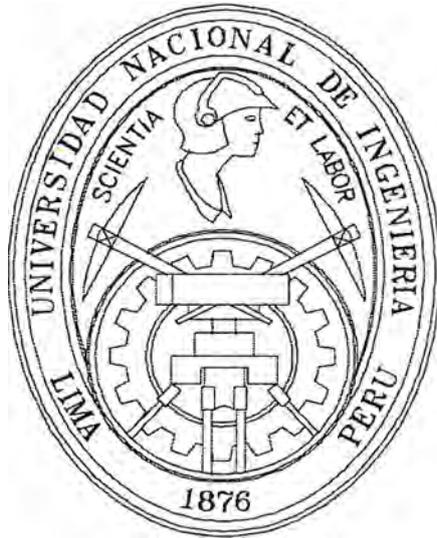


UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**“FLUJO DE POTENCIA ÓPTIMO UTILIZANDO ALGORITMOS
GENETICOS”**

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRICISTA

PRESENTADO POR:

HUBER ALONSO SALAS SANTOS

PROMOCIÓN 1999-I

LIMA-PERÚ
2000

**FLUJO DE POTENCIA ÓPTIMO UTILIZANDO ALGORITMOS
GENÉTICOS**

Dedico el presente proyecto a mis padres, que siempre me alentaron a concluir este proyecto; a mi hermana, que es la fuente de mi superación y a todas las personas que me apoyaron en la terminación de la tesis.

SUMARIO

El presente proyecto presenta un nuevo algoritmo para su utilización en la determinación del flujo de potencia óptimo en un Sistema Eléctrico de Potencia. La capacidad de este algoritmo se prueba en sistemas elementales, utilizando las restricciones de rampa, zonas prohibidas de operación en las centrales térmicas, adicionalmente se usan funciones cuadráticas para representar el costo en las centrales térmicas y para representar el caudal en las hidráulicas.

En el proceso de optimización se muestra la efectividad del uso de las matrices esparcidas para resolver el problema del flujo de potencia; asimismo se estudian los métodos para solucionar el sistema de ecuaciones lineales de la matriz jacobiana.

ÍNDICE

	Página
PRÓLOGO	01
CAPÍTULO I	
INTRODUCCIÓN	03
CAPÍTULO II	
MATRICES ESPARCIDAS	07
2.1 Definición	07
2.2 Estructuras de almacenamiento	09
2.2.1 Estructura de datos puntuales	10
2.2.2 Estructura de datos por bloques	18
2.3 Métodos iterativos aplicados a matrices esparcidas	21
2.3.1 Métodos estacionarios	21
2.3.2 Métodos no estacionarios	22
CAPÍTULO III	
ALGORITMOS GENÉTICOS	26
3.1 Fundamentos	26
3.2 La robustez de los métodos de optimización y búsqueda tradicional	27
3.3 La codificación	32

3.4	La selección	34
3.4.1	Muestreo estocástico sin reemplazo o método de la ruleta	34
3.4.2	Método del ranking	35
3.4.3	Método estocástico sobrante	35
3.4.4	Método estocástico universal	36
3.4.5	Método del torneo	36
3.5	El cruce	37
3.5.1	Cruce de n-puntos	38
3.5.2	Cruce mezclado de n-puntos	38
3.5.3	Cruce uniforme	40
3.5.4	Cruce uniforme modificado	40
3.5.5	Cruce variable	40
3.5.6	Cruce asexual	41
3.6	La mutación	41
3.6.1	Método aleatorio de inversión	41
3.6.2	Método de intercambio	41
3.7	El reemplazo	42
3.7.1	Agregar	42
3.7.2	El primero más débil	42
3.7.3	El más débil	42
3.8	Operadores adicionales	42
3.8.1	Niching	43
3.8.2	Ruido	47

CAPÍTULO IV

FLUJO DE CARGA APROVECHANDO EL CARÁCTER DE LAS MATRICES

4.1	Introducción	51
4.2	Formulación matemática del problema	53
4.2.1	Ecuaciones de comportamiento de la red	53
4.2.2	Ecuaciones de Potencia	55
4.2.3	Ecuaciones de flujo de línea	55
4.3	Evaluación de técnicas de solución	56
4.4	Evaluación de técnicas de solución	62
4.5	Representación matricial y solución del sistema de ecuaciones lineales	71
4.5.1	Selección del formato de estructura de datos	71
4.5.2	Factorización LU	73
4.5.3	Factorización incompleta	75
4.5.4	Solución iterativa del sistema de ecuaciones lineales	76

CAPÍTULO V

OPTIMIZACIÓN DE POTENCIA DE GENERACIÓN

5.1	Formulación del problema	80
5.2	Modelamiento de las pérdidas en la red	82
5.3	Límites de incremento y decremento de generación	85
5.4	Zonas de operación prohibida	87

5.5	Modelamiento de las características de las Centrales	
	Hidroeléctricas	88
5.6	Optimización mediante algoritmo genético	90
5.6.1	Optimización de la potencia hidráulica	90
5.6.2	Optimización de la potencia térmica	91
	CAPÍTULO VI	96
	METODOLOGÍA DE CÁLCULO	
6.1	Diagrama de flujo principal	95
6.2	Concepción de la metodología	96
	CAPÍTULO VII	
	RESULTADOS NUMÉRICOS	98
7.1	Sistemas de potencia utilizados y datos de centrales usados	98
7.1.1	IEEE de 9 barras	98
7.1.2	IEEE de 30 barras	100
7.1.3	Datos de centrales	103
7.1.4	Secuencia de análisis	105
7.2	Análisis de operadores	105
7.3	Análisis de resultados	171
	CAPÍTULO VIII	
	CONCLUSIONES	181
	CAPÍTULO IX	
	BIBLIOGRAFIA	184

ANEXO A	188
TÉCNICAS PARA LA GENERACIÓN DE NÚMEROS ALEATORIOS	
A.1 Introducción	188
A.2 Métodos de congruencias para generar números aleatorios	191
ANEXO B	196
MODELOS DE LOS ELEMENTOS DEL SISTEMA	
B.1 El transformador de dos devanados	196
B.2 La línea de transmisión	197
B.3 El elemento shunt	198
ANEXO C	199
MANUAL DEL PROGRAMA	
C.1 Introducción	199
C.2 Descripción de la ventana principal	200
C.3 File	200
C.4 Analysis	201
C.5 Resultados	203
ANEXO D	205
RESULTADOS DE LA OPTIMIZACIÓN HIDRÁULICA	
D.1 Resultados del LINDO	205
D.2 Resultados del FlexView	207

PRÓLOGO

La presente tesis tiene por objetivo desarrollar un flujo de potencia, donde se optimice la potencia generada, para reducir el costo de generación. Se optimiza la potencia térmica de acuerdo al costo y la potencia hidráulica de acuerdo al caudal disponible.

Los principales objetivos son:

- Mostrar el uso de las matrices esparcidas en sistemas eléctricos de potencia, específicamente en el flujo de potencia.
- Mostrar el uso de los algoritmos genéticos en los problemas de optimización de potencia activa en sistemas eléctricos de potencia.

En el capítulo uno se da una introducción al flujo de potencia, mostrando su desarrollo a través de la historia hasta llegar al flujo de potencia óptimo.

En el capítulo dos se define que es la matriz esparcida, cual es el campo de aplicación, su uso en Sistemas de Potencia, las estructuras de almacenamiento que existen y la forma de aplicación de los métodos de solución de sistemas lineales a matrices esparcidas.

En el capítulo tres se muestra la teoría de los algoritmos genéticos, donde se ve su utilización en problemas de optimización en comparación con otros métodos, luego se ve los operadores más importantes que se utilizan en los algoritmos genéticos

En el capítulo cuatro se ve la forma de realizar el flujo de potencia y la solución del sistema de ecuaciones del flujo de potencia mediante matrices esparcidas.

En el capítulo cinco se calcula los factores de penalidad y se define matemáticamente el problema de optimización de potencia activa, tanto para las centrales hidráulicas como para las centrales térmicas, una vez hecho eso, se crea un algoritmo genético apropiado para la optimización hidráulica y térmica.

En el capítulo seis se muestra el diagrama de flujo principal del flujo de potencia óptimo, en donde se usan los algoritmos desarrollados en los capítulos anteriores.

En el capítulo siete se encuentra el mejor conjunto de operadores de los algoritmos genéticos, para luego probarlos en dos redes de la IEEE considerando diversas variantes.

En el capítulo ocho se pueden ver las conclusiones a la tesis.

CAPÍTULO I INTRODUCCIÓN

El desarrollo del despacho económico, que antecedió al flujo de potencia óptimo, comenzó a inicios de la década del 20 o incluso antes; cuando dos o más unidades fueron comisionadas para tomar una carga en un sistema de potencia en el cual la capacidad de generación total excede la demanda requerida. El problema que enfrentó el operador fue exactamente como dividir la potencia activa (potencia requerida) entre las dos unidades, tal que la carga sea servida y el costo total minimizado. Específicamente, el despacho económico es un proceso computacional por el cual la generación activa total requerida es localizada entre las unidades generadoras disponibles tal que las restricciones impuestas sean satisfechas y la energía requerida en términos de Btu/h o \$/h sea minimizada. Este asume que las unidades son comisionadas y que se incluye un costo de reserva, independientemente de que vayan o no a tomar una carga adicional.

Las técnicas para solucionar el despacho económico son aún básicamente las que se desarrollaron entre los años 30 y los años 50 y son conocidas como ‘Despacho clásico de incrementales iguales’ con pérdidas o tratamiento de redes por medio de modelos aproximados conocidos como las formulas de pérdidas.

A los finales de los años 50 se comenzó a mejorar la representación de la fórmula de pérdidas. Esto ocurrió al mismo tiempo que el flujo de potencia hizo su

aparición. Un flujo de potencia es caracterizado por datos acerca de la red bajo estudio, y potencias activas (P) y reactivas(Q) inyectadas a las barras. El objeto de un flujo de potencia es determinar las tensiones y ángulos en todas las barras de la red, a partir de ahí se pueden calcular todas las otras magnitudes que caracterizan a una condición de operación.

La optimización fue aplicada a un flujo de potencia durante los años 60. Un flujo de potencia óptimo(OPF) es un flujo de potencia que minimiza el costo de combustible, las pérdidas en la red u otra función que se requiera, el OPF reconoce las limitaciones del equipamiento del sistema de potencia. Cuando el costo total de combustible es minimizado, el flujo de potencia óptimo obtiene un despacho económico y se determina la salida de potencia activa y la reactiva de los generadores, la posición de los taps y la salida de otras fuentes de reactivos.

La seguridad de la operación de los sistemas de potencia es una propiedad adicional e importante que fue alcanzada durante la década de los 70, y ha tenido una influencia profunda en el problema de despacho económico como un todo. La seguridad requiere que la operación del sistema de potencia sea confiable para las unidades generadoras, líneas de transmisión, y transformadores – primero para el sistema en condiciones normales, y segundo, bajo condiciones de contingencia, cuando uno o muchos componentes del sistema repentinamente salen de servicio. La incorporación de las restricciones de seguridad dentro del flujo de potencia óptimo (SCOPF) representa un gran reto para los investigadores de sistemas de potencia debido al gran número de variables presentes.

Uno de los principales aspectos de la seguridad del sistema es la así llamada seguridad de estado estacionario. Esta es definida como la habilidad de operación del

sistema en estado estacionario de mantenerse dentro de los límites especificados de seguridad y suplir calidad después de una contingencia, en el periodo de tiempo después de que actuaron los equipos de control rápidos y se recupero el balance del sistema.

La función objetivo en el OPF es usualmente de minimización. En algunos casos, tal como en transferencia de potencia, esta puede ser de maximización. Se designa la función objetivo como f . En nuestro caso la función objetivo será minimizar el costo total de generación.

$$f = \min \left(\sum_{\text{all gen}} F_i(P_i) \right)$$

Las ecuaciones que garantizan las restricciones de flujo de potencia se designan como

$$g(z) = 0$$

Hay que notar que solo nos ocupamos del vector variable \mathbf{z} . Este vector contiene los controles ajustables, las magnitudes de tensión, y ángulos de fase, tanto como los parámetros ajustados del sistema. Luego, se rompe las variables en conjuntos de variables de estado, variables de control, y parámetros ajustados.

El OPF puede también resolverse para una solución óptima con restricciones de desigualdad en variables dependientes, tal como flujos de MVA a través de las líneas. Estas serán designadas como

$$h^- \leq h(z) \leq h^+$$

Adicionalmente, los límites pueden ser colocados directamente en las variables de estado o en las variables de control:

$$z^- \leq z \leq z^+$$

El problema de OPF consiste entonces en minimizar (o maximizar) la función objetivo, sujeta a restricciones de igualdad, restricciones de desigualdad, y límites de las variables de control y de estado.

El flujo de potencia óptimo es un problema matemático de programación muy grande y difícil. Casi todas las aproximaciones de la programación matemática que pueden ser aplicadas a este problema han sido formuladas para desarrollar códigos de computadora que resuelvan el problema de OPF de manera eficiente y confiable. Existen varios métodos para la solución del problema de Flujo de Potencia Óptimo, entre ellos tenemos:

- Método de iteración Lambda
- Métodos del gradiente
- Método de Newton
- Método de programación lineal
- Método del punto interior
- Métodos de computación evolucionaria

Los primeros cinco métodos son conocidos, estos métodos tienen el inconveniente de necesitar funciones continuas, es decir, modelar (según el método) las condiciones reales del sistema. En el presente trabajo se utiliza la computación evolucionaria, específicamente los algoritmos genéticos, ya que este método ha mostrado ser robusto para el problema de minimizar el costo ([4], [16] y [17]).

CAPÍTULO II MATRICES ESPARCIDAS

2.1 Definición

Las matrices esparcidas son aquellas matrices que poseen un gran número de elementos nulos.

Estas matrices esparcidas son usadas en códigos de aplicación para diferencias finitas, elementos finitos o discretizaciones de volumen finito de PDEs (partial differential equations) en problemas computacionales de dinámica de fluidos, mecánica estructural, simulación de semiconductores, sistemas eléctricos de potencia y otras aplicaciones científicas y de ingeniería.

En estos casos, se debe realizar una librería que contenga maneras de almacenar y capacidad de hacer operaciones con matrices esparcidas. El diseño de la librería es basado en los siguientes principios:

Claridad: La implementación de algoritmos numéricos debe parecerse a los algoritmos matemáticos en los cuales se basan.

Reutilización: Un algoritmo en particular puede ser codificado sólo una vez, con códigos idénticos para todas las representaciones de las matrices.

Existe una colección de matrices esparcidas utilizadas en diferentes problemas de ingeniería, como se aprecia en la tabla 2.1. Esta es la colección de Harwell-

Boeing, algunas de estas matrices tienen los elementos, mientras que otras sólo tienen el esqueleto.

Tabla 2.1: Disciplinas de Aplicación

Dispersión acústica	Demografía	Flujo en redes
Control de tráfico aéreo	Economía	Análisis numérico
Astrofísica	Sistemas de potencia	Oceanografía
Biomecánica	Ingeniería Electrónica	Ingeniería petrolera
Ingeniería química	Elementos finitos	Modelamiento de reactores
Cinética química	Dinámica de fluidos	Estadística
Física de circuitos	Óptica de láser	Ingeniería estructural
Simulación de computadoras	Programación lineal	Datos de encuestas

En cuanto a las aplicaciones en la solución de problemas sistemas de potencia, en 1962 los ingenieros en Bonneville Power Administration (BPA) empezaron a usar las técnicas de matrices esparcidas. Un reporte [5] que describe su trabajo apareció en Enero de 1963, pero este no tuvo una influencia inmediata. Al mismo tiempo o un poco después otros ingenieros de sistemas de potencia, de manera independiente empezaron a usar técnicas similares [11,12]. Sin embargo, no fue hasta la aparición de otra publicación de BPA [7] en Enero de 1967 describiendo el uso de las técnicas esparcidas con el método de Newton para resolver el problema no lineal de flujo de carga, que la industria de potencia de EEUU empezó a dar al método una atención seria.

A pesar de que las técnicas de esparcidad no tienen nada matemáticamente nuevo, ellas difieren considerablemente de las operaciones matriciales por computadora. Sin embargo, es posible alcanzar ganancias de velocidad, almacenamiento y exactitud que son aproximadamente proporcionales al grado de esparcidad.

2.2 Estructuras de almacenamiento

Antes de pasar a ver las estructuras de almacenamiento, es necesario precisar la nomenclatura que se usará:

Entrada – Cualquier coeficiente de la matriz el cual es manejado explícitamente. Hay dos tipos básicos de entradas discutidos en esta tesis:

- *Punto* – Cada entrada es un valor simple escalar. Típicamente, las entradas de punto de A son simplemente los elementos no nulos de A .
- *Bloque* – Cada entrada es una matriz rectangular densa, almacenada columna por columna. Entradas de bloque son derivadas de una partición asumida de filas y columnas de la matriz A .

$NNZ(A)$ – El número de entradas punto de una matriz A .

$NZE(A)$ – El conjunto de entradas punto de una matriz A .

$NZI(A)$ – El conjunto de índices de fila que corresponden a las entradas punto de una matriz A . El ordenamiento de los elementos de $NZI(A)$ debe ser el mismo que $NZE(A)$.

$NZJ(A)$ – El conjunto de índices de columna que corresponden a las entradas punto de una matriz A . El ordenamiento de los elementos de $NZJ(A)$ debe ser el mismo que $NZE(A)$.

A_{i*} – El vector consistente de los elementos de la fila i de la matriz A .

A_{*j} – El vector consistente de los elementos de la columna j de la matriz A .

$diag_i(A)$ – El vector consistente de los elementos de una diagonal de la matriz A .

- $i < 0$ implica que la $diag_i(A)$ es un vector cuyos valores son de la $|i|$ ésima diagonal debajo de la diagonal principal.

- $i > 0$ implica que la $diag_i(A)$ es un vector cuyos valores son de la i ésima diagonal encima de la diagonal principal.
- $i = 0$ implica que la $diag_i(A)$ es la diagonal principal.

$PRF(A_{i*})$ – El conjunto de elementos de la fila i de A empezando con la primera entrada punto en la fila i hasta la última entrada punto, incluyendo los ceros.

$PRF(A_{*j})$ – El conjunto de elementos de la columna j de A empezando con la primera entrada punto en la columna j hasta la última entrada punto, incluyendo los ceros.

\hat{A} – La matriz A con una partición de filas y columnas asumidas.

$BNNZ(\hat{A})$ – El número de entradas de bloque de la matriz particionada \hat{A} .

$BNZE(\hat{A})$ – El conjunto de entradas de bloque de una matriz A . Cada entrada de bloque es una matriz rectangular densa almacenada columna por columna.

$BNZJ(\hat{A})$ – El conjunto de índices de columna del bloque correspondiente a las entradas de bloque de una matriz A . El ordenamiento de los elementos de $BNZJ(\hat{A})$ debe ser el mismo que $BNZE(\hat{A})$.

\hat{A}_{i*} – El vector bloque consistente de los elementos de la fila bloque i de la matriz particionada \hat{A} .

\hat{A}_{*j} – El vector bloque consistente de los elementos de la columna bloque j de la matriz particionada \hat{A} .

A – La matriz PA donde P es una matriz de permutación.

Las estructuras básicas, son tres, se tratarán únicamente a dos de ellas, con sus formatos respectivos:

2.2.1 Estructura de datos puntuales

En este caso se almacenan los elementos no nulos separadamente.

Matriz con almacenamiento de Coordenadas: (COOR_Matrix). Lista de elementos no nulos con sus respectivos índices de filas y columnas. Este es el formato más general de matriz esparcida, pero este no es muy eficiente, computacionalmente hablando(requiere más memoria). Este asume que no hay orden de los valores de los elementos no nulos.

Tres arreglos son requeridos para este formato:

- VAL – Un arreglo escalar de longitud $NNZ(A)$ que consiste en los $NZE(A)$, las entradas de A, están en cualquier orden
- INDX – Un arreglo de enteros de longitud $NNZ(A)$ que consiste en los $NZI(A)$, los correspondientes índices de las filas de las entradas de A.
- JNDX – Una arreglo de enteros de longitud $NNZ(A)$ que consiste en los $NZJ(A)$, los correspondientes índices de las columnas de las entradas de A.

Por ejemplo, tenemos

$$A = \begin{bmatrix} 11 & 0 & 13 & 14 & 0 \\ 0 & 0 & 23 & 24 & 0 \\ 31 & 32 & 33 & 34 & 0 \\ 0 & 42 & 0 & 44 & 0 \\ 51 & 52 & 0 & 0 & 55 \end{bmatrix},$$

entonces una representación de A en este formato sería como sigue:

$$\begin{aligned} \text{VAL} &= (11 \ 51 \ 31 \ 32 \ 34 \ 52 \ 13 \ 23 \ 33 \ 14 \ 24 \ 42 \ 55 \ 44) , \\ \text{INDX} &= (1 \ 5 \ 3 \ 3 \ 3 \ 5 \ 1 \ 2 \ 3 \ 1 \ 2 \ 4 \ 5 \ 4) , \\ \text{JNDX} &= (1 \ 1 \ 1 \ 2 \ 4 \ 2 \ 3 \ 3 \ 3 \ 4 \ 4 \ 2 \ 5 \ 4) , \end{aligned}$$

Matriz con almacenamiento Comprimido de Filas: (CRS_Matrix).

Subsecuentemente los elementos no nulos de las filas de la matriz son almacenados contiguamente en posiciones de memoria y un arreglo adicional de enteros especifica

donde empieza cada fila. Este asume que no hay ningún orden entre los valores no nulos dentro de cada fila, pero las filas son almacenadas en orden consecutivo, dependiendo de la cantidad de arreglos.

Una matriz A es almacenada en el formato CRS usando cuatro arreglos:

- VAL – Es un arreglo escalar de longitud $NNZ(A)$ que consiste en las entradas de A :

$$VAL = (NZE(A_{1*}), NZE(A_{2*}), \dots, NZE(A_{m*}))$$

- INDX – Es un arreglo de enteros de longitud $NNZ(A)$ que consiste en las entradas de A :

$$INDX = (NZI(A_{1*}), NZI(A_{2*}), \dots, NZI(A_{m*}))$$

- PNTRB – Es un arreglo de enteros de longitud m tal que cada elemento es el inicio de la siguiente fila.
- PNTRE – Es un arreglo de enteros de longitud m tal que cada elemento es el final de la fila.

Por ejemplo, la matriz anterior en formato CRS:

$$\begin{aligned} VAL &= (11 \ 13 \ 14 \ 23 \ 24 \ 31 \ 32 \ 33 \ 34 \ 42 \ 44 \ 51 \ 52 \ 55) \\ INDX &= (1 \ 3 \ 4 \ 3 \ 4 \ 1 \ 2 \ 3 \ 4 \ 2 \ 4 \ 1 \ 2 \ 5) \\ PNTRB &= (1 \ 4 \ 6 \ 10 \ 12) \\ PNTRE &= (4 \ 6 \ 10 \ 12 \ 15) \end{aligned}$$

Nota: Los dos últimos arreglos se pueden juntar en uno solo, pero en este caso las filas deben colocarse consecutivamente. Este es el formato que se utilizará.

$$PNTR = (1 \ 4 \ 6 \ 10 \ 12 \ 15) ,$$

Matriz con almacenamiento Comprimido de Columnas: (CCS_Matrix), también se le conoce como el formato *Harwell-Boeing*. Este es una variante del

almacenamiento CRS donde las columnas en vez de las filas son almacenadas contiguamente. Hay que notar que el ordenamiento CCS de A es el mismo que el CRS de A^T .

Una matriz A es almacenada en el formato CSC usando cuatro arreglos:

- VAL – Es un arreglo escalar de longitud $NNZ(A)$ que consiste en las entradas de A :

$$VAL = (NZE(A_{*1}), NZE(A_{*2}), \dots, NZE(A_{*k}))$$

- $INDX$ – Es un arreglo de enteros de longitud $NNZ(A)$ que consiste en las entradas de A :

$$INDX = (NZI(A_{*1}), NZI(A_{*2}), \dots, NZI(A_{*k}))$$

- $PNTRB$ – Es un arreglo de enteros de longitud k tal que cada elemento es el inicio de la siguiente columna.
- $PNTRE$ – Es un arreglo de enteros de longitud k tal que cada elemento es el final de la columna.

Por ejemplo, la matriz en formato CCS:

$$\begin{aligned} VAL &= (11 \ 31 \ 51 \ 32 \ 42 \ 52 \ 13 \ 23 \ 33 \ 14 \ 24 \ 34 \ 44 \ 55) \\ INDX &= (1 \ 3 \ 5 \ 3 \ 4 \ 5 \ 1 \ 2 \ 3 \ 1 \ 2 \ 3 \ 4 \ 5) \\ PNTRB &= (1 \ 4 \ 7 \ 10 \ 14) \\ PNTRE &= (4 \ 7 \ 10 \ 14 \ 15) \end{aligned}$$

Nota: Los dos últimos arreglos se pueden juntar en uno solo, pero en este caso las columnas deben colocarse consecutivamente. Este es el formato que se utilizará.

$$PNTR = (1 \ 4 \ 7 \ 10 \ 14 \ 15) ,$$

Matriz con almacenamiento Diagonal Comprimido: (CDS_Matrix). Las sub y super diagonales son almacenadas en posiciones de memoria contigua.

Sea $LDA \geq \min(m, k)$, y NDIAG denota el número de diagonales no cero de A. Dos arreglos son requeridos para el formato CDS:

- VAL – Un arreglo escalar de dos dimensiones LDA por NDIAG consiste en las NDIAG diagonales no nulas de A en cualquier orden.
- IDIAG – Un arreglo entero de longitud NDIAG que consiste en los índices de las diagonales no nulas de A en VAL. Así, si $IDIAG(j)=i$ entonces la j ésima columna de VAL contiene $diag_i(A)$.

Por ejemplo, sea

$$A = \begin{bmatrix} 11 & 0 & 13 & 0 & 0 \\ 21 & 0 & 0 & 24 & 0 \\ 31 & 32 & 33 & 0 & 35 \\ 0 & 42 & 0 & 44 & 0 \\ 0 & 0 & 53 & 0 & 55 \end{bmatrix},$$

entonces A será almacenada en el formato DIA como sigue

$$VAL = \begin{bmatrix} * & * & 11 & 13 \\ * & 21 & 0 & 24 \\ 31 & 32 & 33 & 35 \\ 42 & 0 & 44 & * \\ 53 & 0 & 55 & * \end{bmatrix},$$

$$IDIAG = (-2 \ -1 \ 0 \ 2) ,$$

Los ‘*’ significan elementos que no pertenecen a la matriz.

Almacenamiento Ellpack/Itpack: (ELL_Matrix). Es un tipo de almacenamiento por filas en forma ordenada.

