensión de las meta-heurísticas, no la competición entre ellas. Otras cualidades de las heurísticas y las meta-heurísticas distintas que la eficiencia pueden ser tan importantes a la larga, como la simplicidad, la precisión, la robustez, y, sobre todo la, amigabilidad.” **

*Belén Melián, José A. Moreno Pérez, J. Marcos Moreno Vega; Publicación: *Metaheuristics: A global view*, Centro Superior de Informática: Universidad de La Laguna

Por último y como se ha concluido a lo largo de la presente investigación:

- ◆ Resulta económicamente rentable realizar sistemas de optimización y simulación en procesos de productivos.
- ◆ No solo por la característica de no intervenir en los sistemas reales, sino por que además su tiempo de implementación así como su costo de prueba resulta efectivo

- Desde el punto de vista del tomador de decisión, sistemas como los desarrollados en la presente tesis, ayudan a probar distintos métodos de producción. Verificando su resultado final sin la necesidad de detener los procesos o generar pérdidas económicas.
- Entre los métodos meta-heurísticos más utilizados son los algoritmos de búsqueda por que los mismos están diseñados en base de procedimientos sistemáticos y no en métodos aleatorios.
- Se recalca que el hecho de disminuir el número de iteraciones, convergencia del método Tabú Search, degrada significativamente la solución, para contrarrestar este efecto se debe comenzar analizar al sistema posterior al transiente de inicio de los procesos, es decir afinar la función de penalización (convergencia del error)
- Si disminuimos el número de escenarios, el tiempo de ejecución se reduce drásticamente y la solución se deteriora levemente, por lo que dependiendo de la cantidad de procesos y la urgente necesidad de una respuesta aceptable "no óptima", se puede disminuir el número de escenarios, pero la solución arrojada con estos parámetros jamás deberá ser considerada como una solución definitiva.
- De lo anterior se desprende, que aunque la Meta-Heurística: Tabú Search tenga un eficiente método de análisis de soluciones a través de su procedimiento de memoria: corto y largo plazo (base de los algoritmos de búsqueda), su mayor inconveniente para analizar sistemas de números procesos, estará ligada a las características del equipo en el cual se desee ejecutar el sistema de simulación y optimización, ya que la mayoría de los procesos que estos ejecutan se los realizan en la memoria del equipo.
- El sistema actual debido a que uso es académico tiene la limitante de 50 variable aleatoria independientes de entrada (funciones de distribución de las variables de entrada), esto es fácilmente de mejorar, si aumentamos la cantidad de semillas independientes. (para

confirmar que son semillas independiente verificar la correlación de estas con respecto a cada una de las ingresadas)

ANEXO A

ANEXO A: INTRODUCCIÓN AL ANÁLISIS DE DATOS Y SUS APLICACIONES: "INVESTIGACIONES PUBLICADAS"

A continuación enunciaremos los trabajos mas relevantes sobre meta-heurística y sus aplicaciones en diferentes campos, como observaremos en la misma, no solo incluye publicaciones de optimización a través TABÚ SEARCH.

▶ ANÁLISIS DE PROXIMIDADES MÉTRICO USANDO BÚSQUEDA TABÚ

Mario Villalobos – Javier Trejos

Resumen

"Se aplica la técnica de búsqueda tabú (BT) en análisis de proximidades, obteniéndose buenos resultados, comparables a los obtenidos con sobrecalentamiento simulado. Un estado en BT es una configuración de n puntos en un espacio p dimensional, y un vecino se define por la traslación de longitud h de una o más de las coordenadas de un punto."

▶ ALGORITMOS PARA LA CLASIFICACIÓN PIRAMIDAL SIMBÓLICA

Oldemar Rodriguez – Paula Brito – Edwin Diday

Resumen

"En este artículo se define el concepto de pirámide simbólica, además se presentan dos algoritmos para generar este tipo d pirámide a partir de una matriz de datos simbólicos. El primer algoritmo (CAPS) encuentra un "orden total compatible con la pirámide" de los n objetos, mientras que el segundo (CAPSO) construye la pirámide a partir de un orden dado a priori en los objetos, dicho orden se recibe como entrada en el algoritmo. Ambos algoritmos, además de producir la pirámide, para cada grada encuentran el objeto simbólico asociado a cada nodo y su extensión. También se presentan los teoremas de convergencia."