Sea MAXNZ el número máximo de entradas por cada fila de A, esto es,

$$MAXNZ(A) = \max_{1 \leq i \leq m} (NNZ(A_{i*}))$$

y sea $LDA \geq \min(m, k)$. Una matriz A es almacenada en el formato ELL usando dos arreglos:

- **VAL** – Un arreglo escalar de dos dimensiones LDA por $MAXNZ$ tal que la fila i de **VAL** consiste en $NZE(A_{i*})$ es menor que $MAXNZ$.
- **INDX** – Un arreglo de enteros de dos dimensiones LDA por $MAXNZ$ tal que ahora la fila i de **INDX** consiste en el valor entero de la columna del $NZE(A_{i*})$, el resto se rellena con el valor i .

Por ejemplo, sea

$$A = \begin{bmatrix} 11 & 0 & 13 & 14 & 0 \\ 0 & 0 & 23 & 24 & 0 \\ 31 & 32 & 33 & 34 & 0 \\ 0 & 42 & 0 & 44 & 0 \\ 51 & 52 & 0 & 0 & 55 \end{bmatrix},$$

su representación formato ELL sería:

$$VAL = \begin{bmatrix} 11 & 13 & 14 & 0 \\ 23 & 24 & 0 & 0 \\ 31 & 32 & 33 & 34 \\ 42 & 44 & 0 & 0 \\ 51 & 52 & 55 & 0 \end{bmatrix},$$

$$INDX = \begin{bmatrix} 1 & 3 & 4 & 1 \\ 3 & 4 & 2 & 2 \\ 1 & 2 & 3 & 4 \\ 2 & 4 & 4 & 4 \\ 1 & 2 & 5 & 5 \end{bmatrix},$$

Almacenamiento Diagonal Cortado: (JDS_Matrix). Más eficiente que CDS, aunque tiene un costo de una operación de reunir/dispersar.

El formato JDS requiere la especificación de una matriz de permutación P la cual permuta las filas de A en orden descendiente de $NNZ(A_{i*})$. Esta matriz tiene la siguiente característica $P^T=P^{-1}$ y es representada por el vector de enteros $IPERM$. Sea A la matriz permutada y sea i' la fila de $A = PA$ correspondiente a la fila i de A . Se necesitan cuatro arreglos:

- **VAL** – Un arreglo escalar de longitud $NNZ(A)$ que consiste de las entradas de A . El primer elemento de **VAL** es el primer elemento de $NZE(A_{1*})$ seguido por el primer elemento de $NZE(A_{2*})$ y así sucesivamente hasta el primer valor de $NZE(A_{m'*})$ donde m' es la última fila de A que tiene entradas no nulas (usualmente $m'=m$). El $m'+1$ ésimo elemento de **VAL** es el segundo elemento de $NZE(A_{1*})$ y así sucesivamente.
- **INDX** – Un arreglo de enteros de longitud $NNZ(A)$ que consiste en los índices de las columnas de las entradas de A .
- **PNTR** – Un arreglo de enteros de longitud $MAXNZ+1$, donde $MAXNZ$ esta definido como en el formato **ELL**, tal que cada elemento indica la posición en **VAL** donde se inicia la siguiente columna.
- **IPERM** – Un arreglo de enteros de longitud m tal que $i=IPERM(i')$. Hay que notar que **IPERM** representa P^T .

Por ejemplo, una permutación apropiada (y su inversa) para la matriz usada anteriormente sería:

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad P^T = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

y \bar{A} sería:

$$\bar{A} = \begin{bmatrix} 31 & 32 & 33 & 34 & 0 \\ 11 & 0 & 13 & 14 & 0 \\ 51 & 52 & 0 & 0 & 55 \\ 0 & 0 & 23 & 24 & 0 \\ 0 & 42 & 0 & 44 & 0 \end{bmatrix},$$

Los elementos de los arreglos para el formato JDS serían:

$$\begin{aligned} \text{VAL} &= (31 \ 11 \ 51 \ 23 \ 42 \ 32 \ 13 \ 52 \ 24 \ 44 \ 33 \ 14 \ 55 \ 34) , \\ \text{INDX} &= (1 \ 1 \ 1 \ 3 \ 2 \ 2 \ 3 \ 2 \ 4 \ 4 \ 3 \ 4 \ 5 \ 4) , \\ \text{PNTR} &= (1 \ 6 \ 11 \ 14 \ 15) , \\ \text{IPERM} &= (3 \ 1 \ 5 \ 2 \ 4) , \end{aligned}$$

Almacenamiento Skyline: (SKS_Matrix). Este formato sirve para almacenar una matriz cuya primera fila o columna esta llena de elementos no nulos. En este formato se necesitan dos arreglos y se tiene dos tipos, un almacenamiento por columnas y otro por filas. En el caso de filas tenemos:

- VAL – Es un arreglo escalar que consiste en:

$$\text{VAL} = (\text{PRF}(A_{1*}), \text{PRF}(A_{2*}), \dots, \text{PRF}(A_{m*}))$$

- PNTR – Es un arreglo de enteros de longitud $m+1$, tal que cada elemento de PNTR indica la posición del inicio de la siguiente fila en VAL, en el caso de que una fila de A no tenga elementos no nulos, el valor del elemento de PNTR es igual al siguiente $\text{PNTR}(i)=\text{PNTR}(i+1)$.

En el caso de columnas es similar. Por ejemplo, si se desea hacer un almacenamiento por filas, sea la matriz A:

$$A = \begin{bmatrix} 11 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 31 & 32 & 33 & 0 & 0 \\ 0 & 42 & 0 & 44 & 0 \\ 51 & 52 & 0 & 0 & 55 \end{bmatrix},$$

luego A en formato SKY sería:

$$\begin{aligned} \text{VAL} &= (11 \ 31 \ 32 \ 33 \ 42 \ 0 \ 44 \ 51 \ 52 \ 0 \ 0 \ 5) , \\ \text{PNTR} &= (1 \ 2 \ 2 \ 5 \ 8 \ 13) , \end{aligned}$$

2.2.2 Estructura de datos por bloques

Cada una de las estructuras mencionadas anteriormente, pueden entrar en forma de bloques, donde cada entrada de la matriz es un LB por LB bloque denso. Sistemas de este tipo se usan cuando hay elementos desconocidos (no se sabe si serán nulos o no) de una ecuación diferencial parcial discretizada. Explotando esta propiedad, se puede reducir la cantidad de almacenamiento entero, incrementar la performance de los núcleos y permitir algoritmos numéricos para realizar análisis en la entrada de bloques. Típicamente LB es un número pequeño, menor que 10, determinado por el número de cantidades medidas en cada punto de la matriz, ej. velocidad, presión, viscosidad, etc. Como fuere, hay casos donde LB puede ser del orden de 100 como es el caso de complejos flujos de reactivos químicos.

Hay dos maneras para ordenar los elementos del bloque en memoria. Uno es ordenar los elementos de cada bloque contiguamente, ej., si LB=2 luego, para cada bloque, nosotros almacenamos el elemento (1,1) seguido por el elemento (2,1) seguido por (1,2) y (2,2). Así, cada bloque es almacenado en LB² posiciones

contiguas de memoria. Otra forma es almacenar todos los elementos (1,1) contiguamente seguidos por todos los elementos (2,1) y así sucesivamente.

A continuación se va a realizar el formato BRS(block row storage), que es análogo al formato CRS.

Sea $l=LB$ la dimensión de los bloques de entrada de A , \hat{A} denota A con una partición de sus filas y columnas y m_b y k_b las dimensiones de los bloques (filas y columnas respectivamente). \hat{A} es almacenada en el formato BRS usando cuatro arreglos:

- VAL – Un arreglo de dimensión $(l, l, BNNZ(\hat{A}))$ que consiste en los bloques de entrada de \hat{A} :

$$VAL = (BNZE(\hat{A}_{1*}), BNZE(\hat{A}_{2*}), \dots, BNZE(\hat{A}_{m_b*}))$$

- BINDX – Un arreglo de enteros de longitud $BNNZ(\hat{A})$ que consiste en los índices de los bloques de entrada de \hat{A}

$$VAL = (BNZJ(\hat{A}_{1*}), BNZJ(\hat{A}_{2*}), \dots, BNZJ(\hat{A}_{m_b*}))$$

- BPNTRB – Un arreglo de enteros de longitud m_b tal que $BPNTRB(i)$ indica la posición en BINDX del inicio de la fila.
- BPNTRE – Un arreglo de enteros de longitud m_b tal que $BPNTRE(i)$ indica la posición en BINDX del final de la fila.

Por ejemplo, sea

$$\hat{A} = \left[\begin{array}{cc|cc|cc} 11 & 12 & 0 & 0 & 15 & 16 \\ 21 & 22 & 0 & 0 & 25 & 26 \\ \hline 0 & 0 & 33 & 0 & 35 & 36 \\ 0 & 0 & 43 & 44 & 45 & 46 \\ \hline 51 & 52 & 0 & 0 & 0 & 0 \\ 61 & 62 & 0 & 0 & 0 & 0 \end{array} \right]$$

entonces $l=2$, \hat{A} será almacenada en formato BRS como sigue:

$$VAL(1:2,1:2,1) = \begin{pmatrix} 11 & 12 \\ 21 & 22 \end{pmatrix}, \quad VAL(1:2,1:2,2) = \begin{pmatrix} 15 & 16 \\ 25 & 26 \end{pmatrix},$$

$$VAL(1:2,1:2,3) = \begin{pmatrix} 33 & 0 \\ 43 & 44 \end{pmatrix}, \quad VAL(1:2,1:2,4) = \begin{pmatrix} 35 & 36 \\ 45 & 46 \end{pmatrix},$$

$$VAL(1:2,1:2,5) = \begin{pmatrix} 51 & 52 \\ 61 & 62 \end{pmatrix},$$

Ahora bien, VAL se puede almacenar de dos maneras, colocando los elementos de cada bloque consecutivamente o tomando los primeros elementos de cada bloque, luego los segundos y así sucesivamente.

$$VAL = (11, 21, 12, 22, 15, 25, 16, 26, 33, 43, 0, 44, 35, 45, 36, 46, 51, 61, 52, 62)$$

o

$$VAL = (11, 15, 33, 35, 51, 21, 25, 43, 45, 61, 12, 16, 0, 36, 52, 22, 26, 44, 46, 62)$$

El resto de arreglos quedan como sigue:

$$\begin{aligned} BINDX &= (1 \ 3 \ 2 \ 3 \ 1) , \\ BPNTRB &= (1 \ 3 \ 5) , \\ BPNTRE &= (3 \ 5 \ 6) , \end{aligned}$$

Como en el formato CRS los dos últimos arreglos se pueden reducir a uno:

$$BPNTR = (1 \ 3 \ 5 \ 6)$$

2.3 Métodos iterativos aplicados a matrices esparcidas

El término “método iterativo” se refiere a un amplio rango de técnicas que usan aproximaciones sucesivas para obtener soluciones más precisas de un sistema lineal en cada paso. Principalmente se tienen dos tipos de métodos iterativos. Los métodos estacionarios son viejos, fáciles de comprender y de implementar, pero usualmente no son efectivos. Los métodos no estacionarios que son un desarrollo relativamente reciente; su análisis es usualmente más difícil de comprender, pero ellos pueden ser altamente efectivos.

Los métodos no estacionarios se basan en la idea de una secuencia de vectores ortogonales. (Una excepción es el método de Chebyshev, el cual se basa en polinomios ortogonales).

La tasa de convergencia de estos métodos depende marcadamente del espectro de la matriz de coeficientes. Así, los métodos iterativos usualmente envuelven una segunda matriz que transforma la matriz de coeficientes a una con un espectro más favorable. La matriz de transformación es llamada *precondicionador*. Un buen precondicionador mejora la convergencia del método iterativo, suficientemente para superar el costo de construir y aplicar el precondicionador. La convergencia del método iterativo puede fallar sin un precondicionador.

A continuación se describe brevemente cada uno de los métodos con sus respectivos tipos.

2.3.1 Métodos estacionarios

- Jacobi

El método de Jacobi está basado en resolver para cada variable localmente con respecto a otras variables; una iteración del método corresponde a resolver una

variable cada vez. El método resultante es fácil de comprender y de implementar, pero tiene convergencia lenta.

- Gauss-Seidel

El método de Gauss-Seidel es como el método de Jacobi, excepto que usa los valores actualizados tan pronto como ellos estén disponibles. En general, si el método de Jacobi converge, el método de Gauss-Seidel convergerá más rápido que el de Jacobi, a pesar de tener una convergencia relativamente lenta.

- SOR(Sobrerelajación Sucesiva)

El método SOR puede ser derivado del método de Gauss-Seidel al introducir un parámetro de extrapolación ω . Una elección adecuada de ω , hace que SOR converja más rápido que el método de Gauss-Seidel por un orden de magnitud.

- SSOR(Sobrerelajación Sucesiva Simétrica)

El método SSOR no tiene ventaja sobre el SOR por si solo; sin embargo es muy usado con un preconditionador para métodos no estacionarios.

2.3.2 Métodos no estacionarios

- Gradiente Conjugado(CG)

El método del gradiente conjugado deriva su nombre del hecho de que este genera una secuencia de vectores conjugados (ortogonales). Estos vectores son los residuales de las iteraciones. Ellos son también los gradientes de una función cuadrática, la minimización de esta es el equivalente a resolver el sistema lineal. CG es un método extremadamente efectivo cuando la matriz de coeficientes es simétrica definida positiva, ya que solo se requiere el almacenamiento para un determinado número de vectores.

- Mínimo residual (MINRES) y simétrico LQ (SYMMLQ)

Estos métodos son alternativas computacionales para el CG en las matrices de coeficientes que son simétricas, pero posiblemente indefinidas. SYMMLQ generaría la misma solución, itera como CG si la matriz de coeficientes es simétrica definida positiva.

- Gradiente Conjugado en Ecuaciones Normales: CGNE y CGNR

Estos métodos son basados en la aplicación del método CG a una de las dos formas de las *ecuaciones normales* para $Ax=b$. CGNE resuelve el sistema $(AA^T)y = \tilde{b}$ para y para luego computar la solución $x=A^Ty$. CGNR resuelve $(A^TA)x=y$ para la solución del vector x donde $\tilde{b} = A^Tb$. Cuando la matriz de coeficientes A es no simétrica y no singular, las matrices de las ecuaciones normales AA^T y A^TA serán simétricas y definidas positivas, y así CG puede ser aplicado. La convergencia puede ser lenta, ya que el espectro de las matrices de las ecuaciones normales será menos favorable que el espectro de A .

- Mínimo residual generalizado (GMRES)

El método GMRES calcula una secuencia de vectores ortogonales (como MINRES), y combina estos resolviendo mínimos cuadrados y actualizándolos. Sin embargo, a diferencia del MINRES (y CG) este requiere almacenamiento de toda la secuencia, así que una gran cantidad de almacenamiento es necesitada. Por esta razón, versiones reiniciadas de este método son usadas. En las versiones reiniciadas, los costos de almacenamiento y cálculo son limitados al especificar un número de vectores a ser generado. Este método es muy usado en general para matrices no simétricas.

- Gradiente Biconjugado (BiCG)

El método del gradiente biconjugado genera dos CG como secuencia de vectores, uno basado en un sistema con la matriz de coeficientes original A , y uno con A^T . En vez de ortogonalizar cada secuencia, ellos son hechos mutuamente ortogonales. Este método, tal como CG, usa almacenamiento limitado. Este es muy usado cuando la matriz es no simétrica y no singular; sin embargo, la convergencia puede ser irregular, y hay la posibilidad de que el método falle. BiCG requiere una multiplicación con la matriz de coeficientes y con su transpuesta en cada iteración.

- Cuasi-Mínimo Residual (QMR)

El método cuasi-mínimo residual aplica una solución de mínimos cuadrados y actualiza los residuales BiCG, mientras suaviza el comportamiento de convergencia irregular de BiCG, el cual puede guiar a aproximaciones más confiables. Este método tiene una mirada hacia delante que evita que falle BiCG. Incluso sin la mirada hacia delante, QMR evita grandemente la falla que puede ocurrir en BiCG. Por otro lado, QMR no efectúa una verdadera minimización del error o del residual, mientras este converge suavemente, este a veces no mejora el BiCG en términos del número de pasos de iteración.

- Gradiente Conjugado Cuadrado (CGS)

El método del gradiente conjugado cuadrado es una variante de BiCG que aplica y actualiza operaciones para la secuencia A y la secuencia A^T ambas a los mismos vectores. Idealmente, este haría doble la tasa de convergencia, pero en la práctica la convergencia puede ser mucho más irregular que para el BiCG, el cual a veces guía resultados no confiables. Una ventaja práctica es que el método no necesita la multiplicación con la transpuesta de la matriz de coeficientes.

- Gradiente biconjugado estabilizado (Bi-CGSTAB)

El gradiente biconjugado estabilizado es una variante del BiCG, como CGS, pero usando diferentes actualizaciones para la secuencia de A^T con el fin de obtener una convergencia más suave que CGS.

- Iteración de Chebyshev

La iteración de Chebyshev recursivamente determina polinomios con coeficientes escogidos para minimizar la norma del residuo en un sentido de max-min. La matriz de coeficientes debe ser definida positiva y un conocimiento de los eigenvalores extremos es requerido. Este método tiene la ventaja de no requerir productos internos.

CAPÍTULO III ALGORITMOS GENÉTICOS

3.1 Fundamentos

Los algoritmos genéticos son algoritmos de búsqueda basados en mecanismos de selección natural y genética natural. Este algoritmo simula “la sobrevivencia del más fuerte” de la teoría de evolución de Darwin. En cada generación, un nuevo conjunto de criaturas artificiales (cadenas) es creado usando bits o piezas de los más fuertes de la generación anterior. Los algoritmos genéticos explotan la información histórica para especular en nuevos puntos de búsqueda con un mejor desempeño.

Los algoritmos genéticos fueron desarrollados por John Holland, sus colegas, y sus estudiantes en la Universidad de Michigan. Las metas de su investigación fueron dos: (1) abstraer y explicar rigurosamente el proceso adaptivo de los sistemas naturales, y (2) diseñar software de sistemas artificiales que retengan los mecanismos importantes de los sistemas naturales. Esta aproximación ha guiado a importantes descubrimientos en los campos de la ciencia de sistemas naturales y artificiales.

El tema central de la investigación de algoritmos genéticos ha sido la robustez, el balance entre la eficiencia y la eficacia necesaria para sobrevivir en diferentes ambientes. Las implicaciones de la robustez para sistemas artificiales son numerosas. Si los sistemas artificiales pueden ser hechos más robustos, rediseños costosos pueden ser reducidos o eliminados. Si se alcanzan niveles más altos de adaptación

los sistemas existentes pueden realizar funciones más grandes y mejores. Diseñadores de sistemas artificiales –ambos software y hardware- pueden maravillarse con la robustez, la eficiencia y la flexibilidad de los sistemas biológicos. Los sistemas biológicos poseen características de auto-reparación, auto-guía, y reproducción, mientras que ellas apenas existen en los más sofisticados sistemas artificiales.

3.2 La robustez de los métodos de optimización y búsqueda tradicionales

Ahora surge la pregunta si los métodos convencionales de búsqueda cumplen con nuestros requerimientos de robustez. La literatura actual identifica tres tipos principales de métodos: los basados en cálculo, los enumerativos, y los aleatorios. Se examinará cada tipo para ver que conclusión puede ser hecha sin una prueba formal.

Los métodos basados en cálculo han sido bastante estudiados. Y se subdividen en dos clases principalmente: directos e indirectos.

Los métodos indirectos buscan un extremo local resolviendo usualmente un conjunto de ecuaciones no lineales igualando el gradiente de la función objetivo a cero. Esta es la generalización multidimensional de la noción de cálculo elemental de puntos extremos, como se ilustra en la figura 3.1. Dada una función suave, sin restricciones, encontrar un posible pico empieza restringiendo la búsqueda a aquellos puntos con pendientes de cero en todas las direcciones.

Por otro lado los métodos de búsqueda directos buscan el óptimo moviéndose en una dirección relativa al gradiente local. Esto es la simple noción de *subir la colina*: encontrar el óptimo local moviéndose en la dirección de máxima pendiente.

Ambos métodos son locales en el espacio, estos métodos han sido mejorados, extendidos, y desmenuzados, con un razonamiento simple se muestra su pobre robustez.

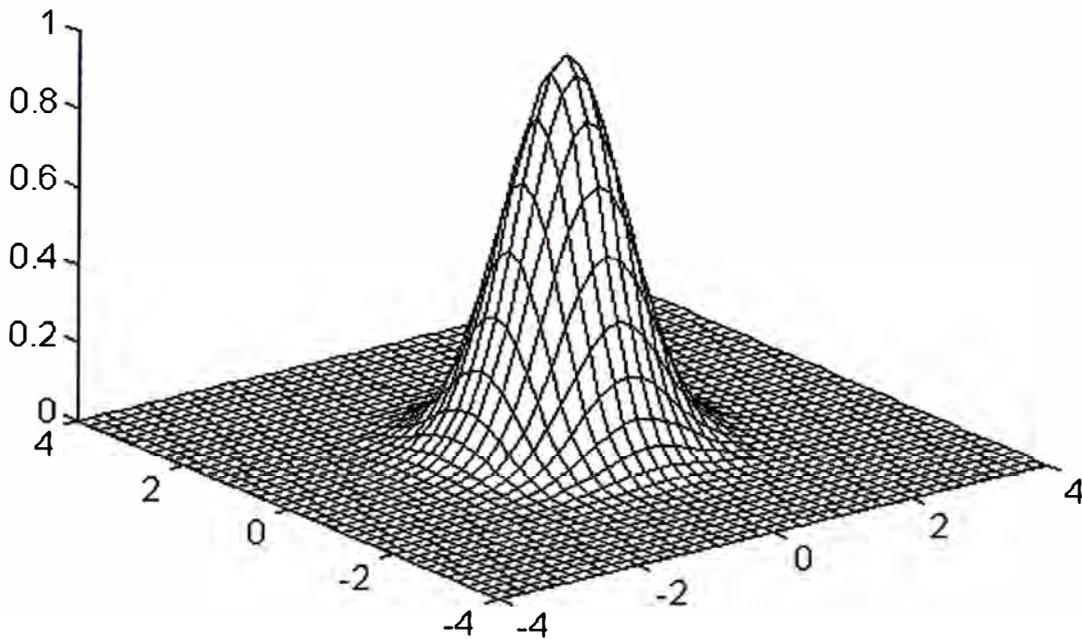


Fig. 3.1 La función de un pico es fácil para métodos basados en el cálculo

Primero, ambos métodos son de alcance local; la búsqueda es del mejor en una vecindad del punto actual. Por ejemplo, en la figura 3.2. si se empieza la búsqueda en una vecindad del pico más bajo puede presentarse la pérdida del principal pico (pico más alto). Además, una vez que el pico más bajo es alcanzado, mejoras adicionales deben ser vistas a través de una búsqueda aleatoria u otro truco.

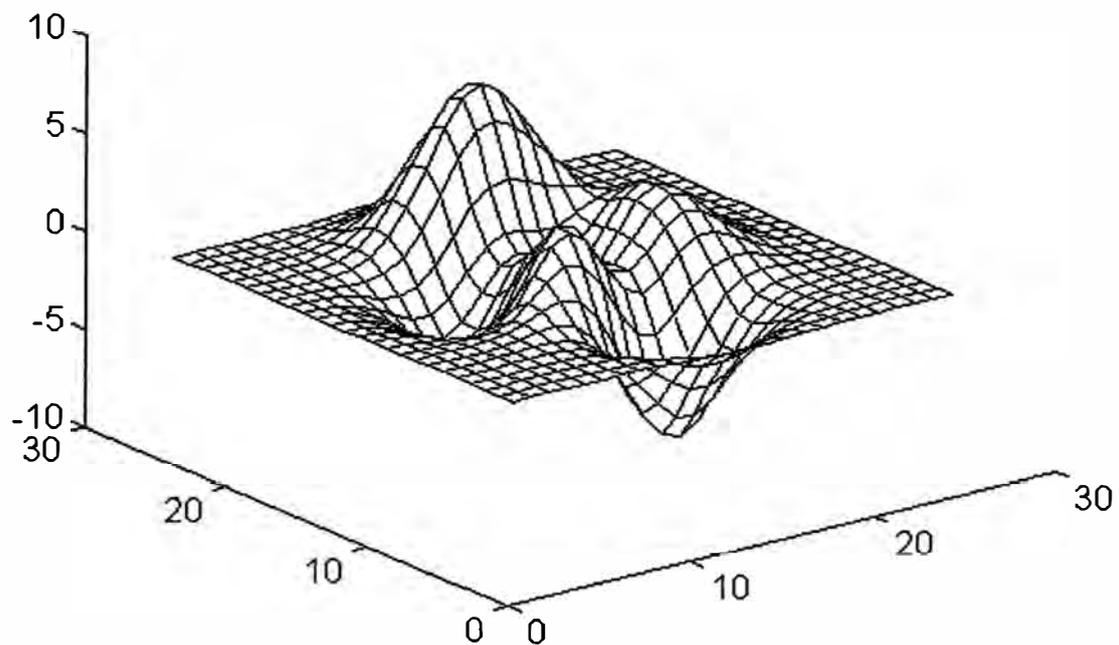


Fig. 3.2 Una función de múltiples picos

Segundo, los métodos basados en el cálculo dependen de la existencia de la derivada en todos los puntos. En muchos casos se realiza aproximaciones a funciones objetivo cuadráticas con restricciones ideales con derivadas siempre presentes. El mundo real está cargado con discontinuidades, un espacio de búsqueda con ruido, multimodal como se ve en la figura 3.3. Se ve que estos métodos se aplican a un limitado número de problemas. Ellos son insuficientemente robustos en dominios imprevistos.

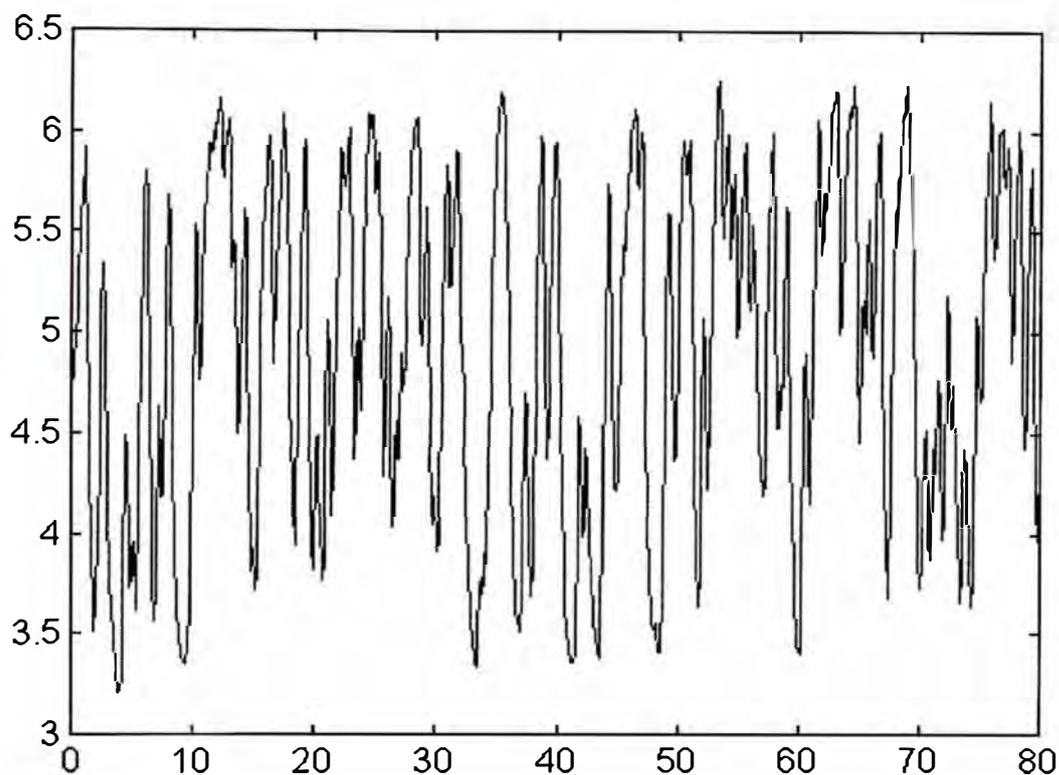


Fig. 3.3 Función ruidosa y discontinua, no adecuada para los métodos tradicionales

Los esquemas enumerativos han sido considerados en muchas formas y tamaños. La idea es bastante simple; dentro de un espacio de búsqueda finito, o un espacio de búsqueda infinito discretizado, el algoritmo de la búsqueda empieza mirando valores de función de objetivo a cada punto en el espacio, uno a la vez. A pesar de la simplicidad de este tipo de algoritmo es atractivo, y la enumeración es un tipo de búsqueda muy humano, tales esquemas deben finalmente ser descartados, por su baja eficiencia. Muchos espacios prácticos son simplemente tan grandes que la búsqueda de uno a la vez es muy lenta, a pesar que tienen una posibilidad de usar la información con un fin práctico. Incluso el muy usado método de *programación dinámica* falla en problemas de un tamaño moderado y una cierta complejidad.

Los algoritmos de búsqueda aleatorios han alcanzado una popularidad por los investigadores que han reconocido las limitaciones de los esquemas basados en el cálculo y los enumerativos. Todavía, los caminos y esquemas aleatorios que investigan y guardan el mejor deben desconsiderarse debido al requisito de eficiencia. Investigadores en estos métodos, pueden esperar que a la larga no sea mejor que los esquemas enumerativos. En nuestra prisa por descartar los métodos de búsqueda estrictamente aleatorios, se debe tener cuidado de separarlos de las técnicas aleatorias. El algoritmo genético es un ejemplo de un procedimiento de búsqueda que usa una técnica aleatoria como una herramienta para guiar una búsqueda altamente explotadora a través de una codificación de un parámetro del espacio. El asunto importante es reconocer en esta coyuntura que la búsqueda aleatoria no implica necesariamente búsqueda de las direcciones.

A pesar de que esta discusión no ha sido un examen exhaustivo de los métodos de optimización tradicional, se puede establecer una conclusión: los métodos de la búsqueda convencionales no son robustos. Esto no implica que ellos no son útiles. Los esquemas mencionados se han usado en combinaciones híbridas innumerables y permutaciones con éxito en muchas aplicaciones; sin embargo, este punto en perspectiva es mejor (Fig. 3.4). En la figura se plotea un índice de efectividad de un problema para un esquema especializado, un esquema del enumerativo, y un esquema robusto idealizado.

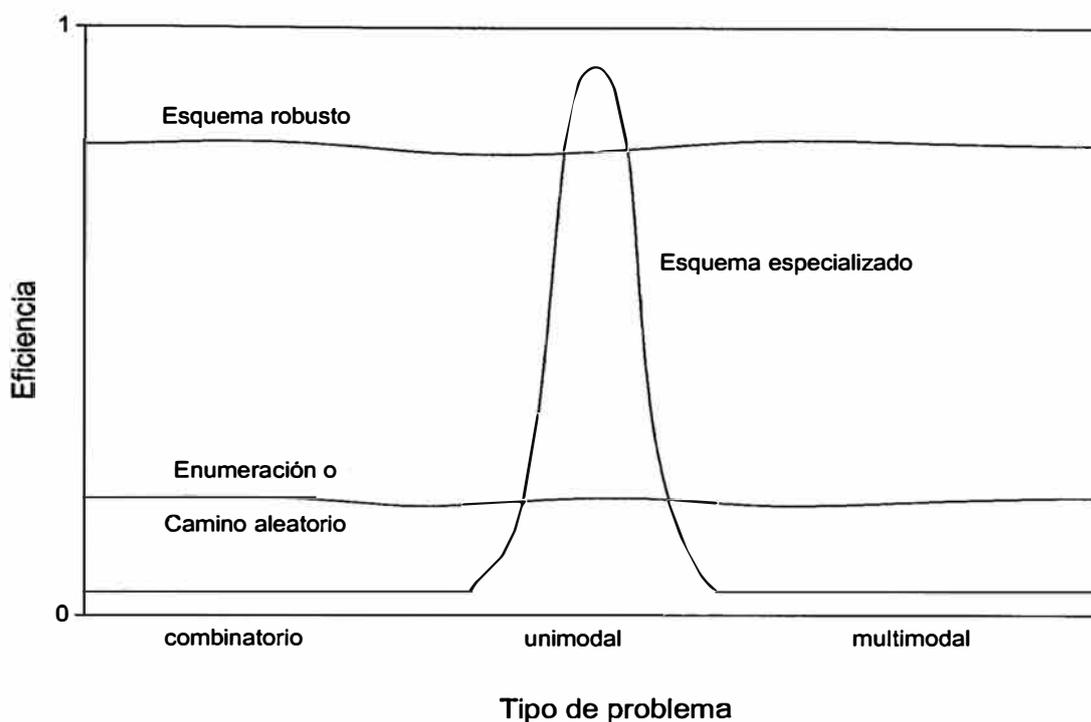


Fig. 3.4 Eficiencia de los métodos

3.3 La codificación

La codificación es el proceso de llevar nuestras variables del problema a una cadena para ser utilizada en el algoritmo genético. Por ejemplo nosotros podemos codificar un parámetro x de la siguiente manera, tenemos una cadena $A = a_1 a_{l-1} \dots a_2 a_1$, el parámetro decodificado sería $x = \sum a_i 2^{i-1}$. A pesar de que esta codificación ha dado alguna flexibilidad, no provee la variedad de opciones que nosotros requerimos para agarrar el espectro de problemas que encontramos en ciencia, negocios, e ingeniería.

Por un lado codificar un problema para una búsqueda genética no es un problema porque el programador de algoritmo genético está limitado por su imaginación, y por otro lado esta libertad de elección es una mezcla que favorece al nuevo usuario, el arreglo de posibles codificaciones alternativas es a la vez fortalecedora y desconcertante.

Hay dos principios básicos para escoger una codificación GA: *el principio de ladrillos con sentido y el principio de mínimos alfabetos.*

El principio de ladrillos con sentido es el siguiente:

El usuario debe seleccionar una codificación pequeña, los esquemas de bajo orden son pertinentes al problema subyacente y relativamente no relacionados a los esquemas con otras posiciones fijas.

La segunda regla de codificación es el principio de mínimos alfabetos, que dice:

El usuario debe seleccionar el más pequeño alfabeto que permita una natural expresión del problema.

En la tabla 3.1, se ve cuatro cadenas binarias con su valor de peso (el cual se obtuvo al decodificar las cadenas con enteros binarios sin signo y evaluando el peso de acuerdo a la relación $f(x)=x^2$). En la tabla, se considera una codificación no binaria. Considerar un mapeo uno a uno de los enteros binarios [0,31] a las 32 letras del alfabeto que consiste de las 26 letras {A-Z} y los seis dígitos {1-6}, como se muestra en la tabla 3.2.

Tabla 3.1 Comparación de poblaciones de cadenas binarias y no binarias

Cadena binaria	Valor X	Cadena no binaria	Peso
0 1 1 0 1	13	N	169
1 1 0 0 0	24	Y	576
0 1 0 0 0	8	I	64
1 0 0 1 1	19	T	361

Tabla 3.2 Correspondencia entre el código binario y no binario

Binaria	No binaria
0 0 0 0 0	A
0 0 0 0 1	B
.	.
.	.
.	.
1 1 0 0 1	Z
1 1 0 1 0	1
1 1 0 1 1	2
.	.
.	.
.	.
1 1 1 1 1	6

3.4 La selección

Un algoritmo genético simple, que da buenos resultados en muchos problemas prácticos, está compuesto de tres operadores: la reproducción, el cruce y la mutación.

La selección es el primer paso en el algoritmo genético y es un proceso en el cual se selecciona los individuos a ser copiados de acuerdo a su peso (fitness). En poblaciones naturales el peso es determinado por la habilidad de una criatura a sobrevivir a los predadores, pestilencia, y otros obstáculos y alcanzar la madurez. En nuestras poblaciones artificiales, la función objetivo se relaciona con el peso y éste es el que decide si la cadena vive o muere.

La selección puede ser implementada a manera de algoritmo de varias formas, en nuestro caso utilizaremos los siguientes tipos:

3.4.1 Muestreo estocástico sin reemplazo o método de la ruleta

Este es el método más sencillo y más común. En este método se asigna una probabilidad de selección $p_s(i)$, de acuerdo a la razón del peso del miembro i con la suma de los pesos de los miembros de la población:

$$p_s(i) = \frac{f(i)}{\sum_{j=1}^n f(j)}$$

En el programa se escoge un valor aleatoriamente de 0 a la suma de pesos y de ahí se va sumando peso a peso hasta que el valor aleatorio sea menor que esta suma. El miembro que supera este valor, se le escoge para que pase al cruce.

3.4.2 Método del ranking

Primero se ordena la población de acuerdo a su peso. Una vez que la población es rankeada, hay varias maneras de usar esta información. Aquí, se usa el ranking lineal (Baker 1975) el cual da la siguiente fórmula de probabilidad:

$$p(i) = \frac{1}{\text{popsize}} * \left(\text{nu_max} - (\text{nu_max} - \text{nu_min}) * \frac{i-1}{\text{popsize} - 1} \right)$$

donde $1 \leq \text{nu_max} \leq 2$, y $\text{nu_min} = 2 - \text{nu_max}$.

Se ve en este caso que la probabilidad de ser seleccionado se uniformiza más que en el caso del método de la ruleta.

3.4.3 Método estocástico sobrante

En este método se crea una fracción para cada hijo y de acuerdo a esta, se va realizando la selección de los padres.

Para calcular la fracción se toma la parte decimal de la división del peso del individuo con respecto al promedio:

$$\text{expected}(i) = \frac{\text{fitness}(i)}{\text{avg}}$$

Los valores de la fracción son menores que 1, luego un mecanismo de tiro de la moneda es empleado, para ver si es seleccionado o no el individuo.

3.4.4 Método estocástico universal

Este método trabaja como sigue. Llamando al método de ruleta con n slots, cada uno de tamaño $p_s(i)$, donde la circunferencia total de la rueda es 1. SUS simula la misma rueda. Sin embargo, en vez de girar la rueda n veces, con un simple puntero indica el ganador de cada vuelta, SUS gira solo una vez, usando n punteros uniformemente espaciados afuera de la rueda.

3.4.5 Método del torneo

En este caso se elige una cantidad de miembros (tourneysize), que entran a un torneo, en este torneo el que tenga mayor peso es el que se escoge.

Existen métodos de escalamiento, que se aplican mayormente al método de la ruleta, para evitar la convergencia prematura. Los más importantes métodos de escalamiento son los siguientes:

1. *Escalamiento lineal*

En el escalamiento lineal simplemente se calcula el peso escalado f' del peso inicial f usando una ecuación lineal de la forma:

$$f' = af + b$$

En esta ecuación los coeficientes a y b son usualmente escogidos para hacer dos cosas: forzar la igualdad del peso promedio escalado e inicial, y causar que el peso escalado máximo sea un múltiplo especificado (usualmente dos) del promedio del peso inicial. Se debe tener precaución en evitar que el escalamiento produzca valores negativos de peso.

El escalamiento lineal trabaja bien excepto cuando ocurren valores negativos del peso escalado.

2. *Truncamiento sigma(σ)*

Este procedimiento, se llama así por el uso de la desviación estándar de la población, una constante es sustraída del peso inicial como sigue:

$$f' = f - (\bar{f} - c \cdot \sigma)$$

En esta ecuación la constante c es escogida como un razonable múltiplo de la desviación estándar de la población (entre 1 y 3) y los resultados negativos ($f' < 0$) son igualados a cero.

3. *Escalamiento con una ley de potencia*

Gillies (1985) sugirió una ley de potencia del escalamiento donde el peso escalado es tomado como alguna potencia especificada del peso inicial f :

$$f' = f^k$$

En general el valor de k es dependiente del problema y puede requerir cambios durante la corrida para estrechar o ampliar el rango como se necesite.

3.5 El cruce

El cruce es uno de los mecanismos primarios de búsqueda en los Algoritmos Genéticos. Mediante este mecanismo se toman los dos cromosomas seleccionados para ser padres y se producen dos nuevos cromosomas basados en la información de los padres. Se escoge una probabilidad para que se produzca el cruce (pCross), se elige un número aleatorio de 0 a 1, si el número es menor que pCross se produce el cruce.

Se utilizaron diversos métodos para hacer el cruce, a continuación se explican cada uno de ellos:

3.5.1 Cruce de n-puntos

La idea detrás del cruce de n-puntos es alinear dos cromosomas juntos, cortar ambos cromosomas en $n+1$ fragmentos, y crear dos nuevos hijos mezclando los fragmentos.

Como ejemplo se presenta, el cruce de 3-puntos con dos cromosomas de longitud 10:

$$\begin{array}{l} p^0 = p_0^0 \ p_1^0 \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ p_6^0 \ p_7^0 \ p_8^0 \ p_9^0 \\ p^1 = p_0^1 \ p_1^1 \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ p_6^1 \ p_7^1 \ p_8^1 \ p_9^1 \end{array}$$

$$\begin{array}{l} p^0 = p_0^0 \ p_1^0 \ | \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ | \ p_6^0 \ | \ p_7^0 \ p_8^0 \ p_9^0 \\ p^1 = p_0^1 \ p_1^1 \ | \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ | \ p_6^1 \ | \ p_7^1 \ p_8^1 \ p_9^1 \end{array}$$

$$\begin{array}{l} p^0 = p_0^0 \ p_1^0 \ | \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ | \ p_6^0 \ | \ p_7^0 \ p_8^0 \ p_9^0 \\ p^1 = p_0^1 \ p_1^1 \ | \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ | \ p_6^1 \ | \ p_7^1 \ p_8^1 \ p_9^1 \\ \hline c^0 = p_0^0 \ p_1^0 \ | \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ | \ p_6^0 \ | \ p_7^1 \ p_8^1 \ p_9^1 \\ c^1 = p_0^1 \ p_1^1 \ | \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ | \ p_6^1 \ | \ p_7^0 \ p_8^0 \ p_9^0 \end{array}$$

Los puntos de cruce son escogidos aleatoriamente.

3.5.2 Cruce mezclado de n-puntos

Este método es similar al anterior, excepto que los genes son mezclados antes del cruce, y desmezclados después del cruce.

Veamos como funciona este método mediante un ejemplo:

$$\begin{array}{l}
p^0 = p_0^0 \ p_1^0 \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ p_6^0 \ p_7^0 \ p_8^0 \ p_9^0 \\
p^1 = p_0^1 \ p_1^1 \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ p_6^1 \ p_7^1 \ p_8^1 \ p_9^1 \\
\\
p^0 = p_2^0 \ p_5^0 \ p_6^0 \ p_9^0 \ p_0^0 \ p_7^0 \ p_8^0 \ p_4^0 \ p_3^0 \ p_1^0 \\
p^1 = p_2^1 \ p_5^1 \ p_6^1 \ p_9^1 \ p_0^1 \ p_7^1 \ p_8^1 \ p_4^1 \ p_3^1 \ p_1^1 \\
\\
p^0 = p_2^0 \ p_5^0 \ | \ p_6^0 \ p_9^0 \ p_0^0 \ p_7^0 \ | \ p_8^0 \ | \ p_4^0 \ p_3^0 \ p_1^0 \\
p^1 = p_2^1 \ p_5^1 \ | \ p_6^1 \ p_9^1 \ p_0^1 \ p_7^1 \ | \ p_8^1 \ | \ p_4^1 \ p_3^1 \ p_1^1 \\
\\
p^0 = p_2^0 \ p_5^0 \ | \ p_6^0 \ p_9^0 \ p_0^0 \ p_7^0 \ | \ p_8^0 \ | \ p_4^0 \ p_3^0 \ p_1^0 \\
p^1 = p_2^1 \ p_5^1 \ | \ p_6^1 \ p_9^1 \ p_0^1 \ p_7^1 \ | \ p_8^1 \ | \ p_4^1 \ p_3^1 \ p_1^1 \\
c^0 = p_2^0 \ p_5^0 \ | \ p_6^1 \ p_9^1 \ p_0^1 \ p_7^1 \ | \ p_8^0 \ | \ p_4^1 \ p_3^1 \ p_1^1 \\
c^1 = p_2^1 \ p_5^1 \ | \ p_6^0 \ p_9^0 \ p_0^0 \ p_7^0 \ | \ p_8^1 \ | \ p_4^0 \ p_3^0 \ p_1^0 \\
\\
p^0 = p_0^0 \ p_1^0 \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ p_6^0 \ p_7^0 \ p_8^0 \ p_9^0 \\
p^1 = p_0^1 \ p_1^1 \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ p_6^1 \ p_7^1 \ p_8^1 \ p_9^1 \\
c^0 = p_0^1 \ p_1^1 \ p_2^0 \ p_3^1 \ p_4^1 \ p_5^0 \ p_6^1 \ p_7^1 \ p_8^0 \ p_9^1 \\
c^1 = p_0^0 \ p_1^0 \ p_2^1 \ p_3^0 \ p_4^0 \ p_5^1 \ p_6^0 \ p_7^0 \ p_8^1 \ p_9^0
\end{array}$$

En este método para obtener el orden de la mezcla se ha realizado de dos maneras; la mezcla lenta donde se elige la permutación de los números de 0 a $l-1$ de manera aleatoria, y la manera rápida donde se escoge un número de genes a ser copiado de acuerdo de acuerdo a una fórmula [14] para calcular la probabilidad de copiar c genes, para un cruce mezclado de n -puntos, de un cromosoma de longitud l está dado por:

$$\frac{\binom{c-1}{c-k} \binom{l-c-1}{n-k} + \binom{l-c-1}{k-1} \binom{c-1}{n-k}}{2 \binom{l-1}{l-n-1}}$$

donde $k = (n+1)/2$ cuando n es impar, $k = (n+2)/2$ cuando n es par.

3.5.3 Cruce uniforme

Dado dos padres cromosomas de longitud l , cada padre copia $l/2$ genes a cada hijo, se realiza una selección de cada gene de manera aleatoria y luego se copian los genes escogidos y el resto del otro padre.

A continuación se muestra un ejemplo:

$$\begin{array}{r}
 p^0 = p_0^0 \ p_1^0 \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ p_6^0 \ p_7^0 \ p_8^0 \ p_9^0 \\
 p^1 = p_0^1 \ p_1^1 \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ p_6^1 \ p_7^1 \ p_8^1 \ p_9^1 \\
 \\
 p^0 = p_0^0 \ p_1^0 \ |p_2^0| \ p_3^0 \ |p_4^0| \ |p_5^0| \ p_6^0 \ |p_7^0| \ |p_8^0| \ p_9^0 \\
 p^1 = p_0^1 \ p_1^1 \ |p_2^1| \ p_3^1 \ |p_4^1| \ |p_5^1| \ p_6^1 \ |p_7^1| \ |p_8^1| \ p_9^1 \\
 \\
 p^0 = p_0^0 \ p_1^0 \ p_2^0 \ p_3^0 \ p_4^0 \ p_5^0 \ p_6^0 \ p_7^0 \ p_8^0 \ p_9^0 \\
 p^1 = p_0^1 \ p_1^1 \ p_2^1 \ p_3^1 \ p_4^1 \ p_5^1 \ p_6^1 \ p_7^1 \ p_8^1 \ p_9^1 \\
 \hline
 c^0 = p_0^1 \ p_1^1 \ p_2^0 \ p_3^1 \ p_4^0 \ p_5^0 \ p_6^1 \ p_7^0 \ p_8^0 \ p_9^0 \\
 c^1 = p_0^0 \ p_1^0 \ p_2^1 \ p_3^0 \ p_4^1 \ p_5^1 \ p_6^0 \ p_7^1 \ p_8^1 \ p_9^0
 \end{array}$$

3.5.4 Cruce uniforme modificado

Este método fue introducido por Pawlowsky [14]. La idea es copiar todos los genes que son iguales en ambos padres a los hijos, luego copiar la mitad restante de genes de cada padre a cada hijo como en el cruce uniforme.

3.5.5 Cruce variable

En el cruce variable, cada cromosoma es prefijado con extra bits, con los extra bits se determinan el método de cruce a escoger. En nuestro caso el extra bit, indica si el cruce de n-puntos será mezclado o no. La idea es que a lo largo de la búsqueda del vector solución óptimo, el GA pueda también encontrar el método de cruce óptimo.

3.5.6 Cruce asexual

En este método no se realiza un intercambio de la información genética, cada padre genera su propio hijo, mediante el intercambio de sus bits; este método de cruce es similar a la mutación con intercambio, la diferencia está en que la probabilidad de cruce es normalmente unas 80 veces mayor que la probabilidad de mutación.

3.6 La mutación

La mutación juega un decidido rol secundario en la operación de algoritmos genéticos. La mutación es necesaria porque a pesar de que la selección y el cruce efectivamente buscan y recombinan, ocasionalmente ellos pueden llegar a ser sobrecegos y perder algún material genético (1 o 0 en una locación particular). En sistemas genéticos artificiales, el operador de mutación protege en contra de tales pérdidas irreversibles. En el GA simple, la mutación es la alteración aleatoria ocasional del valor de una posición de la cadena.

La mutación ha sido implementada de dos formas:

3.6.1 Método aleatorio de inversión

Existe una probabilidad de que un bit pueda ser mutado, esta probabilidad (p_{Mut}) es seteada por el usuario, usualmente es un valor muy pequeño. Un mecanismo de tiro de la moneda es empleado; si un número aleatorio entre 0 y 1 es menor que p_{Mut} el bit es invertido, así los ceros llegan a ser unos y unos llegan a ser ceros.

3.6.2 Método de intercambio

En este caso si un número aleatorio entre 0 y 1 es menor que p_{Mut} se intercambia dos posiciones de bits elegidas aleatoriamente.

3.7 El reemplazo

El reemplazo es el último paso del ciclo reproductivo. Dos padres son seleccionados de una población, ellos tienen dos hijos, pero no todos los cuatro pueden regresar a la población, así que los padres pueden ser reemplazados. La técnica usada para decidir que individuos se quedan en la población y cuales son reemplazados es equivalente a la selección de los que más influyen en la convergencia.

Los métodos que se han usado son los siguientes:

3.7.1 Agregar

Este es el método más común. En este caso se toman los dos hijos y se agregan a una nueva población y así sucesivamente hasta que se crea una población con todos los nuevos hijos.

3.7.2 El primero más débil

En este método el hijo reemplaza al primer individuo de la población más débil que él. Luego en esta nueva población, nuevamente se busca el primero más débil que el segundo hijo y así sucesivamente.

3.7.3 El más débil

En este método el hijo reemplazará al individuo más débil de toda la población, luego se procede de igual manera con el segundo hijo con la nueva población.

3.8 Operadores adicionales

Existen diversos operadores adicionales a los ya mencionados, pero en el programa implementado sólo se han usado los más importantes que son el Niching y el ruido. Adicionalmente tiene la opción del elitismo, cuando está activada esta opción se copia el mejor de una generación a la otra, siempre que en la nueva

generación no haya uno mejor. El gap es el número que indica que cantidad de la población se pasa directamente a la nueva, tiene valores bajos de 0,1 a 0,2.

3.8.1 Niching

El mecanismo del GA simple es replicar los cromosomas con más altos pesos y descartar los cromosomas con más bajos pesos, guiando la convergencia de la población. Como sea, teniendo soluciones alternativas de pesos idénticos, la población convergirá a una de ellas. El GA simple pierde soluciones y subsoluciones debido a tres efectos: *presión de la selección*, *ruido de la selección* y *el operador de disrupción*. La presión de la selección es el resultado del valor esperado del proceso de selección: soluciones con pesos más bajos son esperadas a desaparecer de una población finita. El ruido de la selección resulta de la varianza del proceso de selección: en una población finita, elecciones aleatorias entre idénticamente pesadas, subsoluciones competentes agregan ruido al conteo esperado por cada individuo, eventualmente forzando buenas soluciones de la población. Operadores de disrupción resultan de la aplicación del cruce y la mutación, los cuales son capaces de destruir directamente buenas soluciones.

Técnicas para diversificar a la población típicamente reducen o la presión de la selección, el ruido de la selección o la disrupción del operador (o alguna combinación de estos). Técnicas que promueven la diversidad de la población no se pueden calificar de métodos de niching.

Los métodos de niching son técnicas que promueven la formación y mantenimiento de subpoblaciones estables en el GA. Los métodos niching pueden ser aplicados a la formación y el mantenimiento de subsoluciones provisionales en el camino a una singular, solución final. Ellos son tradicionalmente vistos, como sea, en

el contexto de formar y mantener múltiples, soluciones finales. Un método de niching debe ser capaz de mantener estas soluciones por un exponencial a infinito periodo de tiempo, con respecto al tamaño de la población.

Se ha usado dos métodos de niching:

- *Apiñamiento determinístico*

En el GA *generacional*, la población entera es reemplazada cada generación por los hijos formados a través del cruce y la mutación. En el GA de *estado estacionario*, por otro lado, procesa sólo unos pocos individuos a la vez, e inserta los hijos resultantes dentro de la población. Algunas técnicas se esfuerzan por preservar diversidad en este paso de reemplazo. En vez de practicar el reemplazo mayorista generacional, o la selección reemplazante (reemplazar el peor), ellos intentan reemplazar los miembros de la población con el fin de mantener la diversidad. A los métodos que insertan nuevos elementos dentro de una población reemplazando elementos similares se les llama *métodos de apiñamiento*.

De Jong(1975) presenta un algoritmo que llama el “modelo del factor de apiñamiento”. El apiñamiento de De Jong es un GA de estado estacionario, ya que sólo una fracción de la población se reproduce y muere en cada generación. Cada miembro nuevo reemplaza un miembro existente, preferiblemente el más similar. Para ejecutar una aproximación del reemplazo del más similar, una pequeña muestra es tomada de la población existente, y el nuevo miembro reemplaza el elemento más cercano de la muestra.

Específicamente, el apiñamiento de De Jong trabaja como sigue. Una proporción de la población, especificada por el $gap(GG)$. Después del cruce y de la mutación, $(1-GG) \times n$ individuos de la población son escogidos para morir. Cada hijo encuentra

el elemento que este reemplaza al tomar una muestra aleatoria de CF individuos de la población, donde CF es llamado factor de apiñamiento. El hijo reemplaza el individuo más similar de la muestra. La similitud es definida usando la distancia de Hamming, conocida como *comparación genotípica*.