▶ REGRESIÓN Y ANÁLISIS FACTORIALES

Róger Lafosse

Resumen

"Este analiza algunos resultados recientes que se han aportado en análisis factorial en el contexto de "multi-sets" (incluyendo cubos de datos). Los análisis factoriales más clásicos se basan en la noción de descomposición en valores singulares. Por ello, los análisis de 2 tablas se introducen aquí a partir de un criterio cualitativo, el de no redundancia de las relaciones parciales entre "factores comunes". Una proposición de extensión de la noción de regresión lineal simple, considerada aquí entre los dos conjuntos de individuos que definen los dos conjuntos de variables, conduce a medidas específicas para cada una de las tablas. Entonces, ciertas yuxtaposiciones de gráficos pueden ser justificadas. El conjunto del procedimiento es prolongado para analizar la dependencia de K tablas con una $(K+1)$ -ésima. El ACOM de Chessel y Hanafi (1996) es interpretado en este contexto como un ACP de conjuntos de variables."

► **ANÁLISIS DIGITAL. UNA NUEVA TÉCNICA PARA LA REPRESENTACIÓN DE FUNCIONES ACOTADAS QUE, EN UN INTERVALO FINITO, SATISFACEN LAS CONDICIONES DE DIRICHLET**

Osaldo Skliar – Victor Medina – Tatiana Láscaris Comneno

Resumen

“Se presenta una nueva técnica para aproximar, con tanta precisión como se desee, cualquier función acotada de una variable – especificada en un intervalo finito en el cual puede tener a lo sumo un número finito de máximos y mínimos locales - mediante una suma de trenes de ondas cuadrada.”

► **UN PROBLEMA DE LOCALIZACIÓN DE PLANTAS DE GRAN ESCALA**

M.A. Gutiérrez – S. De los Cobos – B.R. Pérez – J. Goddard

Resumen

“En este artículo se desarrolla un algoritmo heurístico y su correspondiente implementación para resolver un problema de localización de plantas (facility location) de gran escala, en donde surgen potencialmente más de 640 plantas a localizar a lo largo de la República Mexicana. Originalmente se trató de obtener solución exacta al problema, usando dos técnicas clásicas: descomposición de Benders y ramificación y acotamiento. Ambas técnicas resultan adecuadas y eficientes para resolver problemas de tamaño chico, pero las implantaciones en computadora para este problema no convergieron después de muchas horas de proceso. Se requería obtener una solución al problema mediante alguna técnica que quizá no diera la solución exacta, pero sí una solución de buena calidad. Para la solución de este problema real, se empleó la técnica de recocido simulado (simulated annealing) con excelentes resultados.”

► **ANÁLISIS FACTORIAL DE CORRESPONDENCIAS INTRACLASES PONDERADO**

B. Castro Iñigo – A. Zárraga Castro – M.A García Montoya

Resumen

“En este artículo se presenta una técnica de Análisis Factorial útil para el estudio de tablas de datos en las que existe una partición sobre el conjunto de individuos, definida de forma natural. Dicha técnica permite poner de manifiesto las relaciones existentes entre los individuos, pertenecientes a una misma clase, discriminándolos, a su vez, de los componentes de otras clases por los pesos relativos que se asignen a las mismas. Con una selección apropiada de la ponderación asignada a cada clase, la nueva metodología obtiene las variables locales del Análisis Parcial, contando con la ventaja sobre éste de proporcionar una representación de los individuos, como elementos activos, en los ejes locales.”

► **UN ALGORITMO PARA EL ENTRENAMIENTO DE MÁQUINAS DE VECTOR SOPORTE PARA REGRESIÓN**

J. Goddard, S.G. del Cobos Silva, B.R. Pérez Salvador, M.A. Gutiérrez Andrade

Resumen

“El propósito del presente artículo es doble. Primero se proporciona una introducción a las ideas básicas de la Máquinas de Vector Soporte para regresión. Posteriormente, se presenta un algoritmo novedoso y sencillo, basado en el trabajo de Campbell y Cristianini [16], que resuelve de manera fácil el correspondiente problema de programación cuadrática. Se ilustra el algoritmo con ejemplos, y se compara con el método de regresión clásico.”

► **UN ALGORITMO PARALELO PARA EL PROBLEMA DEL CONJUNTO INDEPENDIENTE**

Rafael López Bracho y María Paula Ortuño Sánchez

Resumen

“Un conjunto S de vértices de una gráfica G es independiente si no existen dos vértices de S que sean adyacentes, esto es, la subgráfica de G inducida por S no tiene aristas. En este trabajo presentaremos un algoritmo paralelo que permite la obtención de todos los conjuntos independientes maximales de una gráfica. Presentaremos los fundamentos del algoritmo y algunas propiedades derivadas de éstos.”

► **GRAFOS DESTINADOS AL ANÁLISIS DE DATOS MULTIVARIANTE CONDICIONADO POR SU ESTRUCTURA**

M.A. García Montoya – B. Castro Iñigo – A. Zárraga Castro

Resumen

“En ese artículo se muestran dos nuevos métodos para la construcción de grafos que pueden utilizarse en combinación con las técnicas descriptivas de Análisis de Datos Multivariante: grafo basado en la distancia ji-cuadrado y grafo basado en los componentes fuertemente conectadas. Además se presentan sendos ejemplos correspondientes a cada uno de los grafos, tomando como datos la matriz intermedia de las Tablas Input- Output.”

► **A DICHOTOMOUS PROPERTY OF THE TOTAL VARIATION OF A PROCESS WITH INDEPENDENT INCREMENTS**

Jaime Lobo Segura

Resumen

“Establecemos una propiedad para la variación total de un proceso d incremento independiente, que es dicotómica en el sentido de que solo dos alternativas son posibles. Para este efecto introducimos los métodos del análisis no-standard con el estudio de los procesos PII en casi- intervalos. Finalmente discutimos, en el caso de procesos continuos, una condición equivalente para una de las alternativas del teorema principal.”

► **ALGUNOS ASPECTOS TEÓRICOS DE LAS FUNCIONES CASI PERIÓDICAS N-DIMENSIONALES**

Vernor Arguedas – Edwin Castro

Resumen

"Presentamos una definición de función casi periódica, la cual generaliza la definición usual en \mathbb{R} . A partir de esa definición demostramos algunas propiedades topológicas para esta clase de funciones. Al final del artículo, demostramos algunas propiedades algebraicas usando el teorema de estructura que se incluye en el apéndice."

► **EL ANÁLISIS MULTIVARIADO DE DATOS COMO HERRAMIENTA EN LA EXPLORACIÓN DE MODELOS EN LA COMPRESIÓN DE VARIABLES PSICOLÓGICAS**

Carlos R. Seijas

Resumen

"La psicología como ciencia de la conducta en un estado paradigmático necesita del apoyo de modelos teóricos que validen y confiabilicen los constructos que esta rama del conocimiento utiliza para explicar los fenómenos que le competen estudiar. Para sustentar los avances en la construcción de modelos provenientes de diversos campos de la psicología, se presenta la aplicación de herramientas del análisis multivariado de datos tales como el Análisis Factorial de Componentes Principales (ACP), el Análisis Factorial de Correspondencias (AFC) y el Análisis Factorial Discriminante (AFD), en el estudio de variables como la predicción del desempeño universitario de acuerdo a variables clásicas de la psicometría (Seijas, 1998a) y a variables cognoscitivas (Seijas, 1999b), los estilos de aprendizaje (Seijas, 1998b), la autoestima (Seijas, 1999a), y finalmente variables transculturales como son el individualismo y el colectivismo (Seijas, 1999c). Con esto se persigue mostrar la utilidad que el análisis multivariado de datos puede prestar a las ciencias de la conducta en su meta de forjar constructos teóricos con poder explicativo de la realidad, para que puedan ser de instrumentales operativos en el quehacer diario de todos los interesados en el análisis y manejo de la conducta."

► **DYNAMICS OF A TWO-DIMENSIONAL DISCRETE-TIME SIS MODEL**

Jaime H. Barrera – Ariel Cintrón Arias – Nicolás Davidenko –
Lisa R. Denogean – Saúl Ramón Franco González

Abstract

"We analyze a two-dimensional discrete-time SIS model with a non-constant total population. Our goal is to determine the interaction between the total population, the susceptible class and the infective class, and the implications this may have for the disease dynamics. Utilizing a constant recruitment rate in the susceptible class, it is possible to assume the existence of an asymptotic limiting equation, which enables us to reduce the system of two-equations into a single, dynamically equivalent equation. In this case, we are able to demonstrate the global stability of the disease-free and the endemic equilibria when the basic reproductive number (R_0) is less than one and greater than one, respectively. When we consider a non-constant recruitment rate, the total population bifurcates as we vary the birth rate and the death rate. Using computer simulations, we observe different behavior among the infective class and the total population, and possibly, the occurrence of a strange attractor."