La meta original de De Jong en el diseño del apiñamiento fue mantener la diversidad de los bits con el fin de prevenir la convergencia prematura.

Cinco años después al trabajo de De Jong, Cavicchio (1970) introdujo muchos de los llamados esquemas de *preselección*, y demandaba que uno de ellos era satisfactorio en preservar la diversidad de la población.

En 1992, a través del análisis del apiñamiento de De Jong y la preselección de Cavicchio, Mahfoud pudo determinar porque ninguno de los métodos fue satisfactorio, y desarrollar modificaciones de diseño que produzcan un algoritmo de apiñamiento satisfactorio. El algoritmo resultante, *apiñamiento determinístico* (crowding), poseía extensivas capacidades de niching cuando se aplica a ambos, problemas de optimización multimodal y de clasificación. El apiñamiento determinístico trabaja de la siguiente manera. Primero, se agrupa a la población dentro de $n/2$ pares. Luego se cruza a todos los pares y opcionalmente se muta a los hijos. Cada hijo compete en un torneo en contra de uno de los padres que lo produjo. Dado un par de padres y sus dos hijos, es posible dos conjuntos de torneos padre-hijo. El conjunto de torneos que fuerzan las competencias del más cercano, donde la cercanía es la distancia promedio entre las parejas de padre-hijo.

A este método se le ha adicionado la opción de tener prevención del incesto, ésta fue propuesta por Eshelman (1991) la cual desalentaba los clones. La prevención del incesto intenta ser un “casamentero” de padres con la intención de que sus hijos

tomen los mejores genes de sus padres. Esto se logra tomando un grupo de miembros y tomar los más lejanos como padres, de esa manera se mantiene la diversidad y se continúa con la evolución.

- *Compartición*

Holland (1975, 1992) discute el concepto de limitar el número de individuos a ocupar un nicho, a la capacidad de este nicho. Él manifiesta que si un nicho se ha asociado con un peso ajustado en cada paso, si cada individuo que ocupa este nicho es forzado a compartir igualmente este peso, luego una situación estable se alcanza cuando cada nicho contiene un número de individuos proporcional a ese peso. Si algunos nichos llegan a estar sobresaturados, los individuos que ocupan estos nichos buscan nichos menos llenos. Métodos de niching que usan este concepto de Holland son llamados métodos de compartición. Métodos de compartición pueden ser definidos como algoritmos que requieren miembros de la población similares para compartir el peso. Para inducir compartición, estos algoritmos típicamente alteran el peso de cada miembro de la población basados en su proximidad a otros miembros de la población.

El método que se ha usado trabaja de la siguiente manera. El *peso compartido* del nuevo individuo, f' , es igual a su viejo peso f dividido por su *cómputo de nicho*. El cómputo de nicho es la suma de los valores de las *funciones de compartición* (sh) entre el mismo y cada individuo en la población (incluyendo el mismo). El peso compartido de un miembro de la población i es

$$f'(i) = \frac{f(i)}{\sum_{j=1}^n sh(d(i, j))}$$

La función de compartición sh es una función de la distancia d entre dos miembros de la población; esta regresa un '1' si los elementos son idénticos, un '0' si ellos exceden de algún nivel de desigualdad, y un valor intermedio para niveles intermedios de desigualdad. La función que se ha usado (la más común) tiene la siguiente forma:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}} \right)^\alpha, & \text{si } d < \sigma_{share}; \\ 0 & \text{en otro caso} \end{cases}$$

3.8.2 Ruido

El tamaño de la cadena binaria depende del número de parámetros a optimizar y la precisión deseada. Esta codificación binaria posibilita: (1) el análisis teórico del GA, tal como el *teorema del esquema* y la *hipótesis de los bloques construidos* y (2) los operadores genéticos elegantes, tal como el *cruce* y la *mutación*. Como sea, el esquema de representación binaria, el cual discretiza el espacio de búsqueda, tiene algunos inconvenientes cuando se aplica a problemas de optimización de una alta precisión numérica.

Con el fin de mejorar la performance del GA binario, se introduce intencionalmente ruido dentro de la variable de búsqueda x_i . El ruido posibilita el muestreo de la región de búsqueda que un cromosoma binario no puede acceder directamente. Se considera también la asignación del ruido a la función objetivo con el fin de mejorar la confiabilidad de la búsqueda global. En un ambiente de ruido, examinar más individuos usualmente incrementa la probabilidad de obtener una mejor solución con una resolución más alta.

Para proveer un examen inicial de las propiedades de convergencia de un algoritmo genético, considere un espacio de estado de posibles soluciones codificados como una cadena de k bits $\{0,1\}$. Hagamos que exista una población, m , de tales cadenas binarias y hagamos que cada posible cadena tenga un peso, f_i , $i= 1, 2, \dots, 2^k$. Hagamos un óptimo global que tiene el peso máximo. Para el GA binario, los estados de un individuo pueden ser definidos por cada posible configuración de una cadena de bits. Así, hay 2^k estados para una cadena binaria de tamaño k . En este esquema, el número de puntos de búsqueda en el espacio del problema es limitado por el número de bits. Incrementando el tamaño del bit de un cromosoma se incrementa la resolución de la búsqueda.

Con el fin de aliviar el problema de convergencia en el GA binario, se introduce el ruido dentro de la variable de búsqueda x_i y evaluamos la función objetivo. Figura 3.5 muestra el concepto de la asignación de ruido en los parámetros de búsqueda x_i . El parámetro x_i es obtenido de la cadena de bits del individuo. El ruido es luego agregado al parámetro antes de la evaluación de la función objetivo (se asume que la función objetivo tiene dos parámetros x_1, x_2).

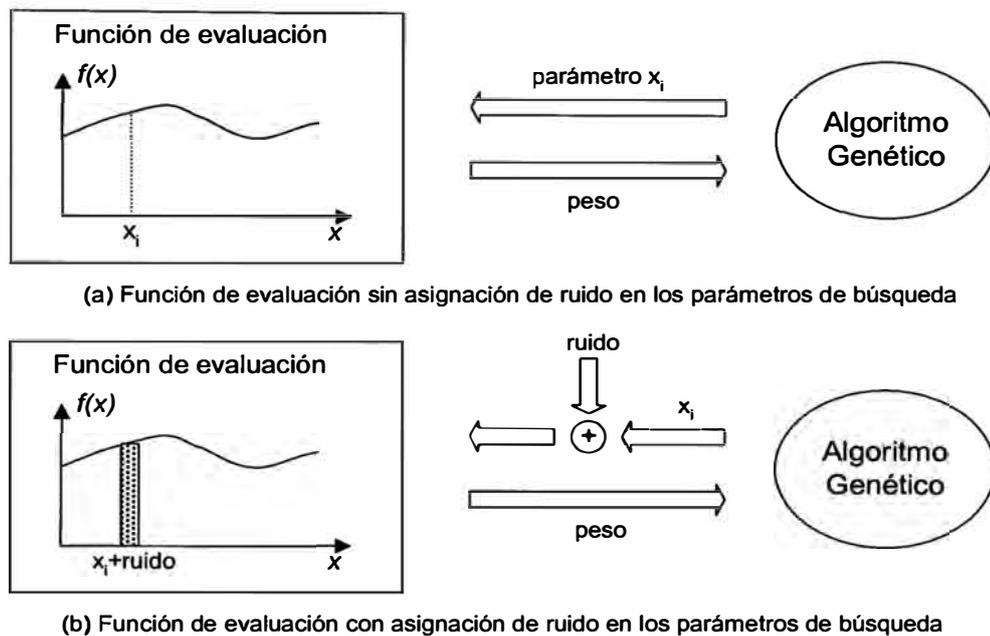


Fig. 3.5 Esquema de asignación de ruido para el parámetro de búsqueda en la función de evaluación.

La función de evaluación sin asignación de ruido:

$$FITNESS = f(x_1, x_2)$$

Función objetivo con asignación de ruido:

$$FITNESS = f(x_1 + ruido, x_2 + ruido)$$

En la figura 3.6 se muestra como el ruido asiste al cromosoma binario para una mayor exploración de los rangos de búsqueda. Asumir un individuo de tres bits de longitud representa uno de ocho puntos en el rango de búsqueda [MIN, MAX]. Luego, un GA puede sólo hacer un muestreo ocho de estados del espacio de búsqueda total. Sin embargo, en vez de emplear más bits para incrementar la precisión, nosotros introducimos ruido aleatorio dentro del parámetro x_i . La función consecuentemente analiza x_i más algún ruido. En el programa se implementa dos tipos de ruido aleatorio: (1) Ruido aleatorio uniformemente distribuido, $U[x_i - D_x/2, x_i + D_x/2]$ y (2) Ruido aleatorio de la distribución Gausseana, $m = x_i$, $\sigma = D_x/2$. Aquí $D_x = (MAX - MIN) / (2^n - 1)$ donde $n =$ numero de bits.

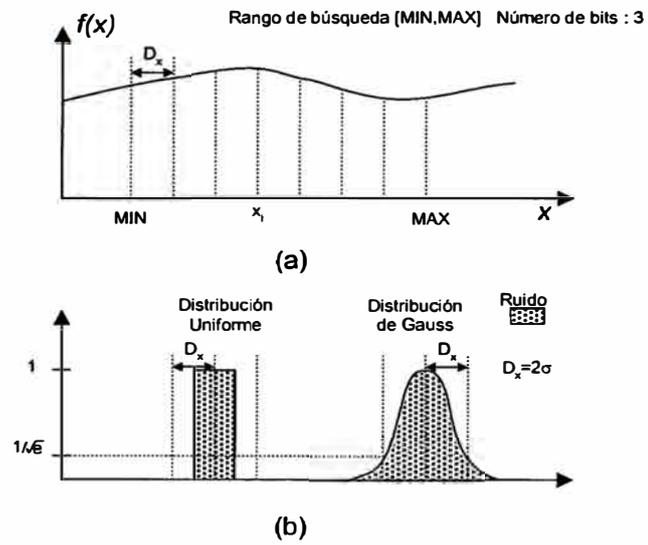


Fig. 3.6 Asignación del ruido a la variable de búsqueda.

CAPÍTULO IV

FLUJO DE CARGA APROVECHANDO EL CARÁCTER DE LAS MATRICES

4.1 Introducción

El cálculo de flujo de carga provee los flujos de potencia y tensiones para un sistema de potencia especificado sujeto a la capacidad reguladora de los generadores, capacitores, cambio de taps bajo carga y FACTS. Esta información es esencial para la continua evaluación del desempeño de la corriente de un sistema de potencia y para analizar la eficiencia de las alternativas para expansión del sistema para asumir el incremento de la carga. Estos análisis requieren el cálculo de numerosos flujos de carga para ambas condiciones de operación, normal y contingencia.

El problema de flujo de carga consiste en el cálculo del flujo de potencia y voltajes de una red para condiciones específicas de las barras. Una representación unifilar es adecuada ya que el sistema de potencia es usualmente balanceado. A cada barra se le asocian cuatro cantidades: la potencia activa y reactiva, la magnitud del voltaje, y el ángulo de fase. Tres tipos de barras son representadas en el cálculo del flujo de carga y en una barra, dos de las cuatro cantidades son especificadas. Es necesario seleccionar una barra, llamada barra slack(SL), para proveer la potencia reactiva y activa adicional para suministrar las pérdidas de la red, ya que estas son desconocidas hasta que la solución final es obtenida. En esta barra la magnitud del voltaje y el ángulo de fase son especificados. Las barras restantes del sistema son

designadas ya sea como barras de voltaje controlado o como barras de carga. La potencia activa y la magnitud del voltaje son especificados como barras de voltaje controlado(PV). La potencia activa y reactiva son especificadas como barras de carga(PQ).

Las conexiones a la red son descritas usando números que se asignan a cada barra. Estos números especifican los terminales de las líneas de transmisión y transformadores. Números también se usan para identificar los tipos de barras.

Las dos consideraciones primarias en el desarrollo de un programa de computo de ingeniería son: (1) la formulación de una descripción matemática del problema; y (2) la aplicación de un método numérico para la solución. El análisis del problema debe también considerar la relación entre estos dos factores.

La formulación matemática del problema del flujo de carga resulta en un sistema de ecuaciones algebraicas no lineales. Estas ecuaciones pueden ser establecidas usando ya sea la estructura de las barras o de los lazos. Los coeficientes de las ecuaciones dependen de la selección de las variables independientes, por ejemplo, voltajes o corrientes. Así, la matriz de impedancias o de admitancias pueden ser usadas.

Las primeras aproximaciones que se hicieron para encontrar la solución del flujo de carga emplearon la estructura de lazos en forma de admitancias (Y_{LOOP}). La matriz de admitancias de lazo fue obtenida por una inversión de la matriz. Estos métodos no han sido extendidos porque la preparación de la data es tediosa. Además, la inversión de la matriz consume un tiempo. Luego se empezó a trabajar con la estructura de barras en la forma de admitancias(Y_{BUS}). Este método se ha difundido más por la simplicidad de la preparación de la data. Esta aproximación sigue siendo

la más económica desde el punto de vista de requerimientos de tiempo y de memoria del computador.

4.2 Formulación matemática del problema

4.2.1 Ecuaciones de comportamiento de la red

La ecuación que describe el comportamiento de la red en un sistema de potencia usando la estructura de barras en forma de impedancia es

$$[E_{BUS}] = [Z_{BUS}] * [I_{BUS}] \quad (4.1)$$

en forma de admitancia tenemos

$$[I_{BUS}] = [Y_{BUS}] * [E_{BUS}] \quad (4.2)$$

Las matrices de admitancia y de impedancia de barra pueden estar formadas por la red incluyendo la barra de tierra. Los elementos de la matriz, luego incluirán los efectos de los elementos shunts a tierra tales como capacitores y reactores, elementos shunts del equivalente de los transformadores y las líneas. Cuando la barra de tierra es incluida y seleccionada como el nodo de referencia, los voltajes de barra y ecuaciones que describen el comportamiento de la red (4.1) y (4.2) son medidas con respecto a tierra.

Si la barra a tierra no es incluida en la red, los elementos de las matrices de impedancia y de admitancia de barra no incluirán los efectos de los elementos shunts y una de las barras debe ser seleccionada como nodo de referencia. En este caso, los efectos de los elementos shunts son tratados como fuentes de corrientes en las barras de la red y los voltajes de barra en las ecuaciones (4.1) y (4.2) son medidas con respecto a la barra seleccionada como referencia.

Usando la estructura de lazo de referencia, la ecuación de comportamiento de la red en forma de impedancia es

$$[E_{LOOP}] = [Z_{LOOP}] * [I_{LOOP}] \quad (4.3)$$

o en la forma de admitancia es

$$[I_{LOOP}] = [Y_{LOOP}] * [E_{LOOP}] \quad (4.4)$$

Cuando las matrices de impedancia y de admitancia de lazo son formadas por la red sin incluir los elementos shunts, las dimensiones de las matrices son $l \times l$, donde l es el número de enlaces o lazos básicos calculados de

$$l = e - n + 1$$

e es el número de elementos, excluyendo las conexiones shunt, y n es el número de nodos. En este caso, los efectos de los elementos shunt son tratados como fuentes de corriente en las barras de la red.

Si los elementos shunt e_s son incluidos en forma de las matrices lazo, el número de elementos de la red es incrementado por e_s . El número total de elementos es, entonces, $e + e_s$ y el número total de nodos es incrementado a $n + 1$. Consecuentemente, el número de lazos y la dimensión de las matrices lazo son incrementados por $e_s - 1$.

Las diferentes formas de las ecuaciones de red son resumidas en la tabla 4.1:

<i>Estructura de Referencia</i>	<i>Forma de parámetros</i>	
	<i>Impedancia</i>	<i>Admitancia</i>
<i>Barras</i>	$[E_{BUS}] = [Z_{BUS}] * [I_{BUS}]$	$[I_{BUS}] = [Y_{BUS}] * [E_{BUS}]$
<i>Lazo</i>	$[E_{LOOP}] = [Z_{LOOP}] * [I_{LOOP}]$	$[I_{LOOP}] = [Y_{LOOP}] * [E_{LOOP}]$

Tabla 4.1

4.2.2 Ecuaciones de Potencia

La potencia activa y reactiva en la barra p es

$$P_p - jQ_p = E_p^* I_p \quad (4.5)$$

y la corriente es

$$I_p = \frac{P_p - jQ_p}{E_p^*} \quad (4.6)$$

donde I_p es positiva cuando fluye dentro del sistema.

En la formulación de la ecuación de red, si los elementos shunt a tierra son incluidos, entonces la corriente total en la barra p es

$$I_p = \frac{P_p - jQ_p}{E_p^*} - y_p E_p \quad (4.7)$$

donde y_p es la admitancia shunt total en la barra e $y_p E_p$ es la corriente shunt fluyendo de la barra p a tierra.

4.2.3 Ecuaciones de flujo de línea

Después que la solución iterativa de las tensiones de barra es terminada, los flujos de línea pueden ser calculadas. La corriente en la barra p en la línea que conecta la barra p a la q es

$$i_{pq} = (E_p - E_q) y_{pq} + E_p \frac{y'_{pq}}{2} \quad (4.8)$$

donde y_{pq} = admitancia de línea

y'_{pq} = admitancia total que se carga a la línea

$E_p \frac{y'_{pq}}{2}$ = contribución de corriente en la barra p debida a la carga de la línea

El flujo de potencia, activa y reactiva, es

$$P_{pq} - jQ_{pq} = E_p^* i_{pq}$$

o

$$P_{pq} - jQ_{pq} = E_p^* (E_p - E_q) y_{pq} + E_p^* E_p \frac{y'_{pq}}{2} \quad (4.9)$$

donde en la barra p el flujo de potencia activa de la barra p a la q es P_{pq} y la reactiva es Q_{pq} . Similarmente, en la barra q el flujo de potencia de q a p es

$$P_{qp} - jQ_{qp} = E_q^* (E_q - E_p) y_{pq} + E_q^* E_q \frac{y'_{pq}}{2} \quad (4.10)$$

La potencia perdida en la línea p - q es la suma algebraica del flujo de potencia determinado de las ecuaciones (4.9) y (4.10).

4.3 Evaluación de técnicas de solución

Una evaluación de los métodos para obtener una solución de un flujo de carga debe incluir lo siguiente:

- El tiempo computacional requerido en el proceso de entrada de datos para obtener los parámetros para el cálculo iterativo.
- Requerimientos de memoria
- Tiempo de iteración
- El tiempo computacional requerido para modificar la red y para efectuar cambios en la operación del sistema.

El primer paso en los métodos de flujo de carga es la codificación de la red y la formación de la matriz de red apropiada. En la estructura de barras la asignación de los números de barra provee una información adecuada para describir las conexiones en la red. En la estructura de lazo es necesario identificar los lazos básicos de la red.

La matriz de admitancia de barra puede formarse por un procedimiento directo y simple debido a que no existe acoplamiento mutuo. Un elemento diagonal Y_{pp} de

esta matriz es igual a la suma de las admitancias de los elementos de la red conectados a la barra p . Un elemento fuera de la diagonal Y_{pq} es igual al negativo de la admitancia del elemento que conecta la barra p a la barra q . Además, ya que la matriz de admitancia es esparcida, esto es, un gran número de elementos son cero, relativamente pocos elementos tienen que ser calculados. La matriz de admitancia de barra es almacenada como una matriz esparcida, con lo que se ahorra memoria computacional.

La formación de la matriz de impedancia de barra requiere una inversión de matriz, transformaciones no singulares, o el uso de un algoritmo. Al contrario de la matriz de admitancia de barra, la matriz de impedancia de barra es una matriz completa, una matriz que tiene elementos no nulos excepto en la fila y la columna asociada con la barra de referencia. Consecuentemente, para un sistema de 101 barras, del cual una barra es de referencia, 20 000 palabras serían requeridas para almacenar la matriz compleja entera. Ya que la matriz de impedancia de barra es simétrica, sólo los elementos de la diagonal y la mitad de los elementos fuera de la diagonal necesitan ser almacenados. Esto reduce los requerimientos de almacenamiento a 10100 palabras. En contraste, un sistema de 101 barras con un promedio de cuatro líneas por barra requiere sólo 1000 palabras de memoria de computadora para almacenar todos los elementos no nulos complejos, de la matriz de admitancia de barra. Tomando ventaja de la simetría se reduciría el almacenamiento a 600 palabras.

La formación de la matriz de admitancia de lazo envuelve la inversión de la matriz, usando o la matriz de impedancia de lazo obtenida por una transformación singular o por la matriz de admitancia aumentada obtenida por una transformación

no singular. Como una alternativa con la matriz de impedancia de barra y un algoritmo se puede obtener la matriz de admitancia de lazo. La matriz de admitancia de lazo es una matriz completa.

Programas de computadora han sido desarrollados para evaluar la efectividad de los métodos presentados para la solución del flujo de carga. Estos programas fueron usados para obtener soluciones de flujo de carga y obtener tiempos relativos.

El tiempo computacional requerido para realizar la solución iterativa depende de lo siguiente:

1. El número de operaciones lógicas y aritméticas requeridas para completar una iteración.
2. La tasa de convergencia de la técnica de solución
3. El tamaño y características del sistema de potencia.

En comparación con el método de Gauss-Seidel, el método de Gauss usando la matriz de admitancia de barra o la matriz de impedancia de barra requiere iteraciones adicionales para obtener una solución. Ya que el tiempo por iteración para estos dos métodos es casi el mismo, el método de Gauss no fue evaluado con detalle. El método de relajación usando la matriz de admitancia de barra también requería más iteraciones y más tiempo por iteración y así, este no fue estudiado con detalle.

En el desarrollo de los programas de computadora para los métodos de Gauss-Seidel y de Newton-Raphson usando la matriz de admitancia de barra, se saca ventaja de la esparcidad de la matriz de la red para reducir el número de operaciones aritméticas por iteración.

El tiempo por iteración obtenido para los métodos principales se presentan en la Fig.4.1. El método de Gauss-Seidel usando la matriz de admitancia de barra requiere

el mínimo número de operaciones aritméticas para completar una iteración. Esto es debido a la esparcidad de la matriz de la red y a la simplicidad de la técnica de solución. Consecuentemente este método requiere el mínimo tiempo por iteración.

El método de Newton-Raphson usando la matriz de admitancia de barra saca ventaja de la esparcidad de la matriz de la red con el fin de reducir el número de operaciones aritméticas. Como sea, el cálculo de los elementos del Jacobiano por cada iteración requiere tiempo computacional adicional. El tiempo por iteración en ambos métodos se incrementa directamente con el número de barras de la red, porque el número de elementos no nulos agregados a la matriz de la red por cada nueva barra es aproximadamente el mismo.

El método de Gauss-Seidel usando la matriz de impedancia de barra requiere un procedimiento de solución relativamente simple. Como sea, el tiempo por iteración para este método es más grande y varía aproximadamente con el cuadrado del número de barras, porque la matriz de impedancia es una matriz completa.

El método de Gauss usando la matriz de admitancia del lazo requiere operaciones aritméticas y lógicas adicionales. El tiempo por iteración también varía aproximadamente con el cuadrado del número de barras, porque la matriz de admitancia de lazo es una matriz completa.

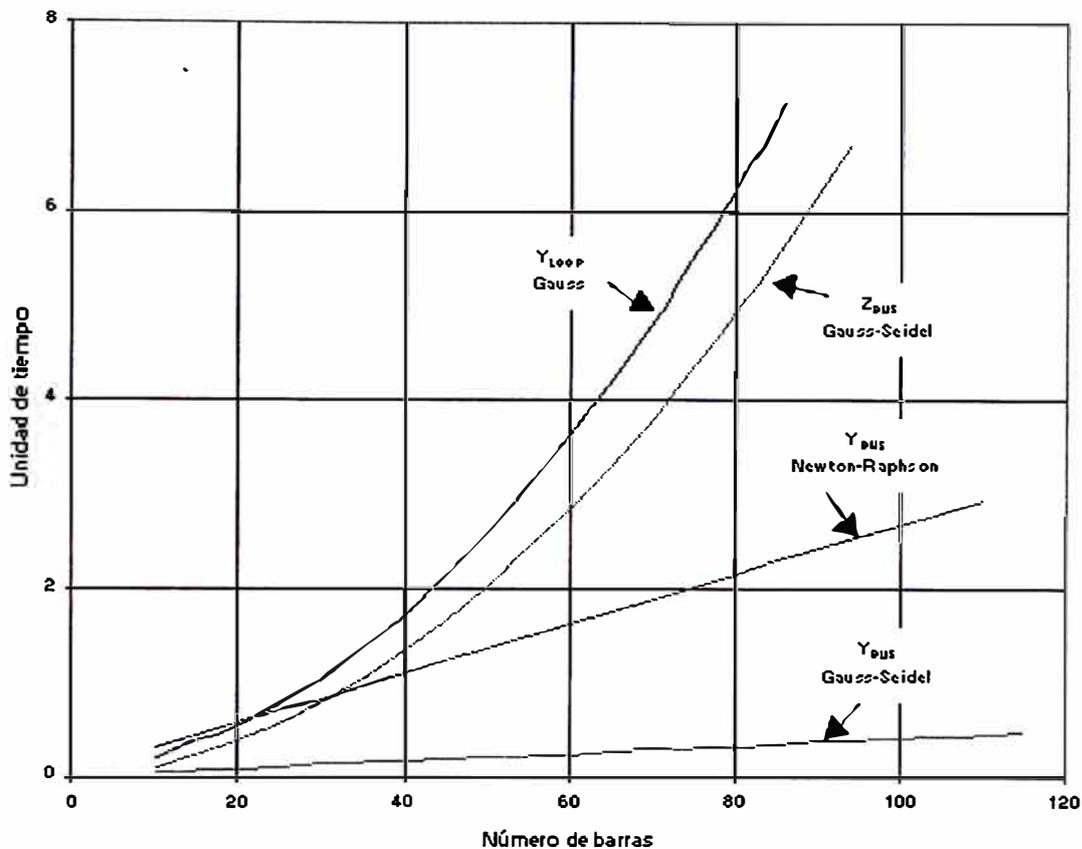


Figura 4.1 Tiempo en realizar una iteración para las diversas técnicas usadas

La razón de convergencia del método de Gauss-Seidel usando la matriz de admitancia de barra es lenta, requiere relativamente un gran número de iteraciones para obtener una solución. Además, el número de iteraciones es incrementado directamente con el número de barras de la red, mientras que el número de iteraciones para los otros métodos queda prácticamente constante, independiente del tamaño. Un significativo incremento de la tasa de convergencia del método de Gauss-Seidel usando Y_{BUS} se obtiene aplicando factores de aceleración.

La tolerancia requerida para obtener una solución varía con los diferentes métodos. El método de Gauss-Seidel usando Y_{BUS} requiere pequeñas tolerancias de tensión para obtener una exactitud comparable con la obtenida por los métodos que usan Y_{LOOP} o Z_{BUS} . En el método de Newton-Raphson usando el Y_{BUS} se tiene la

ventaja de especificar las tolerancias para la potencia activa y reactiva neta en la barra; las tolerancias son dadas directamente en cantidades que tienen un gran significado para el ingeniero que especifica la exactitud deseada. En la tabla 4.2 se muestra el número de iteraciones por solución:

Tabla 4.2

Número de barras	Y_{BUS}	Y_{BUS}	Z_{BUS}	Y_{LOOP}
	Gauss-Seidel†	Newton-Raphson‡	Gauss-Seidel§	Gauss¶
14	24	4	5	4
30	33	4	5	4
57	59	4	6	6
92	80	4	5	7
113	92	4	5	-

† Factores de aceleración de 1.7 y tolerancias de 0.0001 y 0.0001 por unidad usadas para las componentes real e imaginaria de la tensión.