► **SPATIAL INTERPOLATION OF DRY DEPOSITION USING EOF MODELS**

Breda Muñoz – Hernández, Virginia M. Lesser and Fred L. Ramsey

Abstract

“Random processes are monitored over space and time by a network of stations distributed across a spatial region. Auxiliary information is often gathered not only at the stations but at other points across the region. The incorporation of auxiliary information in some interpolation techniques has show improvement on the interpolation results. The Empirical Orthogonal Functions (EOF) model is a well-known eigenvector based prediction technique widely used in meteorology and oceanography for modeling the variability of the observed spatio-temporal random process. Similarity matrices are constructed using available auxiliary information and included in the EOF model to develop a spatial interpolation method. The resulting interpolation technique will be applied to real data set and the results compared to ordinary kriging.”

ANEXO: B

ANEXO B:

MODELO Y ANÁLISIS ESTADÍSTICO

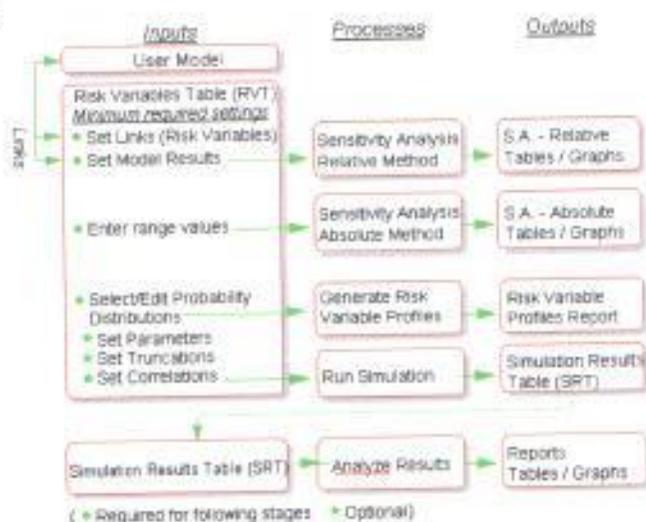
En la actualidad contamos con software que tiene implementado algunas de las herramientas de análisis estadístico, como las antes mencionadas, pero todavía se encuentran en proceso de desarrollo herramientas que combinen la optimización y la simulación de sistemas a través de procesos estocásticos y optimización robusta. Entre los sistemas más utilizados para este tipo de análisis tenemos dos herramientas:

Arena y

RiskEase

La primera nos ayuda en la simulación gráfica del sistema, es decir nos ayuda a comprender de una manera gráfica, la situación de los procesos además de emitir informes de los posibles cuellos de botellas a lo largo de toda la etapa de producción.

Mientras que RiskEase nos provee de herramientas estadísticas como son varianza, correlación, gráficos (histográfico, acumuladas, ojivas, densidad de distribución, ect).



Como hemos indicado, aún en la actualidad no contamos con herramientas que combinen la simulación (procesos estocásticos) con la optimización (meta-heurística), y aun que ambos programas son excelentes herramientas de estudio no tienen incorporado las herramientas de optimización robusta.

Es decir con ellas podemos solamente comprobar los resultados finales más no todo el proceso, por ejemplo después de la optimización tomamos la secuencia óptima encontrada por el algoritmo TABU SEARCH, y la comparamos con otras secuencias seleccionadas al azar, esta comparación nos dará como resultado que la mejor secuencia, es la encontrada por la optimización robusta.

Como hemos observado en ningún caso esta herramienta podrá reemplazar la opción de optimización robusta y en el mejor de los casos nos servirá como un complemento para el análisis de los resultados (herramientas estadísticas), cabe recalcar que las herramientas de análisis proporcionadas por este programa son de excelentes resultados, y de recomendado estudio para su posterior aplicación.

Por último, después de realizar el proceso de optimización en el antes mencionado ejemplo se procedió a realizar el análisis de los resultados con esta herramienta llegando a la misma conclusión, que la secuencia de eventos proporcionada por el algoritmo TABU SEARCH, tiene una mejor ganancia que cualquier otra secuencia escogida al azar, esto se confirmó con los gráficos de las acumuladas de todas las secuencias ingresadas.

ANEXOS: C

ANEXO C:**GLOSARIO****Cadena pseudoaleatoria.-**

Cadena para la que aparentemente no existe ninguna forma de comprimir su descripción.

Conjunto factible.-

Es el conjunto delimitado por las restricciones del problema.

Convergencia.-

Dicho de una sucesión: Aproximarse a un límite, efecto de convergir.

Distribución.-

Cuando se investiga generalmente se tiene una muestra aleatoria, la cual sirve para aproximar algunas medidas de la población de donde se extrajo dicha muestra. Para llevar a cabo lo anterior, es pertinente conocer el comportamiento de la distribución de los datos en la población. Esto permite al investigador poder aplicar las medidas estadísticas adecuadas en el proceso de inferencia estadística. A la tabla, fórmula o gráfico que contiene todos los valores de la variable y sus respectivas probabilidades, se le conoce con el nombre de función de probabilidad de la variable

Distribuciones Independientes.-

Dos sucesos son independientes entre sí, si la ocurrencia de uno de ellos no afecta para nada el que pueda producirse el otro.

Frecuencia Relativa.-

La frecuencia relativa es el cociente entre la frecuencia absoluta (f_i) y el número total de datos (n).

Función Acumulada.-

La Función de Distribución Acumulada corresponde a la probabilidad de que la variable aleatoria X tome un valor numérico menor o igual a, x o representa el acúmulo de las probabilidades hasta alcanzar el valor de interés. Simbólicamente, lo anterior se expresa como:

$$F(x) = P[X \leq x]$$

La función de probabilidad acumulada $F(x)$ cumple con las siguientes propiedades:

- (i) La gráfica de la función nunca decrece.
- (ii) El valor de la función de probabilidad acumulada cuando el valor de la variable es demasiado grande (x tiende a más infinito) se acerca a uno.
- (iii) El valor de la función de probabilidad acumulada cuando el valor de la variable es demasiado pequeño (x tiende a menos infinito) se acerca a cero.

Función objetivo.-

Es la función que debemos maximizar o minimizar y en algunos casos optimizar.

Heurística.-

La heurística propone métodos, determinísticos o estocásticos para salir de mínimos locales y métodos de optimización robusta para la selección de una solución que mejor se comporte entre todos los posibles escenarios.

Histograma.-

Un histograma es un resumen gráfico de la variación de un conjunto de datos. La naturaleza gráfica del histograma nos permite ver pautas que son difíciles de observar en una simple tabla numérica. Esta herramienta se utiliza especialmente en la Comprobación de teorías y Pruebas de validez.

Lista Tabú.-

Evaluando cada una de las soluciones y quedándose con la mejor que no sea tabú. Según se defina el entorno, es decir, el conjunto de soluciones a las que se puede acceder desde x_a , así será su tamaño. Para realizar una búsqueda completa, es deseable que el tamaño no sea elevado.

Memoria Corto Plazo.-

La estrategia de movimientos rechazados del algoritmo de búsqueda tabú en su forma más simple se denomina de memoria a corto plazo, debido a que la búsqueda que se realiza es local, utilizando la memoria de los movimientos prohibidos en la lista tabú. Para problemas más complejos se pueden considerar estrategias de memorias a plazos más largos.

Memoria Largo Plazo.-

La memoria a largo plazo diversifica la búsqueda sobre regiones poco exploradas. Una forma clásica de diversificación consiste en reiniciar periódicamente la búsqueda desde puntos elegidos aleatoriamente. Un método más acorde con el algoritmo propone registrar los atributos de los movimientos más utilizados en los anteriores movimientos, penalizándolos a través de una lista tabú a largo plazo.

Meta-Heurística.-

Una metaheurística puede verse como un marco de trabajo general referido a algoritmos que puede aplicarse a diversos problemas de optimización con pocos cambios significativos si ya existe previamente algún método heurístico específico para el problema. De hecho, las metaheurísticas son ampliamente reconocidas como una de las mejores aproximaciones para atacar los problemas de optimización combinatoria [2, 76, 88].

Modelo.-

Detalla los aspectos específicos de cómo se programaría la fábrica para obtener la combinación deseada, teniendo en cuenta los tiempos de preparación de las máquinas, el tiempo de espera antes del procesamiento y otros detalles que no se pueden incluir en la formulación de la programación lineal.

Optimización Combinatorial.-

Un problema es de optimización combinatoria, si consiste en encontrar entre una cantidad finita de subconjuntos de un conjunto dado, aquél que cumpliendo con ciertas restricciones maximice o minimice una cierta función objetivo.

Optimización Robusta.-

Es la solución que se maneje bien a través de todos los escenarios potencialmente realizables (peor y mejor)

Óptimos Globales.-

Como es lógico pensar la función a optimizar puede tener máximos o mínimos los cuales serán seleccionado de acuerdo a las condiciones del algoritmo, es decir si el algoritmo requiere maximizar (en nuestro caso minimizar el tiempo de producción) la condición de fin estará orientada a obtener el punto máximo de la función.