‡ Tolerancias de 0.001 y 0.001 por unidad usada para las potencias activas y reactivas de la barra. No tiene factores de aceleración.

§ Tolerancias de 0.01 y 0.01 por unidad usada para las componentes real e imaginaria de la tensión. No tiene factores de aceleración.

¶ Tolerancias de 0.01 y 0.01 por unidad usada para las componentes real e imaginaria de la tensión. No tiene factores de aceleración.

El tiempo total para llegar a la solución para los principales métodos se muestra en la figura 4.2. Cuando las barras de tensión controlada son usadas, el método de Gauss-Seidel usando Y_{BUS} usualmente requiere menos iteraciones para obtener la solución, mientras que en Newton-Raphson se requiere algunas mas, pero esto no cambia el hecho de que el método de Newton-Raphson es más rápido.

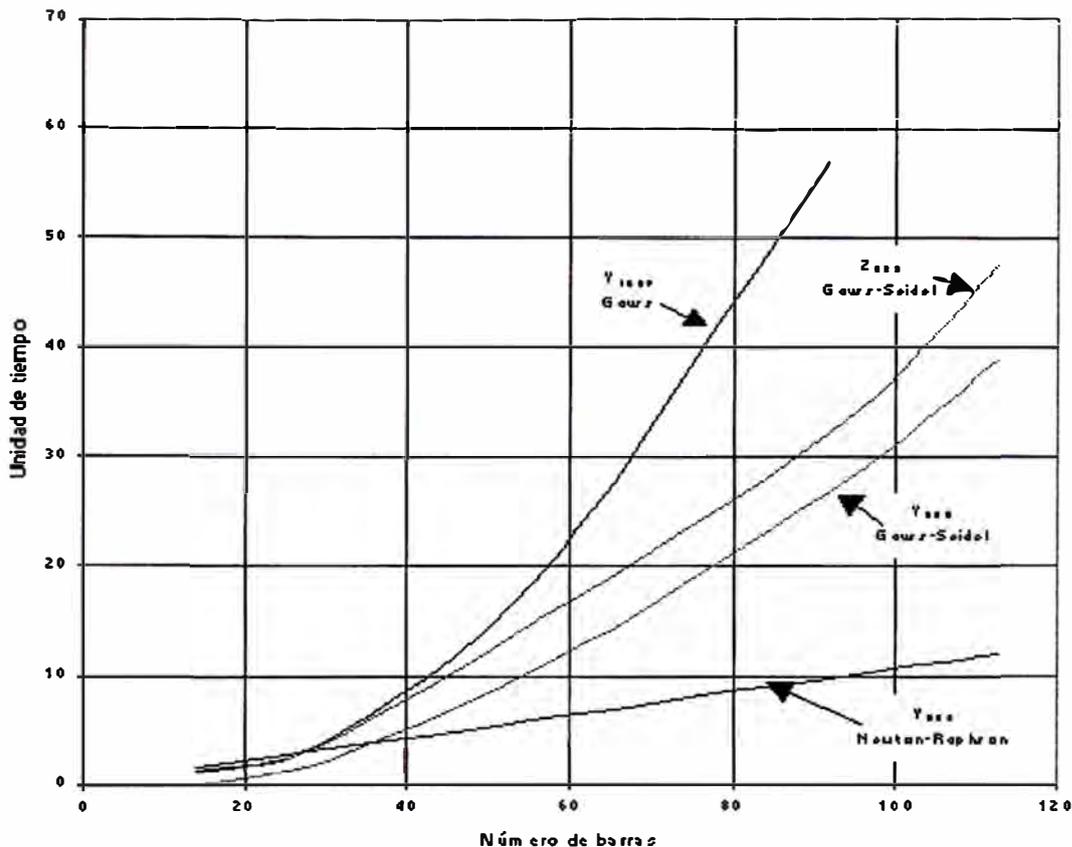


Figura 4.2 Tiempo en llegar a la solución para las diferentes técnicas usadas

4.4 Técnica de solución utilizada

De la discusión anterior, se ha escogido el método de Newton-Raphson usando el Y_{BUS} .

El problema de flujo de potencia puede ser resuelto usando un conjunto de ecuaciones no lineales para expresar la potencia activa y reactiva en término de las tensiones de barra. Sustituyendo de la ecuación de comportamiento de la red (4.2) por I_p en la ecuación (4.5), se tiene

$$P_p - jQ_p = E_p^* \sum_{q=1}^n Y_{pq} E_q \quad (4.11)$$

ya que $E_p = e_p + jf_p$ e $Y_{pq} = G_{pq} + jB_{pq}$, la ecuación (4.11) se convierte en

$$P_p - jQ_p = (e_p - jf_p) \sum_{q=1}^n (G_{pq} + jB_{pq}) (e_q + jf_q)$$

Separando la parte real e imaginaria,

$$\begin{aligned} P_p &= \sum_{q=1}^n \{ e_p (e_q G_{pq} - f_q B_{pq}) + f_p (f_q G_{pq} + e_q B_{pq}) \} \\ Q_p &= \sum_{q=1}^n \{ f_p (e_q G_{pq} - f_q B_{pq}) - e_p (f_q G_{pq} + e_q B_{pq}) \} \end{aligned} \quad (4.12)$$

Esta formulación resulta en un conjunto de ecuaciones no lineales simultáneas, dos por cada barra del sistema. La potencia activa y reactiva P_p y Q_p son conocidas y los componentes reales e imaginarios del voltaje e_p y f_p son desconocidos para todas las barras excepto para la slack, donde la tensión es especificada y queda fija. Así, hay $2(n-1)$ ecuaciones para resolver un flujo de carga.

El método de Newton-Raphson requiere que un conjunto de ecuaciones lineales se formen expresando la relación entre los cambios de potencia activa y reactiva y los componentes de las tensiones de barra como sigue:

$$\begin{array}{|c|} \hline \Delta P_1 \\ \hline \dots \\ \hline \Delta P_{n-1} \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \frac{\partial P_1}{\partial e_1} & \dots & \frac{\partial P_1}{\partial e_{n-1}} & \frac{\partial P_1}{\partial f_1} & \dots & \frac{\partial P_1}{\partial f_{n-1}} \\ \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \frac{\partial P_{n-1}}{\partial e_1} & \dots & \frac{\partial P_{n-1}}{\partial e_{n-1}} & \frac{\partial P_{n-1}}{\partial f_1} & \dots & \frac{\partial P_{n-1}}{\partial f_{n-1}} \\ \hline \end{array} \begin{array}{|c|} \hline \Delta e_1 \\ \hline \dots \\ \hline \Delta e_{n-1} \\ \hline \end{array} \quad (4.13)$$

$$\begin{array}{|c|} \hline \Delta Q_1 \\ \hline \dots \\ \hline \Delta Q_{n-1} \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \frac{\partial Q_1}{\partial e_1} & \dots & \frac{\partial Q_1}{\partial e_{n-1}} & \frac{\partial Q_1}{\partial f_1} & \dots & \frac{\partial Q_1}{\partial f_{n-1}} \\ \hline \dots & \dots & \dots & \dots & \dots & \dots \\ \hline \frac{\partial Q_{n-1}}{\partial e_1} & \dots & \frac{\partial Q_{n-1}}{\partial e_{n-1}} & \frac{\partial Q_{n-1}}{\partial f_1} & \dots & \frac{\partial Q_{n-1}}{\partial f_{n-1}} \\ \hline \end{array} \begin{array}{|c|} \hline \Delta f_1 \\ \hline \dots \\ \hline \Delta f_{n-1} \\ \hline \end{array}$$

donde los coeficientes de la matriz es el Jacobiano y la n th barra es la slack. En forma matricial, la ecuación queda:

$$\begin{array}{|c|} \hline \Delta P \\ \hline \Delta Q \\ \hline \end{array} = \begin{array}{|c|c|} \hline J_1 & J_2 \\ \hline J_3 & J_4 \\ \hline \end{array} \begin{array}{|c|} \hline \Delta e \\ \hline \Delta f \\ \hline \end{array} \quad (4.14)$$

Las ecuaciones para determinar los elementos del Jacobiano pueden ser derivada de las ecuaciones de potencia de la barra. La ecuación de la potencia activa derivada de (4.12) es.

$$P_p = e_p (e_p G_{pp} - f_p B_{pp}) + f_p (f_p G_{pp} + e_p B_{pp}) + \sum_{\substack{q=1 \\ q \neq p}}^n \{ e_p (e_q G_{pq} - f_q B_{pq}) + f_p (f_q G_{pq} + e_q B_{pq}) \} \quad (4.13)$$

Diferenciando, los elementos fuera de la diagonal de J_1 son

$$\frac{\partial P_p}{\partial e_q} = e_p G_{pq} + f_p B_{pq} \quad q \neq p$$

y los elementos de la diagonal de J_1 son

$$\frac{\partial P_p}{\partial e_p} = 2e_p G_{pp} + f_p B_{pp} - f_p B_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (e_q G_{pq} - f_q B_{pq}) \quad (4.14)$$

ya que la ecuación de la corriente en la barra p es

$$I_p = c_p + j d_p = (G_{pp} + j B_{pp}) (e_p + j f_p) + \sum_{\substack{q=1 \\ q \neq p}}^n (G_{pq} + j B_{pq}) (e_q + j f_q)$$

la cual puede ser separada en su parte real e imaginaria

$$c_p = e_p G_{pp} - f_p B_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (e_q G_{pq} - f_q B_{pq}) \quad (4.15)$$

$$d_p = f_p G_{pp} + e_p B_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (f_q G_{pq} + e_q B_{pq}) \quad (4.16)$$

Así, la expresión para los elementos de la diagonal de J_1 pueden ser modificada sustituyendo la componente real de la corriente c_p en la ecuación (4.14) para obtener

$$\frac{\partial P_p}{\partial e_p} = e_p G_{pp} + f_p B_{pp} + c_p$$

De la ecuación (4.12), los elementos fuera de la diagonal de J_2 son

$$\frac{\partial P_p}{\partial f_q} = -e_p B_{pq} + f_p G_{pq} \quad q \neq p$$

y los elementos de la diagonal de J_2 son

$$\frac{\partial P_p}{\partial f_p} = e_p B_{pp} + 2f_p G_{pp} - e_p B_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (f_q G_{pq} + e_q B_{pq}) \quad (4.17)$$

La componente imaginaria de corriente, la ecuación (4.16) es sustituida en la ecuación (4.17) y se obtiene

$$\frac{\partial P_p}{\partial f_p} = -e_p B_{pp} + f_p G_{pp} + d_p$$

La ecuación de la potencia reactiva (4.12) es

$$Q_p = f_p (e_p G_{pp} - f_p B_{pp}) - e_p (f_p G_{pp} + e_p B_{pp}) + \sum_{\substack{q=1 \\ q \neq p}}^n \{f_p (e_q G_{pq} - f_q B_{pq}) - e_p (f_q G_{pq} + e_q B_{pq})\} \quad (4.18)$$

Derivando, los elementos fuera de la diagonal de J_3 son

$$\frac{\partial Q_p}{\partial f_q} = -e_p B_{pq} + f_p G_{pq} \quad q \neq p$$

y los elementos de la diagonal de J_3 son

$$\frac{\partial Q_p}{\partial e_p} = f_p G_{pp} - f_p G_{pp} - 2e_p B_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (f_q G_{pq} + e_q B_{pq}) \quad (4.19)$$

Reemplazando la ecuación (4.16) en la ecuación (4.19) tenemos

$$\frac{\partial Q_p}{\partial e_p} = f_p G_{pp} - e_p B_{pp} - d_p$$

De la ecuación (4.18), los elementos fuera de la diagonal de J_4 son:

$$\frac{\partial Q_p}{\partial f_q} = -e_p G_{pq} - f_p B_{pq} \quad q \neq p$$

y los elementos de la diagonal de J_4 son

$$\frac{\partial Q_p}{\partial f_q} = e_p G_{pp} - 2f_p B_{pp} - e_p G_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (e_q G_{pq} - f_q B_{pq}) \quad (4.20)$$

Reemplazando la ecuación (4.15) en la ecuación (4.20) para obtener

$$\frac{\partial Q_p}{\partial f_q} = -f_p B_{pp} - e_p G_{pp} + c_p$$

Dado un conjunto inicial de tensiones de barra, la potencia activa y reactiva son calculadas de las ecuaciones (4.12). La diferencia entre la potencia programada y la calculada es

$$\begin{aligned} \Delta P_p^k &= P_{p(\text{programado})} - P_p^k \\ \Delta Q_p^k &= Q_{p(\text{programado})} - Q_p^k \quad p = 1, 2, \dots, n-1 \end{aligned}$$

Las tensiones de barra estimadas y calculadas son usadas con el fin de evaluar los elementos del Jacobiano. El conjunto de ecuaciones lineales (4.13) puede ser resuelto para

$$\begin{aligned} e_p^{k+1} &= e_p^k + \Delta e_p^k \\ f_p^{k+1} &= f_p^k + \Delta f_p^k \end{aligned}$$

El proceso es repetido hasta que ΔP_p^k y ΔQ_p^k estén dentro de una tolerancia especificada. La secuencia de pasos para la solución del flujo de potencia mediante el método de Newton-Raphson se muestra en la figura 4.3.

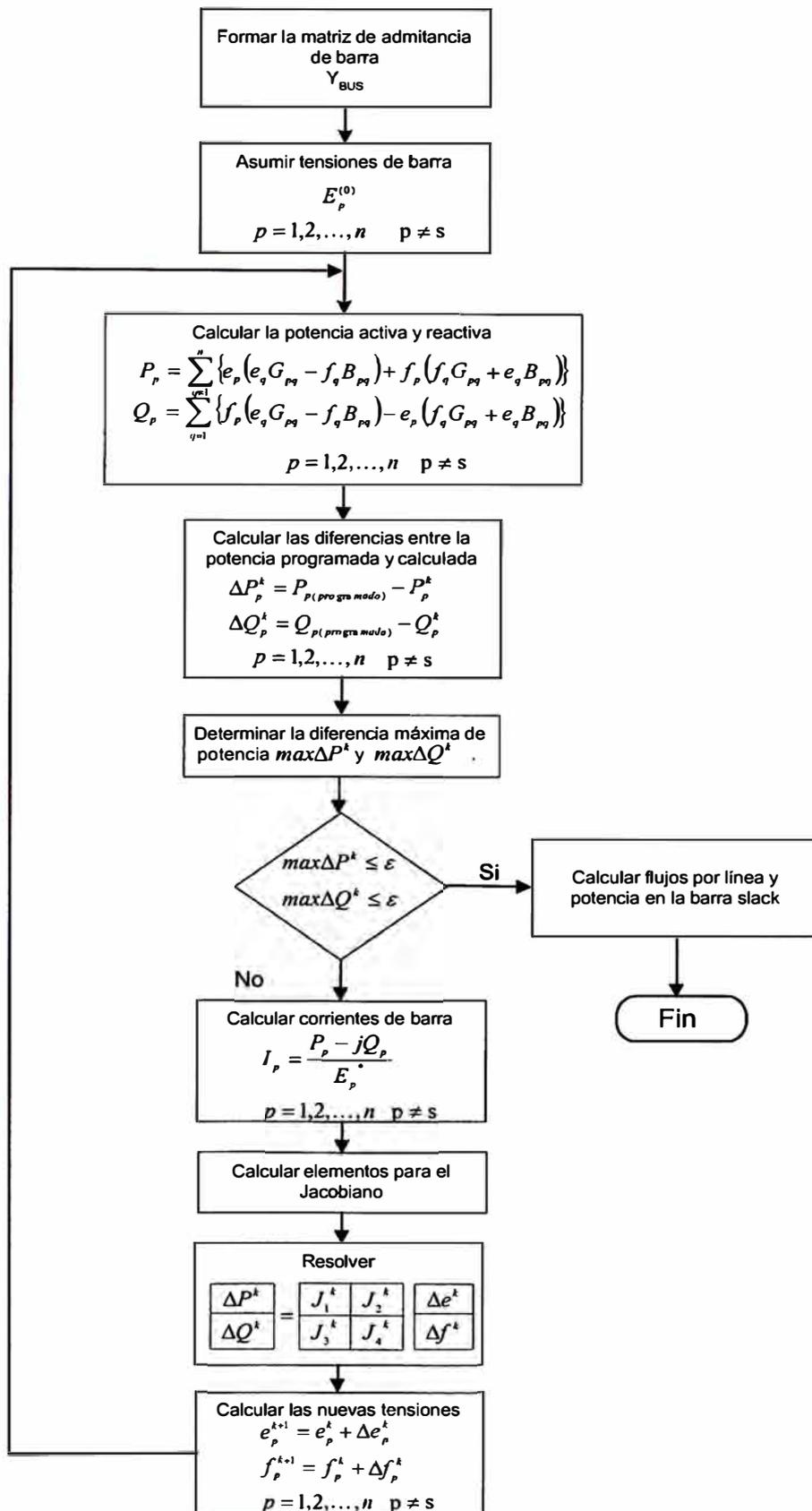


Figura 4.3 Diagrama de flujo de la técnica de Newton- Raphson sin barras PV

Barras de tensión controlada

Una modificación de los procedimientos computacionales normales es requerida para tomar en cuenta las barras de tensión controlada. En estas barras la magnitud de la tensión y la potencia activa son especificadas.

Los valores de e_p y f_p deben satisfacer la relación

$$e_p^2 + f_p^2 = |E_p|_{(\text{programado})}^2 \quad (4.21)$$

con el fin de calcular la potencia reactiva requerida para proveer la tensión de barra programada.

Entonces, para una barra PV, las ecuaciones utilizadas son:

$$P_p = \sum_{q=1}^n \{e_p (e_q G_{pq} + f_q B_{pq}) + f_p (f_q G_{pq} - e_q B_{pq})\}$$

y

$$|E_p|^2 = e_p^2 + f_p^2 \quad (4.22)$$

donde la ecuación (4.22) reemplaza la potencia reactiva en la ecuación (4.12). La

ecuación matricial del Jacobiano sufre unos cambios y queda

$$\begin{array}{|c|} \hline \Delta P \\ \hline \Delta Q \\ \hline \Delta |E|^2 \\ \hline \end{array} = \begin{array}{|c|c|} \hline J_1 & J_2 \\ \hline J_3 & J_4 \\ \hline J_5 & J_6 \\ \hline \end{array} \begin{array}{|c|} \hline \Delta e \\ \hline \Delta f \\ \hline \end{array}$$

Los elementos de las submatrices J_1 , J_2 , J_3 y J_4 ya se conocen; sólo faltan definir los elementos de las submatrices J_5 y J_6 .

De la ecuación (4.22), los elementos fuera de la diagonal de J_5 ,

$$\frac{\partial |E_p|^2}{\partial e_q} = 0 \quad q \neq p$$

y los elementos de la diagonal son

$$\frac{\partial |E_p|^2}{\partial e_p} = 2e_p \quad (4.23)$$

Similarmente, los elementos fuera de la diagonal de J_6 son

$$\frac{\partial |E_p|^2}{\partial f_q} = 0 \quad q \neq p$$

y los elementos de la diagonal son

$$\frac{\partial |E_p|^2}{\partial f_p} = 2f_p \quad (4.24)$$

La diferencia en el cuadrado de la magnitud de la tensión en la barra p es

$$\Delta |E_p^k|^2 = |E_p^k|_{(programado)}^2 - |E_p^k|^2$$

Si la potencia reactiva calculada Q_p^k excede la capacidad máxima $Q_{p(max)}$ la potencia reactiva se ajusta a ese valor y si es menor que la mínima se hace lo mismo -y la barra PV, se transforma en una barra PQ.

El diagrama de flujo incluyendo las barras PV se observa en la figura 4.4

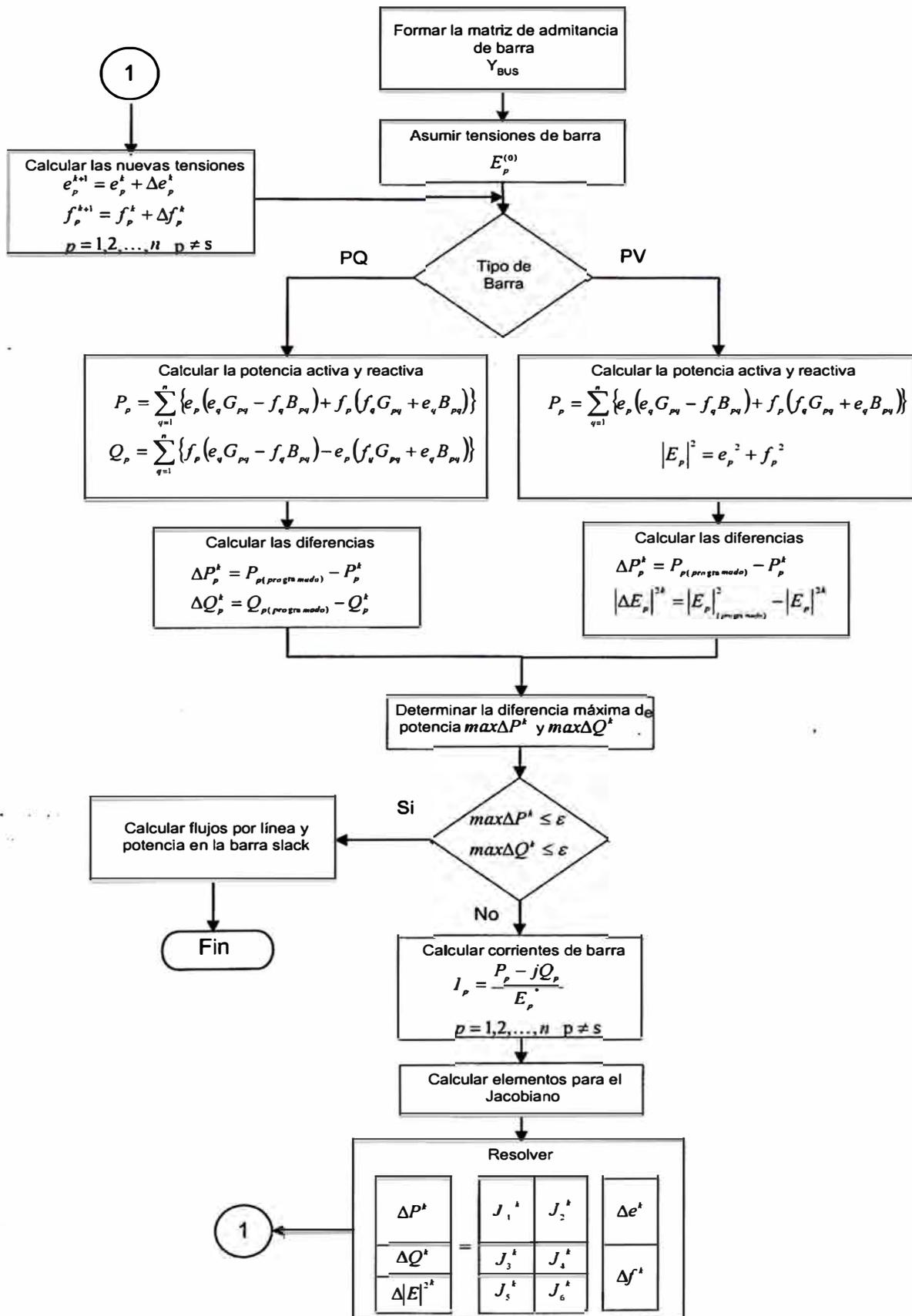


Figura 4.4 Diagrama de flujo de la técnica de Newton-Raphson con barras PV

4.5 Representación matricial y solución del sistema de ecuaciones de lineales

4.5.1 Selección del formato de estructura de datos

La matriz de admitancia de barra de una red tiene tres características básicas:

- Esparcida
- Simétrica
- Diagonalmente dominante

El jacobiano cuando solo tiene barras PQ, tiene cuatro veces la estructura de la matriz de admitancia de barra, entonces se tendría cuatro veces el número de elementos no nulos de Y_{BUS} sin incluir la barra slack. A pesar de que el Jacobiano no pierde su esparcidad pierde la simetría y deja de ser diagonalmente dominante, en el caso de que el sistema contenga barras PV, como ocurre normalmente, el número de elementos no nulos, disminuye conforme aumente el número de barras PV.

Para examinar cual es el posible formato de matriz esparcida que se usara, se verán los usos que estas tienen:

- COO – Estructura de datos más flexible cuando se construye o modifica una matriz esparcida
- CSC – Estructura de datos natural para muchas operaciones comunes de matrices incluyendo multiplicación y construcción o solución de factores triangulares esparcidos.
- CSR – Estructura de datos natural para muchas operaciones comunes de matrices incluyendo multiplicación y construcción o solución de factores triangulares esparcidos.

- DIA – Particularmente muy usado para las matrices que se derivan de aproximaciones de diferencias finitas de ecuaciones diferenciales parciales en grillas uniformes.
- ELL – Apropiado para aproximaciones de elementos o volúmenes finitos en ecuaciones diferenciales parciales donde los elementos son del mismo tipo, pero la grilla es irregular.
- JAD – Apropiado para matrices que son altamente irregulares o para una multiplicación de matrices de propósito general donde las propiedades de la matriz no son conocidas *a priori*.
- SKY – Particularmente mejor preparado para Cholesky o descomposición LU cuando el pivoteo no es requerido. En este caso, todo el llenado ocurrirá dentro de la estructura no nula.

Se observa que para nuestro caso los formatos que más se aplican son el CSC y el CSR, además que es fácil de intercambiarlos y el formato CSC es usado por la colección Harwell-Boeing.

El llenado de la matriz de admitancia se hizo en el formato CSR. Luego de tener la matriz de admitancia de barra, se procede a calcular el Jacobiano como se indico en los acápites anteriores, utilizando el formato CSR de la matriz de admitancia de barras. Ahora se tiene que resolver el sistema de ecuaciones lineales planteado, utilizando un método preconditionado, los preconditionadores que se pueden utilizar son varios, entre ellos se tiene ILU(factorización LU incompleta), IC(factorización de Cholesky incompleta), D(inversa de la diagonal), ILQ(factorización LQ incompleta). En el presente proyecto se ha utilizado el preconditionador ILU, debido a que es el más conveniente para este sistema, como se dijo anteriormente el

jacobiano no es simétrico y no es diagonalmente dominante, con lo cual se descarta IC y D respectivamente. Ahora bien ILQ es una factorización ortogonal requiere de un tiempo mayor que la ILU, adicionalmente el ILU es más simple que el ILQ. A continuación se explicará el preconditionador ILU y luego el método en sí para resolver el sistema de ecuaciones lineales.