Óptimos locales.-

Los métodos de búsqueda local presentan como principal inconveniente, el quedar atrapados en óptimos locales. En FANS se proveen dos mecanismos para tratar con este problema.

El primero está contenido en la variación del operador en las etapas de búsqueda. El segundo mecanismo de escape esta basado en el par $(cond, accion)$ donde $cond$, llamada $HayEstancamiento?() : \mathcal{P} \rightarrow [True, False]$, se utiliza para determinar cuando hay suficiente evidencia de que la búsqueda esta definitivamente estancada. Por ejemplo, se podría reiniciar el algoritmo desde una nueva solución inicial, reiniciar desde una solución obtenida a partir de una modificación especial de la actual, o cualquier otra opción que se considere adecuada.

Percentil.-

Una medida de posición muy útil para describir una población, es la denominada 'percentil'. En forma intuitiva podemos decir que es un valor tal que supera un determinado porcentaje de los miembros de la población.

Pseudo aleatorio.-

Proceso cuyo resultado es aparentemente impredecible, excepto en forma de probabilidad.

Restricciones.-

Es el conjunto de restricciones del problema que delimita el conjunto factible

Semillas Aleatorias.-

La idea (von Neumann) es producir números que parezcan aleatorios, empleando las operaciones aritméticas del ordenador: partiendo de una semilla inicial (u_0, u_1, \dots, u_{p+1}) generar una sucesión mediante $u_i = d(u_{i+1}, \dots, u_{i-p})$, para cierta función d .

Simulación.-

Mecanismos de avance del tiempo para modelos de simulación. Componentes y organización de un modelo de simulación.

Soluciones factibles.-

Es el conjunto de puntos que forman parte del conjunto factible.

Soluciones Óptimas.-

Son las soluciones factibles que optimizan la función objetivo.

Tabú Search.-

Búsqueda Tabú define el entorno reducido $N'(x)$ en aquellas soluciones viables del entorno de x . Así, se considera que a partir de x , sólo las soluciones del nuevo entorno son factibles.

Variables de decisión.-

Son las variables de la función objetivo y sobre las que hay que determinar su valor en el conjunto factible.

Letras Griegas Comúnmente Usadas como Notaciones Estadísticas										
alpha	beta	Ki al cuadrado	delta	mu	nu	pi	rho	sigma	tau	theta
α	β	χ^2	δ	μ	ν	π	ρ	σ	τ	θ

REFERENCIA BIBLIOGRÁFICAS Y CONSULTAS

- [1.] SOBOL I., "El Método de Monte Carlo", Editorial MIR
- [2.] J.M. HAMMERLEY & D.C. HANDSCOMBS, "Monte Carlo, Methods"
- [3.] GEORGE D. FISHMAN, "Conceptos y Métodos en la Simulación Digital de Eventos Discretos"
- [4.] GEOFFREY GORDON, "Simulación de Sistemas".
- [5.] JAIME BARCELÓ, "Simulación de Sistemas Discretos", Publicaciones de Ingeniería de Sistemas.
- [6.] RICARDO SANZ BRAVO, AGUSTÍN JIMÉNEZ AVELLO, MANUEL-ALONSO CASTRO, "Sistema de Control Inteligente, Modelo y Simulación", Departamento de Automática, Ingeniería Electrónica e Informática industrial. Universidad Politécnica de Madrid. 1992.
- [7.] P. KOUVELIS Y G. YU. "Robust Discrete Optimization and Its Applications." Kluwer Academic Publishers, Dordrecht, Holanda, 1997.
- [8.] ANTONIO CREUS SOLE MARCOMBO, "Simulación y Control de Procesos por Ordenador"
- [9.] ANGEL A. SARABIA, "La Teoría General de los Sistemas", Publicaciones de Ingeniería de Sistemas.
- [10.] JAVIER ARACIL, "Dinámica de Sistemas", Publicaciones de Ingeniería de Sistemas.
- [11.] DONAL R. DREW, "Dinámica de Sistemas Aplicada", Publicaciones de Ingeniería de Sistemas.
- [12.] XAVIER PI JOSEP CASANOVAS, "Software de Simulación", Facultad de Informática de la UPC, Barcelona Revista Automática e Instrumentación, Junio 1996.
- [13.] E. H. L. AARTS, J. H. M. KORST, Y P. J. M. VAN LAARHOVEN, "Simulated annealing", E. H. L. Aarts y J.K. Lenstra, editores, "Local Search in Combinatorial Optimization", páginas 91-120. John Wiley & Sons, Chichester, 1997. 1
- [14.] E. H. L. AARTS Y J. K. LENSTRA, editors, "Local Search in Combinatorial Optimization", John Wiley & Sons, Chichester, 1997. 2
- [15.] R. ALCALÁ, J. CASILLAS, O. CORDÓN, Y F. HERRERA, "Improvement to the cooperative rules methodology by using the Ant Colony System algorithm", "Mathware & Soft Computing", 8: 3, páginas 321-335, 2001. 3