4.5.2 Factorización LU

Supóngase que A se puede factorizar como el producto de una matriz triangular inferior L con una matriz triangular superior U : $A=LU$. En este caso, para resolver el sistema de ecuaciones $Ax=b$ se puede proceder por etapas como sigue:

Resolver $Lz = b$ para z
 Resolver $Ux=z$ para x .

La factorización $A=LU$ se puede llevar a cabo con el supuesto de que en el cálculo no aparecen divisores igual a cero. No toda matriz tiene una factorización de esa índole. Comenzamos con una matriz A de $n \times n$ y buscamos matrices

$$L = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix} \quad \text{y} \quad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix},$$

tales que

$$A=LU \tag{4.25}$$

Cuando esto es posible, se dice que A tiene una **descomposición LU**. Se puede ver que las ecuaciones (4.25) *no* determinan a L y U en forma única. De hecho para cada i , se puede asignar un valor distinto de cero a l_{ii} o a u_{ii} (mas no a ambos). Por ejemplo, una elección simple es fijar $l_{ii}=1$ para $i=1, 2, \dots, n$, haciendo de este modo a L una matriz *triangular inferior unitaria*. Otra elección obvia es hacer de U una

matriz *triangular superior unitaria* ($u_{ii}=1$ para cada i). Estos casos revisten una particular importancia.

Para deducir un algoritmo para la factorización LU de A , se comienza con la fórmula para la multiplicación de matrices:

$$a_{ij} = \sum_{s=1}^n l_{is} u_{sj} = \sum_{s=1}^{\min(i,j)} l_{is} u_{sj} \quad (4.26)$$

Aquí nos hemos valido del hecho de que $l_{is}=0$ para $s>i$ y $u_{sj}=0$ para $s>j$.

En este proceso, cada paso determina una nueva fila de U y una nueva columna de L . En el paso k , podemos suponer que ya se calcularon las filas 1, 2, ..., $k-1$ de U , al igual que las columnas 1, 2, ..., $k-1$ de L . Haciendo $i=j=k$ en la ecuación (4.26) se obtiene:

$$a_{kk} = \sum_{s=1}^{k-1} l_{ks} u_{sk} + l_{kk} u_{kk} \quad (4.27)$$

Si u_{kk} o l_{kk} se han especificado, utilizamos la ecuación (4.27) para determinar a la otra. Conocidas u_{kk} y l_{kk} , utilizamos la ecuación (4.26) para escribir para la k -ésima fila ($i=k$) y la k -ésima columna ($j=k$), respectivamente,

$$a_{kj} = \sum_{s=1}^{k-1} l_{ks} u_{sj} + l_{kk} u_{kj} \quad (k+1 \leq j \leq n) \quad (4.28)$$

$$a_{ik} = \sum_{s=1}^{k-1} l_{is} u_{sk} + l_{ik} u_{kk} \quad (k+1 \leq i \leq n) \quad (4.29)$$

Si $l_{kk} \neq 0$, la ecuación (4.28) se puede utilizar para encontrar los elementos u_{kj} . Del mismo modo, si $u_{kk} \neq 0$, la ecuación (4.29) se puede utilizar para obtener los elementos

El algoritmo basado en el análisis precedente se llama **factorización de Doolittle** cuando L es triangular inferior unitaria ($l_{ii}=1$ para $1 \leq i \leq n$) y **factorización de Crout** cuando U es triangular superior unitaria ($u_{ii}=1$ para $1 \leq i \leq n$). Cuando $U=L^T$ de modo que $l_{ii}=u_{ii}$ para $1 \leq i \leq n$, el algoritmo se llama **factorización de Cholesky**.

El pseudocódigo para llevar a cabo la factorización de Doolittle es el siguiente:

```

input n,(aij)
for k=1, 2, ..., n do
  lkk ← 1
  for j=k, k+1, ..., n do
    ukj ← akj - ∑s=1k-1 lks usj
  end
  for i=k+1, k+2, ..., n do
    lik ← (aik - ∑s=1k-1 lks usj) / ukk
  end
end
output (lij), (uij)

```

Ahora se va a dar una condición suficiente para que una matriz cuadrada A tenga una descomposición LU : “Si los n menores principales de la matriz A de $n \times n$ son no singulares, entonces la matriz A tiene una descomposición LU ”.

4.5.3 Factorización incompleta

Se conoce como factorización incompleta cuando durante el proceso de factorización aparecen ciertos elementos no nulos en lugares donde la matriz original tenía ceros, que han sido ignorados. Tal preconditionador es luego dado en forma factorizada $M=LU$ con L y U triangular inferior y superior respectivamente. La eficacia del preconditionador depende de que tan bien M^{-1} se aproxime a A^{-1} .

Como se dijo anteriormente las factorizaciones incompletas pueden ser dadas en varias formas. Para nuestro caso $M=LU$, para resolver este sistema se procede como se ve en la figura (4.5).

<p>Dado y hacer $M=LU$ for $i=1, 2, \dots$ do</p> $z_i = l_{ii}^{-1} \left(y_i - \sum_{j<i} l_{ij} z_j \right)$ <p>for $i= n, n-1, n-2, \dots$ do</p> $x_i = u_{ii}^{-1} \left(z_i - \sum_{j>i} u_{ij} x_j \right)$
--

Figura 4.5 : Resolver preconditionador del sistema $Mx=y$, con $M=LU$.

El tipo más común de factorización incompleta es basada en tomar un conjunto S de posiciones de la matriz, y guardando todas las posiciones fuera de este conjunto igual a cero durante la factorización. La factorización resultante es incompleta en el sentido que el llenado es suprimido.

El conjunto S es usualmente escogido para comprender todas las posiciones (i, j) para todo $a_{i, j} \neq 0$. Una posición que es cero en A pero no lo es en la factorización exacta es llamada posición *llena*, y si esta fuera de S , esta posición es “descartada”. A veces, S coincide con el conjunto de elementos no nulos en A , descartando el llenado. Esta factorización es llamada la factorización $ILU(0)$: la factorización incompleta LU de nivel cero.

4.5.4 Solución iterativa del sistema de ecuaciones lineales

Para la solución del sistema de ecuaciones lineales se ha usado un método con un preconditionador, se han probado los siguientes métodos:

Gradiente Biconjugado (BiCG)

Gradiente Biconjugado estabilizado (BiCGSTAB)

Gradiente conjugado cuadrado (CGS)

Residual cuasi-mínimo (QMR)

En la tabla 4.3 se muestra un resumen de las operaciones, para los métodos probados y en la tabla 4.4 se muestran los requerimientos de almacenamiento para el método.

Método	Producto Interno	SAXPY	Producto Matriz-Vector	Solucionador Precondicionado
BiCG	2	5	1/1	1/1
QMR	2	$8+4^{ab}$	1/1	1/1
CGS	2	6	2	2
BiCGSTAB	4	6	2	2

Tabla 4.3 resumen de operaciones por iteración i . “a/b” significa “a multiplicaciones con la matriz y “b” con su transpuesta.

a Verdaderas operaciones SAXPY +escalamientos de vectores.

b Menos para implementaciones que no actualiza recursivamente el residual

Método	Requerimientos de almacenamiento
BiCG	Matriz + 10n
QMR	Matriz + 16n ^a
CGS	Matriz + 11n
BiCGSTAB	Matriz + 10n

Tabla 4.4 Requerimientos de almacenamiento para los métodos en la iteración i : n denota el orden de la matriz.

a Menos para implementaciones que no actualizan recursivamente el residual.

Ahora se ha probado, para ver el número de iteraciones, en un sistema de 5 barras con una barra PV y un sistema de nueve barras con 2 barras PV, de allí obtenemos la tabla 4.5. Adicionalmente se ha hecho un movimiento en el jacobiano (JJ1), invirtiendo J1 y J2, J3 y J4, J5 y J6 para lograr un mejor funcionamiento del preconditionador.

Caso	Jacobiano	BiCG	BiCGSTAB	CGS	QMR
5 barras	JJ1	2	1	1	8
	JJ	3	200 ^(*)	2	8
9 barras	JJ1	11	200 ^(*)	8	17
	JJ	$x^{(+)}$	$x^{(+)}$	$x^{(+)}$	$x^{(+)}$

Tabla 4.5 Número de iteraciones de los métodos

(*) No cumple con la tolerancia

(+) El método falla

Finalmente, se ha optado por el método CGS con la matriz jacobiana invertida, que es la que mejor funciona. A continuación se explica el método CGS

En el BiCG, el vector residual $r^{(i)}$ puede considerarse como el producto de $r^{(0)}$ y un polinomio de grado i en A , esto es

$$r^{(i)} = P_i(A)r^{(0)} \quad (4.30)$$

El mismo polinomio satisface $\tilde{r}^{(i)} = P_i(A)\tilde{r}^{(0)}$ así que

$$\rho_i = (\tilde{r}^{(i)}, r^{(i)}) = (P_i(A^T)\tilde{r}^{(0)}, P_i(A)r^{(0)}) = (\tilde{r}^{(0)}, P_i^2(A)r^{(0)}) \quad (4.31)$$

Esto sugiere que $P_i(A)$ reduce a $r^{(0)}$ a un vector pequeño $r^{(i)}$, luego puede ser ventajoso aplicar esta operador de “contracción” dos veces, y computa $P_i^2(A)r^{(0)}$. La ecuación (4.31) muestra que los coeficientes de iteración pueden aún ser recuperados de estos vectores, y volverse fácil encontrar las correspondientes aproximaciones para x . Esta aproximación nos guía al método del gradiente conjugado cuadrado. El pseudocódigo para este método es el siguiente:

```

input  $x^{(0)}, M, b, A, tol$ 
 $r^{(0)} = b - Ax^{(0)}$ 
 $\tilde{r} = r^{(0)}$ 
for  $i=1, 2, \dots, n$  do
   $\rho_{i-1} = \tilde{r}^T r^{(i-1)}$ 
  if  $\rho_{i-1} = 0$  método falla
  if  $i=1$ 
     $u^{(1)} = r^{(0)}$ 
     $p^{(1)} = u^{(1)}$ 
  else
     $\beta_{i-1} = \rho_{i-1} / \rho_{i-2}$ 
     $u^{(i)} = r^{(i-1)} + \beta_{i-1}q^{(i-1)}$ 
     $p^{(i)} = u^{(i)} + \beta_{i-1}(q^{(i-1)} + \beta_{i-1}p^{(i-1)})$ 
  endif
  solve  $M\hat{p} = p^{(i)}$ 
   $\hat{v} = A\hat{p}$ 

```

```

$$\alpha_i = \rho_{i-1} / \tilde{r}^T \vartheta$$

$$q^{(i)} = u^{(i)} - \alpha_i \vartheta$$
solve  $M\hat{u} = u^{(i)} + q^{(i)}$ 
$$x^{(i)} = x^{(i-1)} + \alpha_i \hat{u}$$

$$\hat{q} = A\hat{u}$$

$$r^{(i)} = r^{(i-1)} + \alpha_i \hat{q}$$
check tol; continue if necessaryendoutput ( $x^{(i)}$ )
```

CAPÍTULO V OPTIMIZACIÓN DE POTENCIA DE GENERACIÓN

5.1 Formulación del problema

Para formular el problema y su solución matemática, se introduce la siguiente notación:

P_T	potencia eléctrica de la central térmica T
P_T^0	potencia eléctrica de la central térmica T en la hora previa
N_T	número de centrales térmicas
a_T, b_T, c_T	coeficientes de costo del bloque térmico T
P_T^{min}	generación mínima de la unidad T
P_T^{max}	generación máxima de la unidad T
DR_i	límite de la rampa positiva de la unidad T
UR_i	límite de la rampa negativa de la unidad T
P_{pz}^-, P_{pz}^+	márgenes de una zona prohibida
	número de centrales hidroeléctricas
P_w	potencia eléctrica del generador dada por la turbina w
P_D	potencia de la carga en el intervalo t
P_{loss}	potencia perdida
η_w	eficiencia de la turbina w

n_s	número específico de revoluciones
q_w	caudal que ingresa a la turbina
h_w	altura para la turbina w
ρ	densidad del agua
g	aceleración de la gravedad
q_w^{min}	caudal mínimo
q_w^{max}	caudal máximo
V_r	volumen del reservorio r
V_r^0	volumen del reservorio en la hora previa
r_r	caudal que ingresa al reservorio
s_r	derramamiento del reservorio
V_r^{tar}	volumen deseado del reservorio r

La curva entrada-salida de una unidad térmica establece la relación entre la energía que ingresa (Mbtu/h) y la energía neta que sale (MW). Al multiplicar el poder calorífico por el precio del combustible se obtiene la relación de costo vs MW, que en el presente trabajo se aproxima mediante una curva de segundo orden. El problema es minimizar la función costo que en forma matemática se tendría:

$$\text{PB0: } \left\{ \begin{array}{l} \text{Problema: Minimizar } F = \sum_{T=1}^{N_T} f_T(P_T) \\ \text{Sujeto a: } \sum_{T=1}^{N_T} P_T + \sum_{w=1}^{N_w} P_w = P_D + P_{loss} \\ P_T^{min} \leq P_T \leq P_T^{max} \end{array} \right. \quad (5.1)$$

En (5.1), la función de costo de generación se expresa así:

$$f_T(P_T) = a_T P_T^2 + b_T P_T + c_T \quad (5.2)$$

donde a_T , b_T y c_T son constantes.

5.2 Modelamiento de las pérdidas en la red

Las pérdidas en transmisión que son una parte del costo de los requerimientos por alimentar a un sistema eléctrico, deben ser tomadas en cuenta para alcanzar un verdadero despacho económico. Dos son los métodos que se usan en general para incluir las pérdidas. El primero de estos involucra el uso de los factores de penalidad para ajustar el costo incremental de la potencia en la barra de generación al costo incremental de la potencia entregada a la carga. El segundo método envuelve el uso de los coeficientes B.

En este caso se usará los factores de penalidad y las pérdidas se calcularán directamente del flujo de potencia, para el cálculo de los factores de penalidad se usa la siguiente metodología:

Los factores de penalidad se calculan utilizando:

$$Pf_i = \left(\frac{1}{1 - \frac{\partial P_{loss}}{\partial P_i}} \right) \quad (5.3)$$

Una alternativa aproximada al despacho económico es usar una barra de referencia que siempre se mueve cuando se hace un incremento de la generación. Si se supone un cambio en la generación en la barra i de ΔP_i ,

$$P_i^{new} = P_i^{old} + \Delta P_i \quad (5.4)$$

Además, si se asume que la *carga se queda constante* y que para compensar el incremento en ΔP_i , la barra de referencia caerá en ΔP_{slack} ,

$$P_{slack}^{new} = P_{slack}^{old} + \Delta P_{slack} \quad (5.5)$$

Si nada más ha cambiado, ΔP_{slack} sería el negativo de ΔP_i ; además, el flujo en el sistema puede cambiar como resultado de los dos ajustes de la generación. El cambio en el flujo es capaz de producir un cambio en las pérdidas tal que ΔP_{slack} no es necesariamente igual a ΔP_i . Esto es,

$$\Delta P_{slack} = -\Delta P_i + \Delta P_{loss} \quad (5.6)$$

al dividir (5.6) entre el cambio de ΔP_i , se tiene:

$$\begin{aligned} \frac{\Delta P_{slack}}{\Delta P_i} &= \frac{-\Delta P_i + \Delta P_{loss}}{\Delta P_i} = -1 + \frac{\Delta P_{loss}}{\Delta P_i} \\ -\frac{\partial P_{slack}}{\partial P_i} &= 1 - \frac{\partial P_{loss}}{\partial P_i} = \frac{1}{Pf_i} \end{aligned} \quad (5.7)$$

El cambio incremental de la barra *slack* puede ser expresado como sigue

$$\frac{\partial P_{slack}}{\partial P_i} = \frac{\partial P_{slack}}{\partial E_k} \frac{\partial E_k}{\partial P_i} \quad k = 1, 2, \dots, n-1 \quad (5.8)$$

donde E representa los voltajes complejos ya sea en su forma rectangular (e, f) o en su forma polar $V \angle \theta$.

Similar a (5.8) se obtiene:

$$\frac{\partial P_{slack}}{\partial Q_i} = \frac{\partial P_{slack}}{\partial E_k} \frac{\partial E_k}{\partial Q_i} \quad k = 1, 2, \dots, n-1 \quad (5.9)$$

Combinamos (5.8) y (5.9) en una sola ecuación matricial

$$\begin{bmatrix} \frac{\partial P_{slack}}{\partial P} & \frac{\partial P_{slack}}{\partial Q} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_{slack}}{\partial e} & \frac{\partial P_{slack}}{\partial f} \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial P} & \frac{\partial e}{\partial Q} \\ \frac{\partial f}{\partial P} & \frac{\partial f}{\partial Q} \end{bmatrix} \quad (5.10)$$

Cuando invertimos (5.10) se tiene:

$$\begin{bmatrix} \frac{\partial P_{slack}}{\partial e} & \frac{\partial P_{slack}}{\partial f} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_{slack}}{\partial P} & \frac{\partial P_{slack}}{\partial Q} \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial e} & \frac{\partial P}{\partial f} \\ \frac{\partial Q}{\partial e} & \frac{\partial Q}{\partial f} \end{bmatrix}$$

Al transponer la ecuación anterior se obtiene:

$$\begin{bmatrix} \frac{\partial P_{slack}}{\partial e} \\ \frac{\partial P_{slack}}{\partial f} \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial e} & \frac{\partial Q}{\partial e} \\ \frac{\partial P}{\partial f} & \frac{\partial Q}{\partial f} \end{bmatrix} \begin{bmatrix} \frac{\partial P_{slack}}{\partial P} \\ \frac{\partial P_{slack}}{\partial Q} \end{bmatrix} \quad (5.11)$$

Observando la ecuación (5.11), se aprecia que la matriz de coeficientes es la transpuesta del Jacobiano para las barras PQ (J1, J2, J3', J4'). Además de la ecuación (4.14) se desprende:

$$\frac{\partial P_{slack}}{\partial e_k} = e_{slack} G_{nk} + f_{slack} B_{nk} \quad k = 1, 2, \dots, n-1 \quad (5.12)$$

De la ecuación (4.17) se desprende:

$$\frac{\partial P_{slack}}{\partial f_k} = -e_{slack} B_{nk} + f_{slack} G_{nk} \quad k = 1, 2, \dots, n-1 \quad (5.13)$$

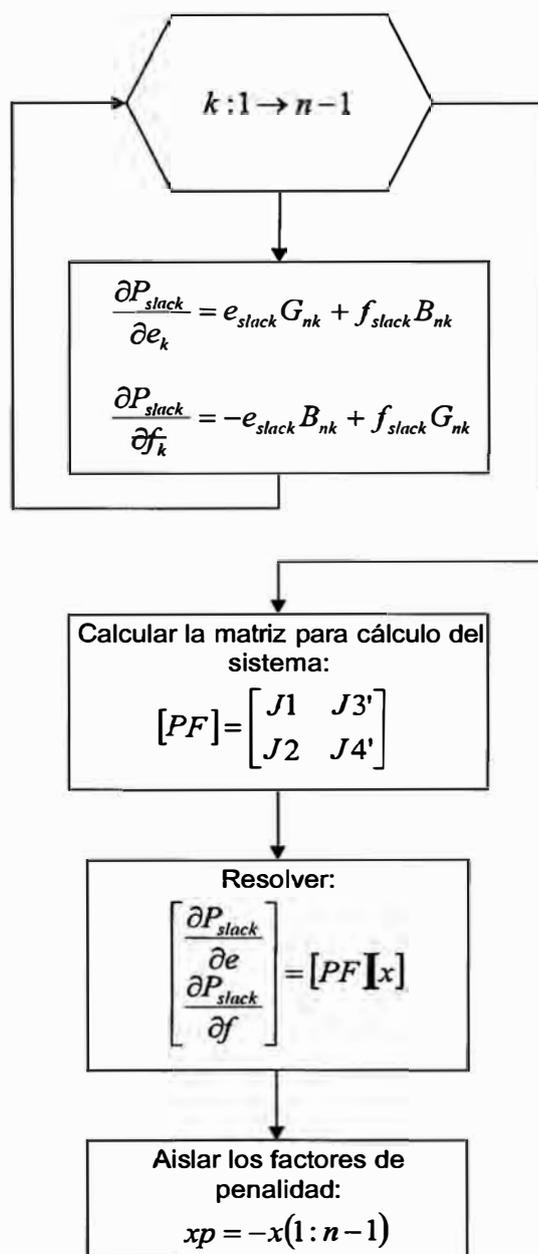


Figura 5.1 Diagrama de flujo de factor de penalidad

5.3 Límites de incremento y decremento de generación

En la actualidad se consideran los efectos de los límites de la rampa en el compromiso de unidades y en el despacho económico. Normalmente se hace la simplificación de que cada unidad puede ser ajustada instantáneamente, sin embargo esto no refleja la operación actual de la unidad generadora.

El rango de operación de todas las unidades en línea es restringido por sus límites de rampa. La Fig. 5.2 muestra las tres posibles situaciones cuando una unidad está en línea desde una hora $t-1$ a una hora t . Fig. 5.2(a) muestra que la unidad está en una operación en estado estacionario, mientras que en la Fig. 5.2(b) muestra que la unidad está en un estado de incremento de la potencia generada y en la Fig. 5.2(c) se aprecia que la unidad está en un estado de decremento de la potencia generada.

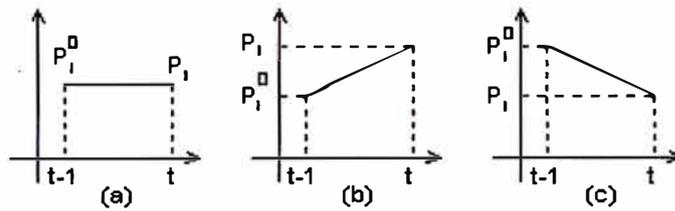


Fig. 5.2. Tres situaciones posibles de una unidad en línea

Luego, las restricciones debidas a los límites de la rampa están dadas por:

1) si se incrementa la generación

$$P_T - P_T^0 \leq UR_T \quad (5.12)$$

2) si se disminuye la generación

$$P_T^0 - P_T \leq DR_T \quad (5.13)$$

Los límites de la rampa UR_T y DR_T están en MW/h. En la práctica, DR_T es más grande que UR_T . Combinando (5.5), (5.6), y (5.1), el problema de optimización restringido es modificado como:

$$PBI : \left\{ \begin{array}{l} \text{Problema: Minimizar } F = \sum_{T=1}^{N_T} f_T(P_T) \\ \text{Sujeto a: } \sum_{T=1}^{N_T} P_T + \sum_{W=1}^{N_W} P_W = P_D + P_{loss} \\ \text{Max}(P_T^{min}, P_T^0 - DR_T) \leq P_T \leq \text{Min}(P_T^{max}, P_T^0 + UR_T) \end{array} \right. \quad (5.14)$$

5.4 Zonas de operación prohibida

En la Fig. 5.3 se muestra la curva típica de una unidad térmica. El rizado de la curva de entrada-salida es el resultado de las pérdidas que se producían cuando una válvula de admisión de vapor se empieza a abrir. Fig. 5.4 presenta la curva incremental de la entrada (pendiente de la curva de la Fig. 5.3).

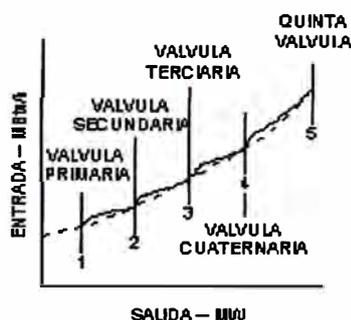


Fig. 5.3. Unidad Térmica – Entrada vs Salida

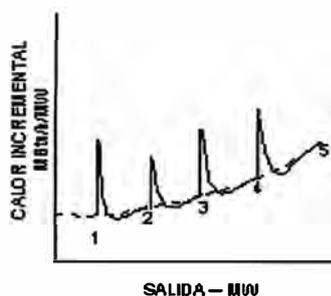


Fig. 5.4. Unidad Térmica – Tasa incremental de calor

Los rizados en la Fig.5.3 se convierten en puntas en la Fig. 5.4. La forma de la curva de entrada-salida en la vecindad de los puntos de la entrada de válvulas es difícil de determinar, por eso es que se trata de evitar la operación en estas áreas.

Un sistema de despacho económico debe reconocer los efectos de los puntos de la válvula, de manera que propicie que las generaciones se muevan rápidamente a través de dichas áreas. En este estudio para evitar este problema se usa la curva que se muestra en la Fig. 5.5.

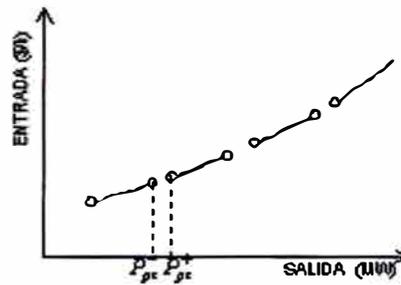


Fig. 5.5. Curva Aproximada

5.5 Modelamiento de las centrales hidroeléctricas

La potencia eléctrica de los generadores es dada por las turbinas de agua y puede ser formulada por funciones no lineales, los cuales dependen de la altura y de la eficiencia. La eficiencia a su vez depende de la altura y de la descarga de agua de la turbina. Esto es descrito por una familia de curvas.

$$P_w = \eta_w(q_w, h_w) \cdot \rho \cdot g \cdot h_w \cdot q_w \quad (5.15)$$

En la figura 5.6 se muestran las curvas de eficiencias de los diversos tipos de turbinas. Estas curvas corresponden:

- la curva *a*, a una turbina *Pelton* de $n_s=20$;
- la curva *b*, a una turbina *Kaplan* de $n_s=500$;
- la curva *c*, a una turbina *Francis* normal de $n_s=250$;
- la curva *d*, a una turbina *Francis* rápida de $n_s=500$;
- la curva *e*, a una turbina *hélice* de $n_s=650$;
- la curva *f*, a una turbina *hélice muy rápida* de $n_s=1050$;

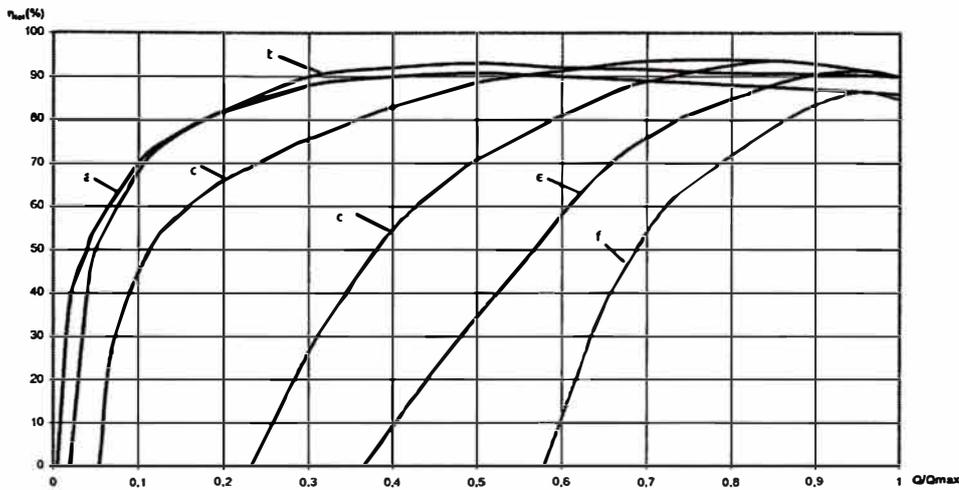


Fig 5.6 Curvas de eficiencia

La descarga de agua de las turbinas es restringida por los límites superior e inferior:

$$q_w^{min} \leq q_w \leq q_w^{max} \quad (5.16)$$

Se considera ahora, un sistema hidráulicamente acoplado que consiste en una serie de reservorios (ver figura 5.7). La descarga del reservorio de arriba se asume que fluye directamente a la planta de abajo sin un retardo de tiempo. Las ecuaciones que se generan son las siguientes:

$$V_r = V_r^0 + r_r - q_r - s_r \quad (5.17)$$

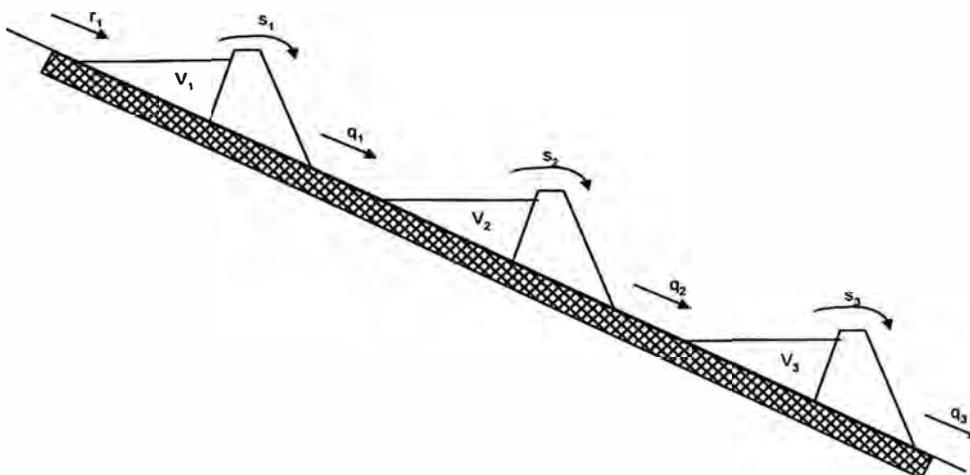


Fig. 5.7 Centrales en serie

La altura del agua de la turbina puede ser calculada como función del volumen de agua almacenada y depende de la forma del reservorio:

$$h_w = f(V_r) \quad (5.18)$$

Adicionalmente se tiene un volumen objetivo para el siguiente intervalo de tiempo:

$$V_r \geq V_r^{tar} \quad (5.19)$$

Como el flujo de potencia es instantáneo, lo que se busca es obtener la mayor potencia posible, para el caudal disponible. El caudal disponible se obtiene de la ecuación (5.17), De acuerdo a las curvas de eficiencias y de la ecuación (5.15), se puede decir que el caudal versus potencia, se puede representar por la siguiente función:

$$\begin{aligned} q_w &= b1P_w + c1 && \text{para } 0 \leq P_w \leq P_w^{max1} \\ q_w &= a2P_w^2 + b2P_w + c2 && \text{para } P_w^{max1} \leq P_w \leq P_w^{max2} \end{aligned} \quad (5.20)$$

El problema a optimizar queda de la siguiente manera:

$$PB2: \left\{ \begin{array}{l} \text{Problema: Maximizar } F = \sum_{w=1}^{N_w} (P_w) \\ \text{Sujeto a: } \sum_{w=1}^{N_w} q_w = q_{disp} \end{array} \right. \quad (5.21)$$

5.6 Optimización mediante algoritmo genético

5.6.1 Optimización de la potencia hidráulica

El programa se inicia maximizando la potencia hidráulica. Se usan cadenas de longitud variable, se obtiene una buena precisión con cadenas de longitud múltiplos de 10, Como ejemplo se codifica una cadena de longitud 10, que contiene dos grupos:

$$\begin{array}{cccccc|cccc}
 d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\
 x & x & x & x & x & x & x & x & x & x \\
 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0
 \end{array} \quad (5.22)$$

Grupo2
Grupo1

donde $d_i \in \{0,1\}$ $i = 1,2,\dots,10$

De la decodificación se obtiene para cada grupo una variable normalizada P_{nm1} y P_{nm2} , luego se procede a calcular la potencia real:

$$P = P_{nm} * \left(\frac{P^{max2}}{2^{l/NumG-1}} \right) \quad (5.23)$$

donde l es la longitud del cromosoma y $NumG$ es el número de grupos.

Una vez que se tienen estas variables, se procede a calcular los caudales respectivos. Finalmente, se calcula la función peso, que se eligió de la siguiente manera:

$$fitness = \frac{\sum_{i=0}^{N_H} P_{nmi} - K_H * abs \left(q_{disp} - \sum_{j=0}^{N_H} q_j \right)}{q_{max}} \quad (5.24)$$

5.6.2 Optimización de la potencia térmica

La aproximación propuesta usa el costo incremental como base λ . Se normaliza el costo incremental a λ^{nm} , donde $0 \leq \lambda^{nm} \leq 1$.

Con esta codificación el número de bits del cromosoma es enteramente independiente del número de unidades. Esto es importante para sistemas con gran número de generadores.

La precisión de la solución depende de cuantos bits son usados para representar a λ^{nm} . El efecto de aumentar el número de bits se traduce en una convergencia muy lenta. En el programa se usan 10 bits.

$$\begin{array}{cccccccccc}
d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\
x & x & x & x & x & x & x & x & x & x \\
2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} & 2^{-5} & 2^{-6} & 2^{-7} & 2^{-8} & 2^{-9} & 2^{-10}
\end{array} \quad (5.25)$$

donde $d_i \in \{0,1\}$ $i = 1,2,\dots,10$

La evaluación es acompañada por la decodificación del cromosoma codificado y se calculan el peso del cromosoma usando el parámetro decodificado. La decodificación de λ^{nm} puede ser expresada como:

$$\lambda^{nm} = \sum_{i=1}^{10} (d_i \times 2^{-i}) \quad (5.26)$$

La relación entre el costo incremental actual, λ^{act} , y el costo incremental normalizado, λ^{nm} , es:

$$\lambda^{act} = \underline{\lambda}_{sys} + \lambda^{nm} (\overline{\lambda}_{sys} - \underline{\lambda}_{sys}) \quad (5.27)$$

donde $\overline{\lambda}_{sys}$ y $\underline{\lambda}_{sys}$ son los valores máximo y mínimo del costo incremental.

De la ecuación (5.21), partiendo del costo incremental, el problema de despacho económico se puede formular como:

$$Pf_i(2a_i P_i + b_i) = \lambda^{act} \quad (5.28)$$

$$\text{si } P_i \geq \text{Max}(\underline{P}_i, P_i^0 - DR_i) \Rightarrow P_i = \text{Max}(\underline{P}_i, P_i^0 - DR_i)$$

$$\text{si } P_i \leq \text{Min}(\overline{P}_i, P_i^0 + UR_i) \Rightarrow P_i = \text{Min}(\overline{P}_i, P_i^0 + UR_i)$$

Como se discutió anteriormente, la operación económica se logra evitando que la unidad opere en la zona prohibida. Cuando el resultado de (5.28) caiga en la zona prohibida, se procede de la siguiente manera: Si hay un incremento con respecto al período anterior P_i se iguala a P_{pz}^+ ; pero si existe un decremento con respecto al período anterior se iguala a P_{pz}^- .

Como la aproximación propuesta usa el criterio del costo incremental como base, la ecuación de restricción puede ser escrita como:

$$\varepsilon = \left| \sum_{i=1}^n P_i - P_D - P_{loss} \right| \quad (5.29)$$

Para acelerar la convergencia el peso es normalizado en el rango de 0 a 1. La función peso adoptado es:

$$fitness = \frac{1}{1 + k_T \left(\frac{\varepsilon}{P_D} \right)} \quad (5.30)$$

Para ambos procesos de optimización, se utiliza el mismo esquema que se muestra en la figura 5.8, la diferencia está en la decodificación y en el cálculo del fitness.

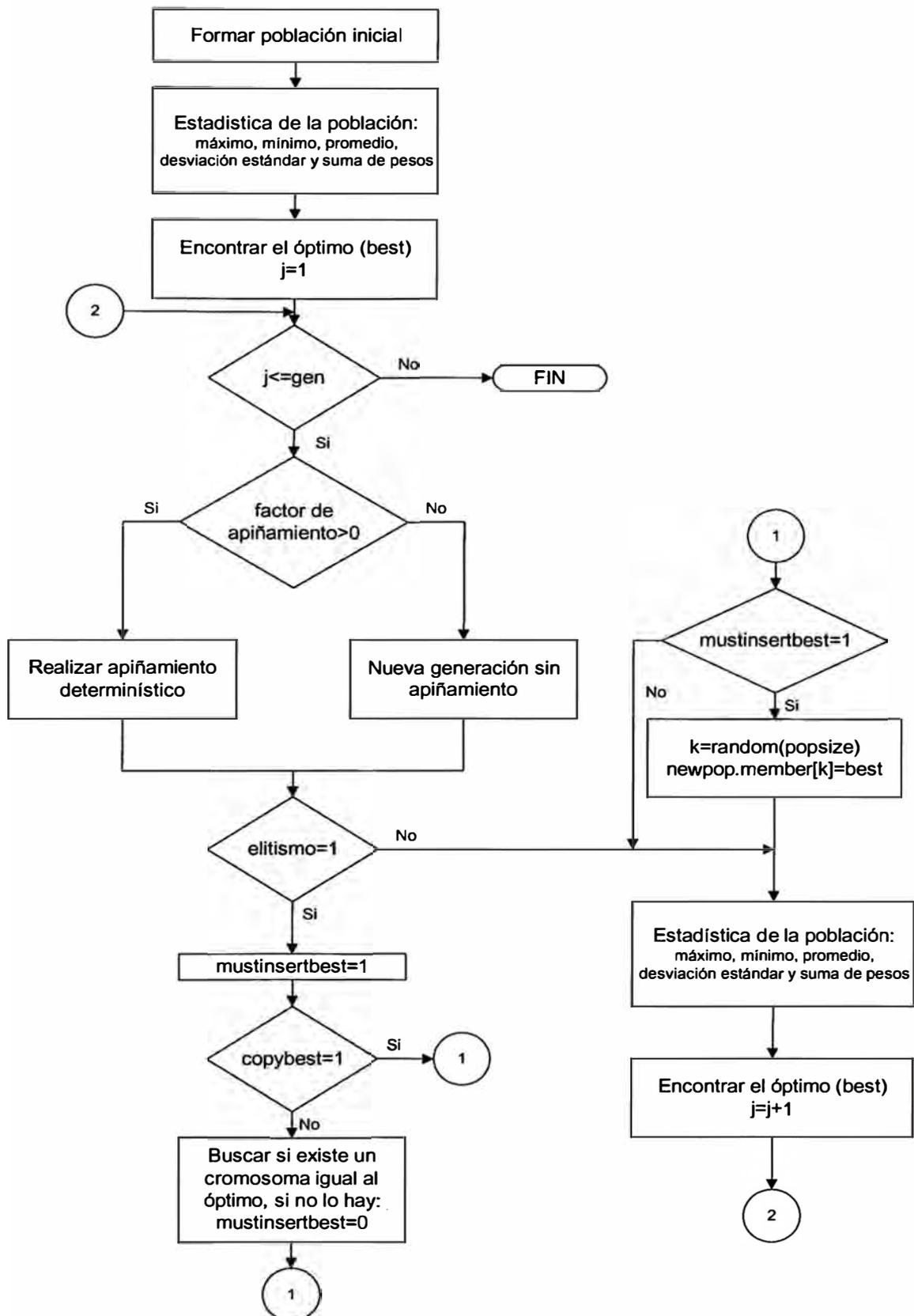
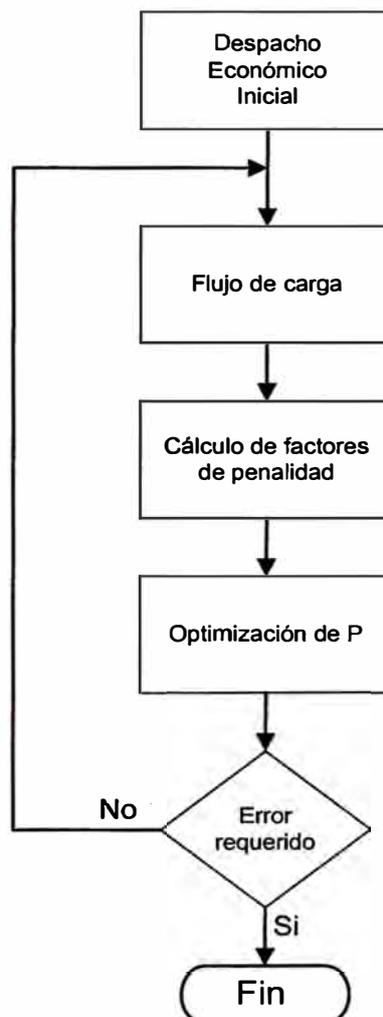


Figura 5.8 Diagrama de flujo de optimización de P

CAPÍTULO VI METODOLOGÍA DE CÁLCULO

6.1 Diagrama de flujo principal

Lo que se ha visto en los capítulos anteriores se utilizará para realizar el programa completo, así se tiene el siguiente diagrama de flujo:



6.2 Concepción de la metodología

En el diagrama de flujo principal se ha resumido todo el proceso para realizar el flujo de potencia óptimo, a continuación se explicará paso a paso el diagrama anterior:

- a) Al inicio se realiza un despacho económico inicial, para esto lo primero que se hace es igualar los factores de penalidad a 1 y la potencia de pérdidas a cero, luego se procede a optimizar la potencia de las centrales hidráulicas, una por una, se tiene la opción de definir centrales con potencias predefinidas también, así con las potencias hidráulicas y definidas se procede a realizar un despacho térmico, considerando constante las potencias mencionadas.
- b) Luego de tener las potencias de los generadores se procede a realizar un flujo de potencia, aquí se tiene ciertas constantes que el programa maneja internamente, estas son: el error en la solución del sistema lineal que es $1e-08$, el número de iteraciones máximo que es 20 y finalmente el error del flujo de potencia que es de 0.001pu. Aquí el programa arroja las tensiones a un archivo.
- c) Con el jacobiano del flujo se procede a calcular los factores de penalidad que se utilizará para realizar el nuevo despacho.
- d) En el bloque de optimización únicamente se procede a realizar un despacho térmico, ya que ya se hizo anteriormente el despacho hidráulico y al ser un flujo de potencia instantáneo, ya tenemos volúmenes definidos para las centrales hidráulicas.
- e) Luego se procede a ver el error requerido, este error se calcula como la diferencia en la potencia en la barra slack calculada mediante el flujo de potencia en la iteración k y la potencia en la barra slack calculada en la iteración $k-1$.

- f) Si el error que ingresamos para todo el proceso se satisface, el programa se termina. Si no se satisface, se procede a realizar nuevamente el flujo de potencia utilizando las nuevas potencias calculadas en el despacho térmico y así continúa el ciclo hasta que se satisface el error.

CAPÍTULO VII RESULTADOS NUMERICOS

Para realizar las aplicaciones se han utilizado dos sistemas de potencia: el sistema IEEE de 9 barras y el sistema IEEE de 30 barras, los cuales se describen a continuación.

7.1 Sistemas de potencia utilizados y datos de centrales usadas

7.1.1 IEEE de 9 barras

El diagrama unifilar del sistema de 9 barras es el mostrado en la figura 7.1.

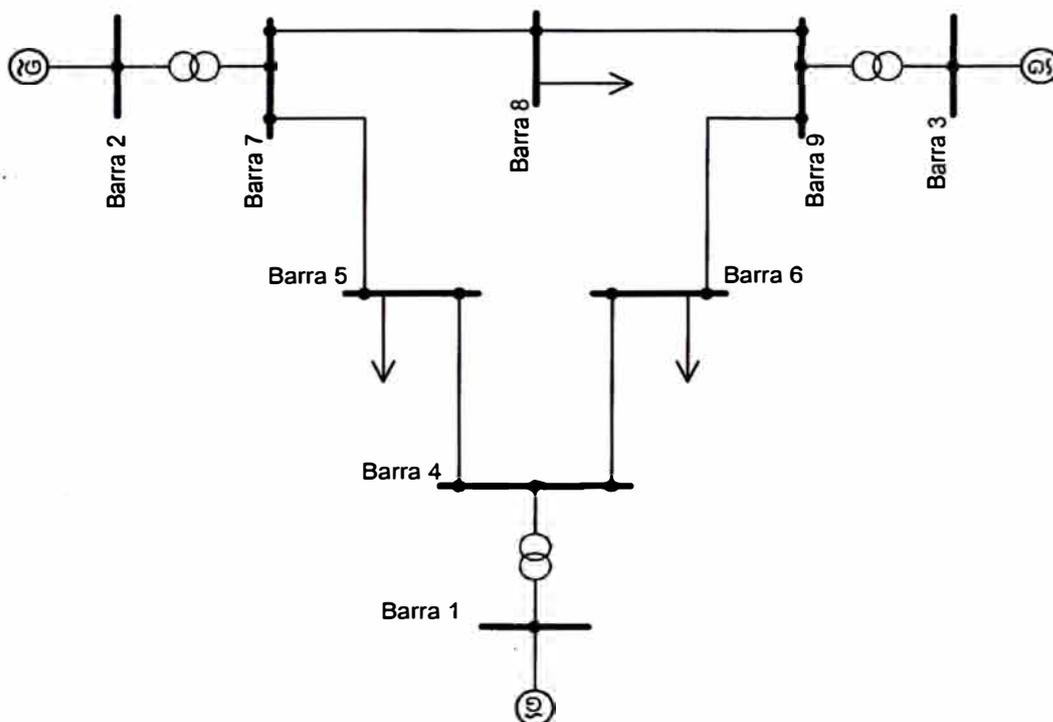


Fig 7.1 Diagrama Unifilar del Sistema IEEE de 9 barras

Los datos del sistema de 9 barras son:

Número	Nombre	Tensión(kV)
1	Barra 1	16,5
2	Barra 2	18,0
3	Barra 3	13,8
4	Barra 4	230,0
5	Barra 5	230,0
6	Barra 6	230,0
7	Barra 7	230,0
8	Barra 8	230,0
9	Barra 9	230,0

Listado de Barras

Número	Nombre	Nombre de Barra	Potencia Activa(MW)	Potencia Reactiva(MVar)
1	Carga 1	Barra 5	125	50
2	Carga 2	Barra 6	90	30
3	Carga 3	Barra 8	100	35

Listado de Cargas

Número	Nombre	Nombre de Barra	Potencia(MVA)	Tension(kV)	Potencia(MW)	Potencia(MVar)	Tipo
1	Gen 1	Barra 1	200	17,16	100,00	0,00	0
2	Gen 2	Barra 2	200	18,45	163,00	30,00	2
3	Gen 3	Barra 3	200	14,15	85,00	0,00	2

Listado de Generadores

Número	Nombre	BPrim	BSec	Tension(kV)	Longitud(km)	R(ohm/km)	X(ohm/km)	G(uS/km)	B(nF/km)
1	Linea 1	Barra 4	Barra 5	230	1	5,290	44,965	0	882,51
2	Linea 2	Barra 5	Barra 7	230	1	16,928	85,169	0	1535,5
3	Linea 3	Barra 7	Barra 8	230	1	4,497	3,809	0	747,13
4	Linea 4	Barra 8	Barra 9	230	1	6,295	53,323	0	1047,98
5	Linea 5	Barra 9	Barra 6	230	1	20,631	89,930	0	1795,13
6	Linea 6	Barra 6	Barra 4	230	1	8,993	48,668	0	792,3

Listado de Líneas

Número	Nombre	BPrim	BSec	Potencia(MVA)	Tensión de Cortocircuito(%)	Pérdidas en el cobre(%)	Pérdidas en el hierro(%)	Corriente Magnetizante(%)	VPrim	VSec
1	Trafo 1	Barra 1	Barra 4	250	14,40	0	0	0	230	16,5
2	Trafo 2	Barra 2	Barra 7	200	12,50	0	0	0	230	18
3	Trafo 3	Barra 3	Barra 9	150	8,79	0	0	0	230	13,8

Listado de Transformadores de 2 devanados

7.1.2 IEEE de 30 barras

El diagrama unifilar del sistema de 30 barras se muestra en la figura 7.2.

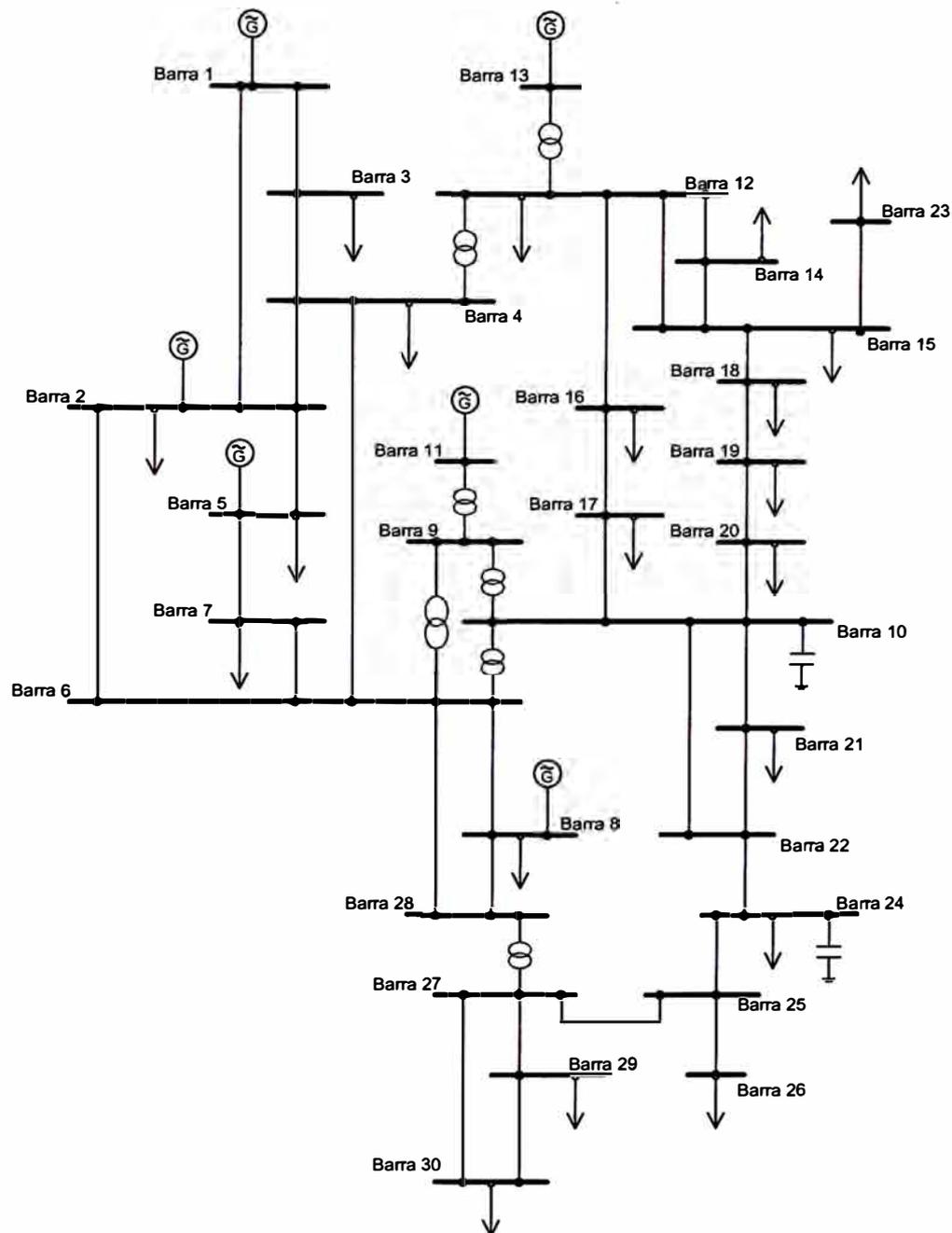


Fig 7.2 Diagrama Unifilar del Sistema IEEE de 9 barras

Los datos del sistema de 30 barras son:

Número	Nombre	Tensión(kV)
1	BARRA_1	60,0
2	BARRA_2	60,0
3	BARRA_3	60,0

4	BARRA_4	60,0
5	BARRA_5	60,0
6	BARRA_6	60,0
7	BARRA_7	60,0
8	BARRA_8	60,0
9	BARRA_9	13,8
10	BARRA_10	138,0
11	BARRA_11	5,0
12	BARRA_12	138,0
13	BARRA_13	13,8
14	BARRA_14	138,0
15	BARRA_15	138,0
16	BARRA_16	138,0
17	BARRA_17	138,0
18	BARRA_18	138,0
19	BARRA_19	138,0
20	BARRA_20	138,0
21	BARRA_21	138,0
22	BARRA_22	138,0
23	BARRA_23	138,0
24	BARRA_24	138,0
25	BARRA_25	138,0
26	BARRA_26	138,0
27	BARRA_27	138,0
28	BARRA_28	60,0
29	BARRA_29	138,0
30	BARRA_30	138,0

Listado de Barras

Número	Nombre	Nombre de Barra	Potencia Activa(MW)	Potencia Reactiva(MVar)
1	CARGA_1	BARRA_2	21,7	12,7
2	CARGA_2	BARRA_3	2,4	1,2
3	CARGA_3	BARRA_4	7,6	1,6
4	CARGA_4	BARRA_5	94,2	19,0
5	CARGA_5	BARRA_7	22,8	10,9
6	CARGA_6	BARRA_8	30,0	30,0
7	CARGA_7	BARRA_10	5,8	2,0
8	CARGA_8	BARRA_12	11,2	7,5
9	CARGA_9	BARRA_14	6,2	1,6
10	CARGA_10	BARRA_15	8,2	2,5
11	CARGA_11	BARRA_16	3,5	1,8
12	CARGA_12	BARRA_17	9,0	5,8
13	CARGA_13	BARRA_18	3,2	0,9
14	CARGA_14	BARRA_19	9,5	3,4
15	CARGA_15	BARRA_20	2,2	0,7
16	CARGA_16	BARRA_21	17,5	11,2
17	CARGA_17	BARRA_23	3,2	1,6

18	CARGA_18	BARRA_24	8,7	6,7
19	CARGA_19	BARRA_26	3,5	2,3
20	CARGA_20	BARRA_29	2,4	0,9
21	CARGA_21	BARRA_30	10,6	1,9

Listado de Cargas

Número	Nombre	Nombre de Barra	Potencia(MVA)	Tension(kV)	Potencia(MW)	Potencia(MVar)	Tipo
1	Gen 1	BARRA_1	200	63,0	98,95	0	0
2	Gen 2	BARRA_2	200	62,7	80,0	30	2
3	Gen 3	BARRA_5	200	60,6	50,0	0	2
4	Gen 4	BARRA_8	200	60,6	20,0	10	2
5	Gen 5	BARRA_11	200	5,3	20,0	10	2
6	Gen 6	BARRA_13	200	14,5	20,0	10	2