- [16.] S. ALONSO, O. CORDÓN, I. FERNÁNDEZ DE VIANA, F. HERRERA, "Análisis de distintas vertientes para la paralelización de los algoritmos de Optimización basada en Colonias de Hormigas". *Actas del Segundo Congreso Español de Meta heurísticas, "Algoritmos Evolutivos y Bioinspirados (MAEB03)"*, páginas 160-167, Gijón, 2003. 4
- [17.] T. BÄCK, D. FOGEL, Z. MICHALEWICZ, editors, "Handbook of Evolutionary Computation", 1998 6
- [18.] S. BALUJA AND R. CARUANA, "Removing the genetics from the standard genetic algorithm", A. Frieditis y S. Russel, editores, "Proceedings of the Twelfth International Conference on Machine Learning (ML-95)", pages 38-46. Morgan Kaufmann Publishers, Palo Alto, CA, 1995. 7
- [19.] M. BIRATTARI, T. STÜTZLE, L. PAQUETE, Y K. VARRENTRAPP, "A racing algorithm for configuring metaheuristics", W.B. Langdon y otros, editor, *GECCO 2002: "Proceedings of the Genetic and Evolutionary Computation Conference"*, páginas 11-18. Morgan Kaufmann Publishers, San Francisco, CA, 2002. 9
- [20.] O. CORDÓN, I. FERNÁNDEZ DE VIANA, F. HERRERA, Y L. MORENO, "A new ACO model integrating evolutionary computation concepts: The Best-Worst Ant System". M. Dorigo, M. Middendorf, y T. Stützle, editores, "Abstract proceedings of ANTS2000 - From Ant Colonies to Artificial Ants: A series of International Workshops on Ant Algorithms", páginas 22-29. IRIDIA, Université Libre de Bruxelles, Belgium, 2000. 23
- [21.] O. CORDÓN, F. HERRERA, L. MORENO, "Integración de Conceptos de Computación Evolutiva en un Nuevo Modelo de Colonias de Hormigas. VIII Conferencia de la Asociación Española para la Inteligencia Artificial, (Seminario Especializado en Computacion Evolutiva)", Murcia (España), 1999, Vol. II, páginas 98-105. 24
- [22.] O. CORDÓN, F. HERRERA, T. STÜTZLE, "A Review on Ant Colony Optimization Metaheuristic: Basis, Models and New Trends". *Mathware & Soft Computing*, 9: 2-3, páginas 141-175, 2002. 25
- [23.] D. COSTA AND A. HERTZ, "Ant can colour graphs. *Journal of the Operational Research Society*", 48, páginas 295-305, 1997. 26
- [24.] G. DI CARO AND M. DORIGO. ANTNET, "Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*", 9:317-365, 1998. 29
- [25.] M. DORIGO Y G. DI CARO, "The Ant Colony Optimization meta-heuristic. En D. Corne, M. Dorigo, y F. Glover, editores, *New Ideas in Optimization*", páginas 11-32. McGraw Hill, London, UK, 1999. 30
- [26.] M. DORIGO, G. DI CARO, Y L. M. GAMBARDELLA, "Ant algorithms for discrete optimization. *Artificial Life*", 5: 2, páginas 137-172, 1999. 31

- [27.] M. DORIGO, V. MANIEZZO, Y A. COLORNI, "The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*", 26: 1, páginas 29-41, 1996. 33
- [28.] M. DORIGO Y T. STÜTZLE, "Ant Colony Optimization. MIT Press", Cambridge, MA. En prensa. 35
- [29.] M. DORIGO Y T. STÜTZLE, "The ant colony optimization metaheuristic: Algorithms, applications and advances". En F. Glover and G. Kochenberger, editores, *Handbook of Metaheuristics*, páginas 251-285. Kluwer Academic Publishers, 2003. 36
- [30.] T. A. FEO Y M. G. C. RESENDE, "Greedy randomized adaptative search procedures. *Journal of Global Optimization*", 6, páginas 109-133, 1995. 39
- [31.] P. FESTA Y M. G. C. RESENDE. GRASP, "An annotated bibliography. En P. Hansen y C. C. Ribeiro, editores, "Essays and Surveys on Metaheuristics", Kluwer Academic Publishers, 2001. 40
- [32.] R. A. FISHER, "Theory of Statistical Estimation. *Proceedings of the Cambridge Philosophical Society*", 22, páginas 700-725, 1925. 41.
- [33.] R.A. FISHER, "The Comparison of Samples with Possibly Unequal Variances, *Annals of Eugenics*", 9, páginas 174-180, 1936. 42
- [34.] L. M. GAMBARDILLA AND M. DORIGO. ANT-Q, A. Prieditis y S. Russell, editores, "Proceedings of the Twelfth International Conference on Machine Learning (ML-95)".
- [35.] M. R. GAREY Y D. S. JONSON, "Computers and Intractability: A Guide to the Theory of NPCompleteness", Freeman, San Francisco, CA, 1979. 47
- [36.] F. GLOVER AND G. KOCHENBERGER, editores. "Handbook of Metaheuristics. Kluwer Academic Publishers", 2003. 48
- [37.] F. GLOVER Y M. LAGUNA, "Tabú Search. Kluwer Academic Publishers", Boston, MA, 1997. 49
- [38.] W. J. GUTJAHR, "A Graph-based Ant System and its convergence. *Future Generation Computer Systems*", 16: 8, páginas 873-888, 2000. 54
- [39.] W. J. GUTJAHR, ACO "Information Processing Letters", 82: 3, páginas 145-153, 2002. 55
- [40.] P. HANSEN Y N. MLADENOVIC, S. Voss, S. Martello, I. H. Osman, y C. Roucairol, editores, "MetaHeuristics - Advances and Trends in Local Search Paradigms for Optimization", páginas 433-458. Kluwer Academic Publishers, Dordrecht, Holanda, 1999. 57

- [41.] J. L. HENNESY, D. A. PATTERSON, "Computer Architecture. A Quantitative Approach", páginas 40-42, Morgan Kaufmann Publishers, 2003. 58
- [42.] J. HOLLAND, "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975. 60
- [43.] NAIM A. KLAIR, "System Modeling and Simulation", Electrical Engineering and Electronics Marcel Dekker, Inc.
- [44.] L. BLUM M., SHUB M., "A Simple unpredictable pseudorandom number generator", SIAM Journal on Computing 15, pp 364-383, 1986 Blum.
- [45.] FIPS 140, "Security requirements for cryptographic modules", Federal Information Processing Standards Publication 140-1, U.S. Department of Commerce / N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.
- [46.] KALISKI JR. B. S., PhD Thesis, "Elliptic curves and cryptography; a pseudorandom bit Generator and other tools", MIT Department of Electrical Engineering and Computer Science, 1988.
- [47.] KNUTH D.E., ADDISON QESLEU READING, "The Art of Computer Programming – Seminumerical Algorithms", Vol 2, 1973.
- [48.] MAURER U.M., "A Universal Statistical Test for Random Bit Generators, Advances in Cryptology Cripto '90, LNCS 537", pp 409-420, 1991.
- [49.] MENEZES A.J. VAN OORSCHOT P.C., "HandBooj of Applied Cryptography", Vanstone S.A., CRC Press 1997.
- [50.] MICALI S., "Efficient, perfect polynomial random number generators, Journal of Cryptology 3", Schnorr pp 157-172, 1991
- [51.] SHAMIR A., "On the generation of cryptographically strong pseudorandom sequences", ACM Transaction on Computer Systems, 1, 1983.
- [52.] ALBERTO ARAGON TORRES, "Métodos Evolutivos para el aprendizaje de Redes Neuronales", Junio 2002 pag. 134 – 206.
- [53.] JOSÉ FLORENTINO AUGUSTO MEDINA JACOBO, "Un enfoque robusto a un problema de diseño de red multiproducto con incertidumbre en parámetros de entrada", Mayo 2005 pag. 134 – 206, Tesis en opción al grado de maestro en ciencias en ingeniería de sistemas.
- [54.] Belén Melián, José A. Moreno Pérez, J. Marcos Moreno Vega, "Metaheuristics: A global view" Departamento de Estadística, I.O. y Computación, Centro Superior de Informática, Universidad de La Laguna