Listado de Generadores

Número	Nombre	BPrim	BSec	Tensión (kV)	Longitud (km)	Corriente Max(kA)	R (ohm/km)	X (ohm/km)	G (uS/km)	B (nF/km)
1	Linea 1	BARRA_3	BARRA_4	60	1	0,962	0,475	1,364	0,000	309,469
2	Linea 2	BARRA_3	BARRA_1	60	1	0,962	1,627	6,667	0,000	1503,130
3	Linea 3	BARRA_1	BARRA_2	60	1	1,925	0,691	2,070	0,000	1945,230
4	Linea 4	BARRA_2	BARRA_4	60	1	0,962	2,052	6,253	0,000	1355,768
5	Linea 5	BARRA_5	BARRA_2	60	1	0,962	1,699	7,139	0,000	1539,980
6	Linea 6	BARRA_7	BARRA_5	60	1	0,962	1,656	4,176	0,000	751,567
7	Linea 7	BARRA_6	BARRA_7	60	1	0,962	0,961	2,952	0,000	626,306
8	Linea 8	BARRA_8	BARRA_6	60	1	0,962	0,432	1,512	0,000	331,574
9	Linea 9	BARRA_14	BARRA_12	138	1	0,418	23,443	48,734	0,000	0,000
10	Linea 10	BARRA_15	BARRA_14	138	1	0,418	42,087	38,031	0,000	0,000
11	Linea 11	BARRA_15	BARRA_12	138	1	0,418	12,607	24,833	0,000	0,000
12	Linea 12	BARRA_16	BARRA_12	138	1	0,418	17,997	37,840	0,000	0,000
13	Linea 13	BARRA_17	BARRA_16	138	1	0,418	15,692	36,793	0,000	0,000
14	Linea 14	BARRA_17	BARRA_10	138	1	0,418	6,170	16,092	0,000	0,000
15	Linea 15	BARRA_18	BARRA_15	138	1	0,418	20,377	41,611	0,000	0,000
16	Linea 16	BARRA_19	BARRA_18	138	1	0,418	12,169	24,605	0,000	0,000
17	Linea 17	BARRA_20	BARRA_19	138	1	0,418	6,475	12,950	0,000	0,000
18	Linea 18	BARRA_20	BARRA_10	138	1	0,418	17,825	39,802	0,000	0,000
19	Linea 19	BARRA_22	BARRA_10	138	1	0,418	13,845	28,547	0,000	0,000
20	Linea 20	BARRA_21	BARRA_10	138	1	0,418	6,627	14,264	0,000	0,000
21	Linea 21	BARRA_21	BARRA_22	138	1	0,418	3,161	4,494	0,000	0,000
22	Linea 22	BARRA_15	BARRA_23	138	1	0,418	19,044	38,469	0,000	0,000
23	Linea 23	BARRA_26	BARRA_25	138	1	0,418	48,448	72,367	0,000	0,000
24	Linea 24	BARRA_25	BARRA_24	138	1	0,418	35,898	62,693	0,000	0,000
25	Linea 25	BARRA_24	BARRA_22	138	1	0,418	21,901	34,089	0,000	0,000
26	Linea 26	BARRA_6	BARRA_28	60	1	0,962	0,608	2,156	0,000	478,940
27	Linea 27	BARRA_28	BARRA_8	60	1	0,962	2,290	7,200	0,000	1576,818
28	Linea 28	BARRA_27	BARRA_25	138	1	0,418	20,815	39,631	0,000	0,000
29	Linea 29	BARRA_30	BARRA_27	138	1	0,418	60,979	114,778	0,000	0,000
30	Linea 30	BARRA_29	BARRA_27	138	1	0,418	41,859	79,090	0,000	0,000
31	Linea 31	BARRA_30	BARRA_29	138	1	0,418	45,687	86,326	0,000	0,000

32	Línea 32	BARRA_2	BARRA_6	60	1	0,962	2,092	6,347	0,000	1377,870
33	Línea 33	BARRA_4	BARRA_6	60	1	0,962	0,428	1,490	0,000	331,574

Listado de Líneas

Número	Nombre	Nombre de Barra	Tension(kV)	Potencia(MVar)	Tension Nominal(kV)
1	Cap1	BARRA_10	138	5,26	138
2	Cap2	BARRA_24	138	25,00	138

Listado de Shunts

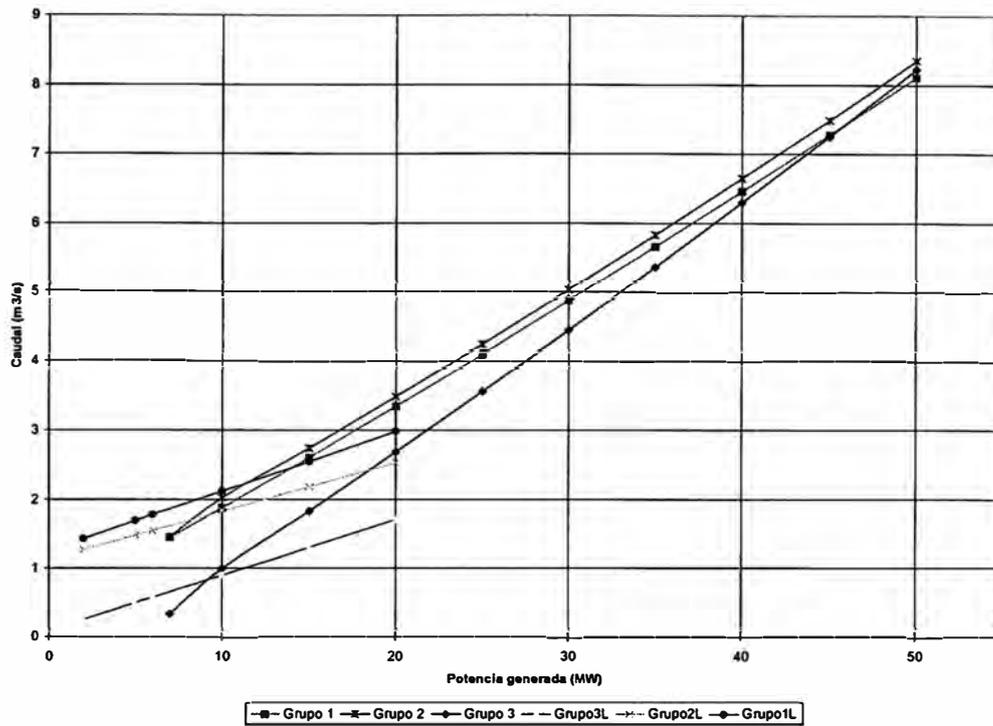
Numero	Nombre	BPrim	BSec	Potencia (MVA)	Tension de Cortocircuito(%)	Pérdidas en el cobre(%)	Pérdidas en el hierro(%)	Corriente Magnetizante(%)	VPrim	VSec
1	Trafo 1	BARRA_6	BARRA_9	50	10,4	0	0	0	60	13,8
2	Trafo 2	BARRA_10	BARRA_6	50	27,8	0	0	0	138	60
3	Trafo 3	BARRA_10	BARRA_9	100	5,5	0	0	0	138	13,8
4	Trafo 4	BARRA_9	BARRA_11	50	10,4	0	0	0	13,8	5
5	Trafo 5	BARRA_12	BARRA_4	65	16,64	0	0	0	138	60
6	Trafo 6	BARRA_12	BARRA_13	65	9,1	0	0	0	138	13,8
7	Trafo 7	BARRA_27	BARRA_28	65	25,74	0	0	0	138	60

Listado de Transformadores de 2 devanados

7.1.3 Datos de centrales

Para probar la optimización hidráulica se han utilizado los datos de la central Charcani V, que se compone de tres grupos y se ha supuesto que se dispone de un caudal de 18 m³/s. En la figura 7.3 se presentan las características Potencia-Caudal correspondientes.

Fig. 7.3 Gráficas Caudal vs Potencia



Las ecuaciones que describen las curvas de la Fig. 7.3 son:

Grupo 1:

$$\begin{cases} q_w = 0.086P_w + 1.26 & 0 < P_w < 14.145 \\ q_w = 0.0003P_w^2 + 0.138P_w + 0.4639 & 14.145 \leq P_w \leq 50.9 \end{cases}$$

Grupo 2:

$$\begin{cases} q_w = 0.07P_w + 1.13 & 0 < P_w < 7.318 \\ q_w = 0.0004P_w^2 + 0.1343P_w + 0.6381 & 7.318 \leq P_w \leq 49.19 \end{cases}$$

Grupo 3:

$$\begin{cases} q_w = 0.0802P_w + 0.0991 & 0 < P_w < 8.91 \\ q_w = 0.0004P_w^2 + 0.1568P_w - 0.6152 & 8.91 \leq P_w \leq 51.1 \end{cases}$$

Para probar la optimización térmica se ha utilizado los datos de tres centrales de la referencia [16], también se utilizan los datos de la referencia[15].

7.1.4 Secuencia de análisis

Primero se procede a realizar el flujo de potencia con las potencias de generadores programadas para corroborar el flujo. Luego se han implementado los siguientes casos:

- IEEE-9 con el Gen2 Hidráulico y el resto fijo
- IEEE-9 con los tres generadores de la referencia [16]
- IEEE-9 con un Gen3 Hidráulico, Gen1 grupo térmico No1 de [16] y Gen2 grupo térmico No1 de [15]
- IEEE-30 con los cinco generadores de la referencia[15]
- IEEE-30 donde se reemplaza el Gen2 del caso anterior por los datos Hidráulicos

7.2 Análisis de operadores

A continuación se analizará los operadores de los algoritmos genéticos, así como sus tipos para realizar la optimización térmica e hidráulica.

En el caso de la *optimización hidráulica*, antes de realizar la optimización se procede a encontrar un K_H de la ecuación (5.24), se han hecho pruebas con tres valores de K_H ; 10, 5 y 1; que se presentan en las figuras 7.4, 7.5 y 7.6.

Fig 7.4a Potencia vs Peso (Kh=1)

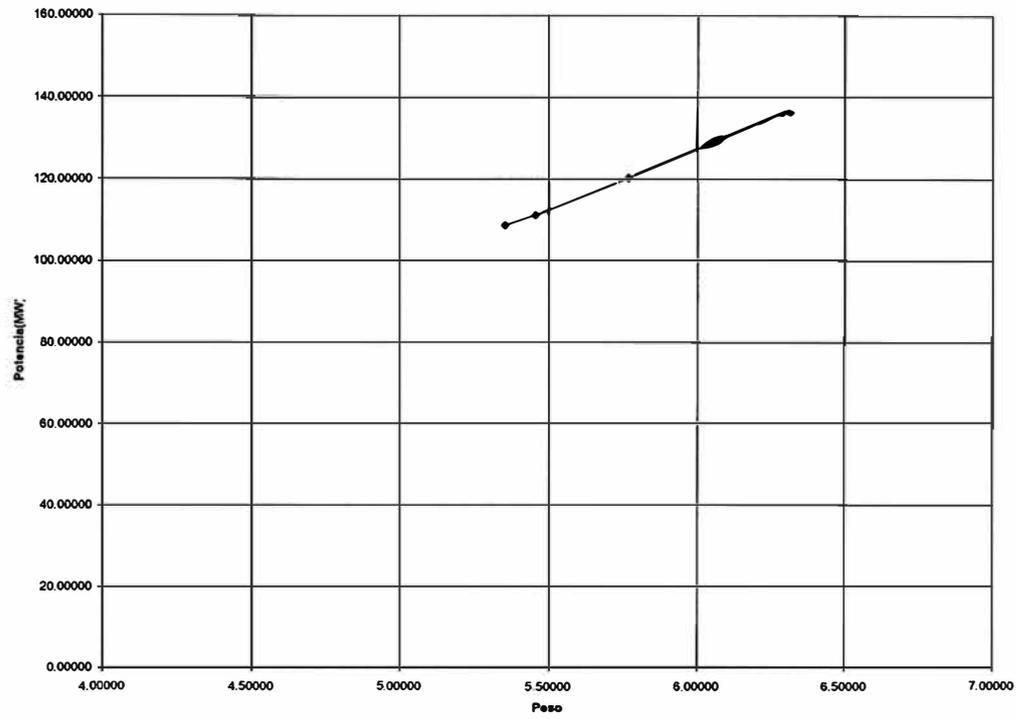


Fig7.4b Error vs Peso(Kh=1)

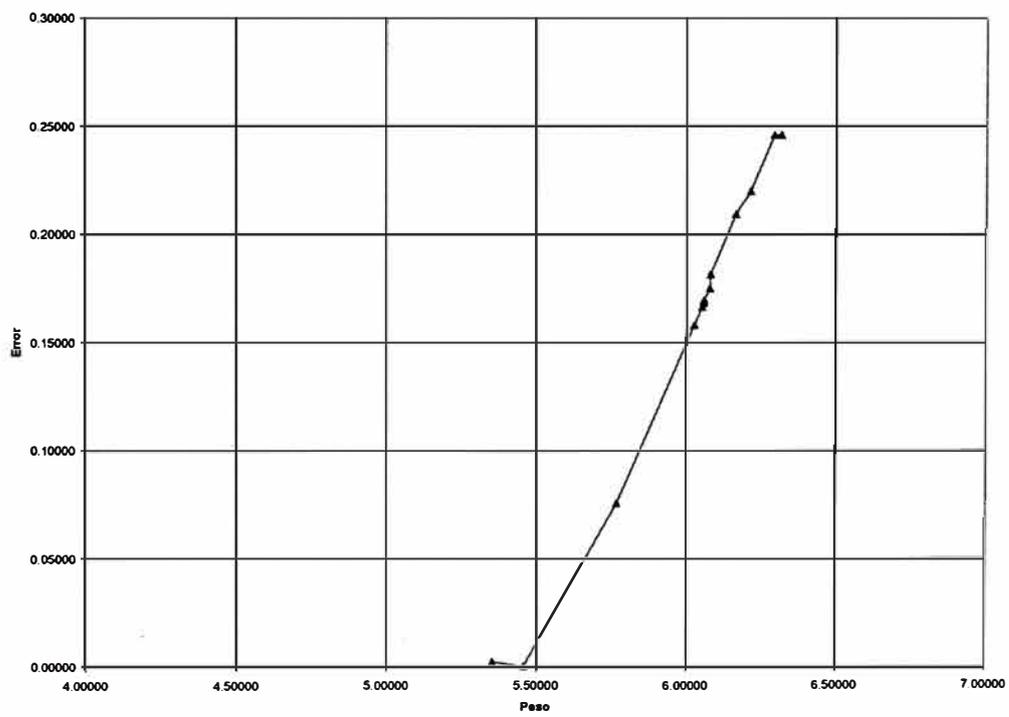


Fig7.5a Potencia vs Peso (Kh=5)

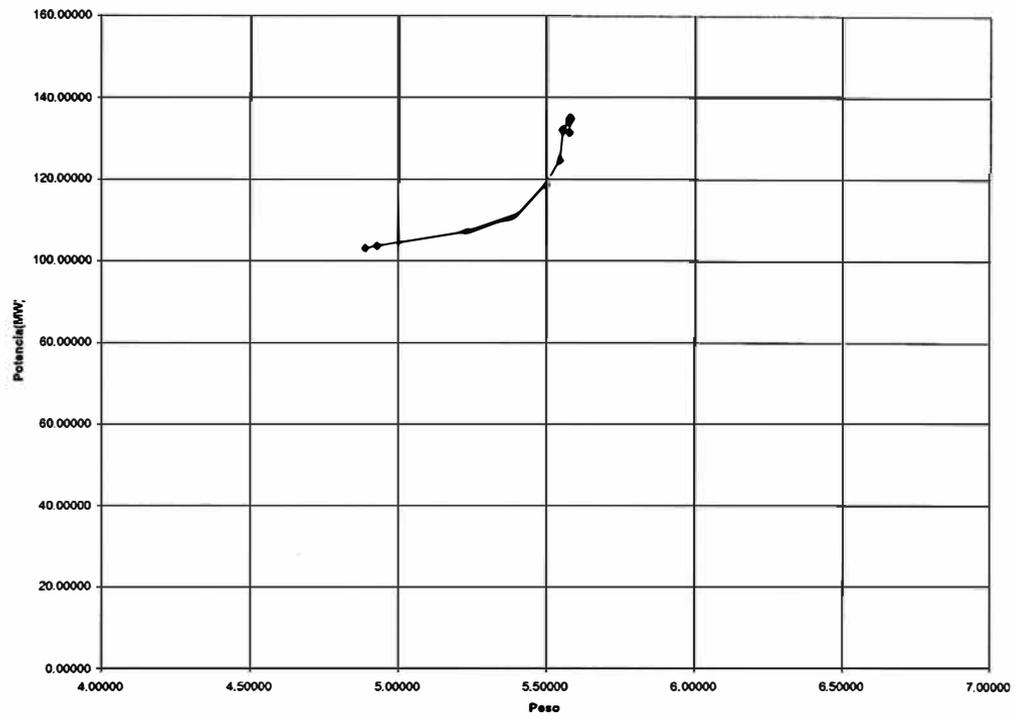


Fig7.5b Error vs Peso (Kh=5)

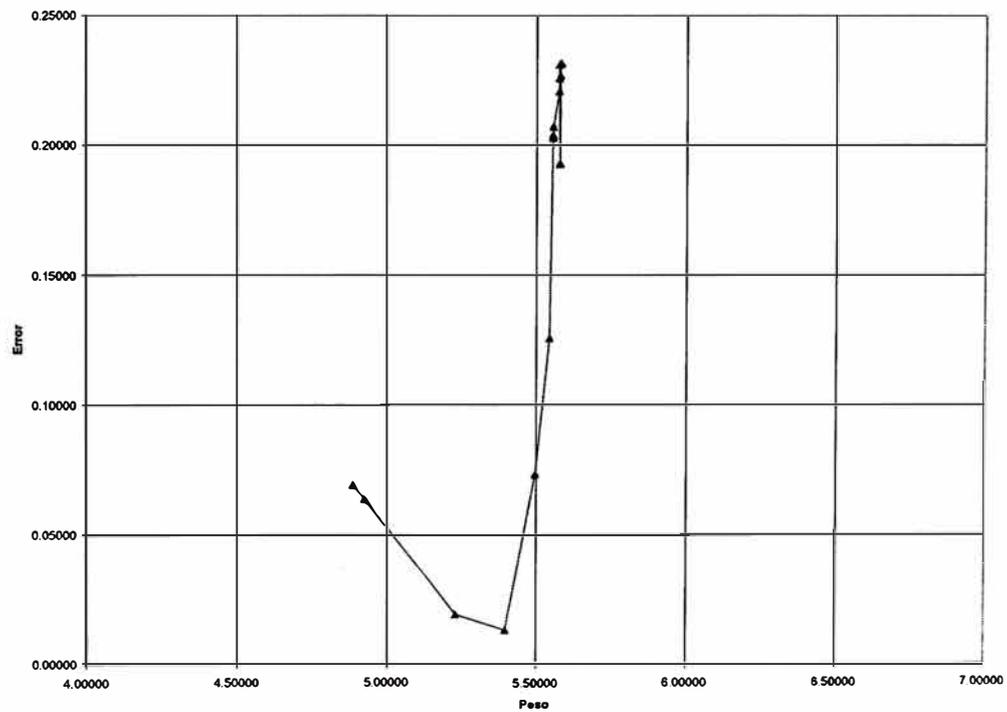


Fig7.6a Potencia vs Peso (Kh=10)

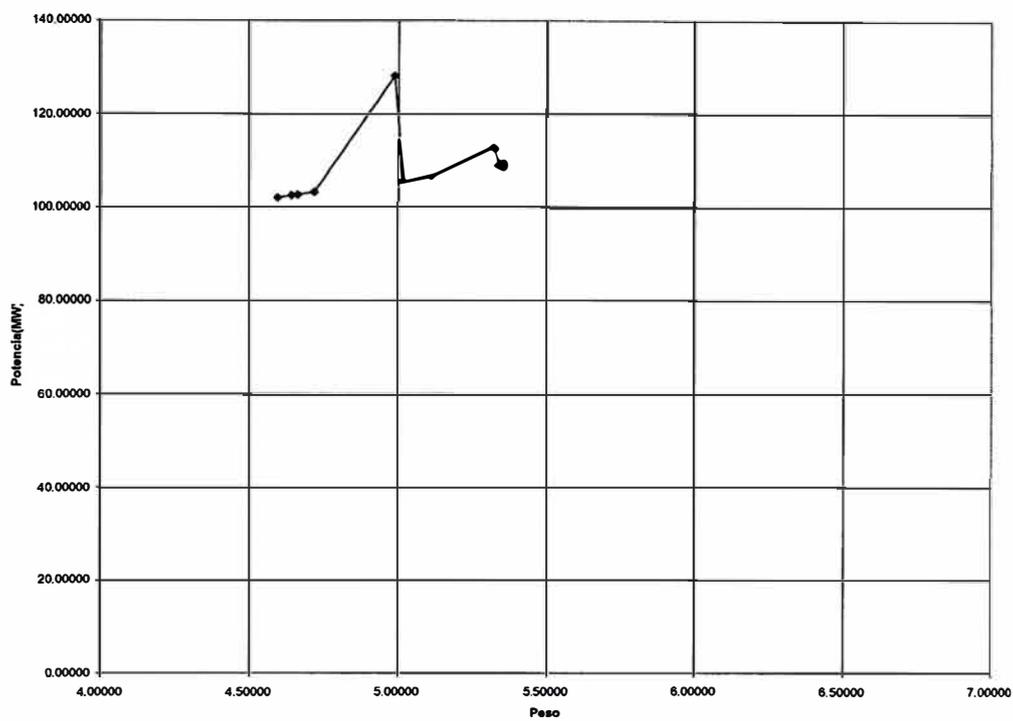
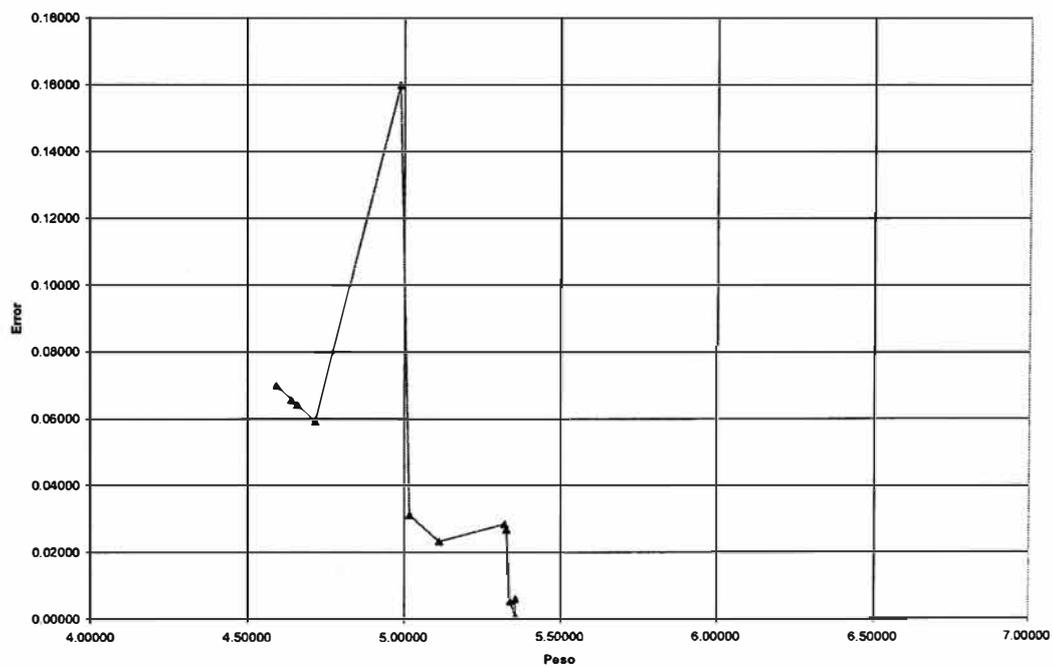


Fig7.6b Error vs Peso (Kh=10)



Se aprecia que para todos los valores de K_H la potencia aumenta de acuerdo al aumento del peso, esto es debido a las ecuaciones que se han utilizado para modelar el peso en las centrales hidráulicas. En el caso del error, para $K_H=1$ el valor del error aumenta conforme aumenta el peso; para $K_H=5$ el valor del error en un primer momento disminuye y luego aumenta; por otro lado en el caso de $K_H=10$ el valor del error tiene una tendencia a disminuir conforme aumenta el peso. A partir de estos resultados, se concluye que el mejor valor de K_H es 10.

Para el análisis de la optimización hidráulica se ha utilizado la secuencia de operadores que se muestra en la tabla 7.1.

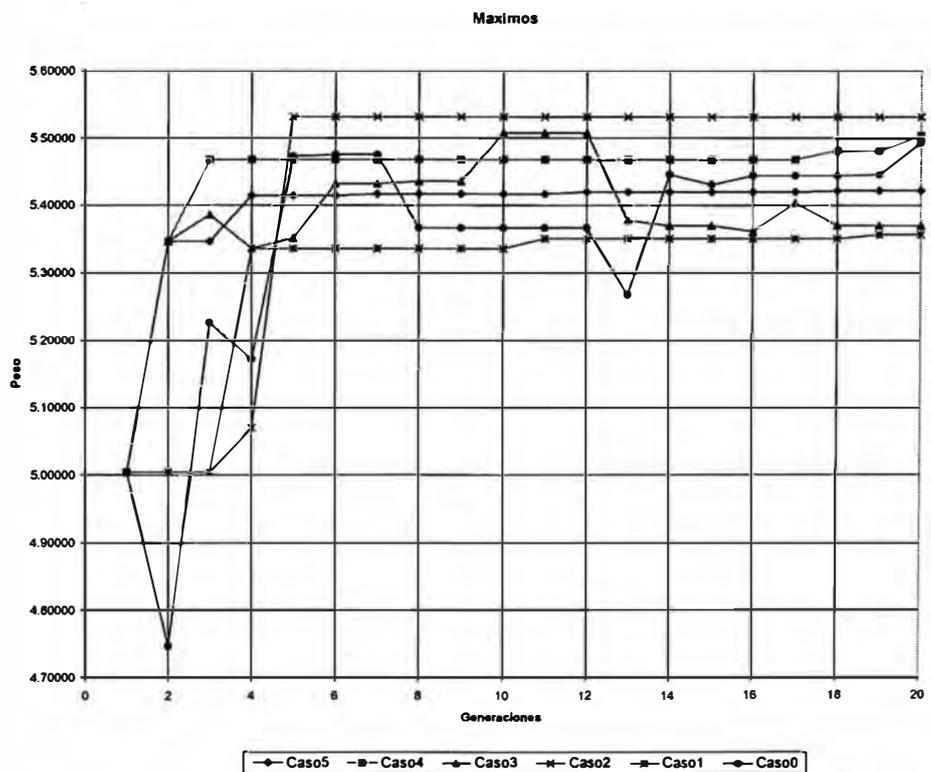
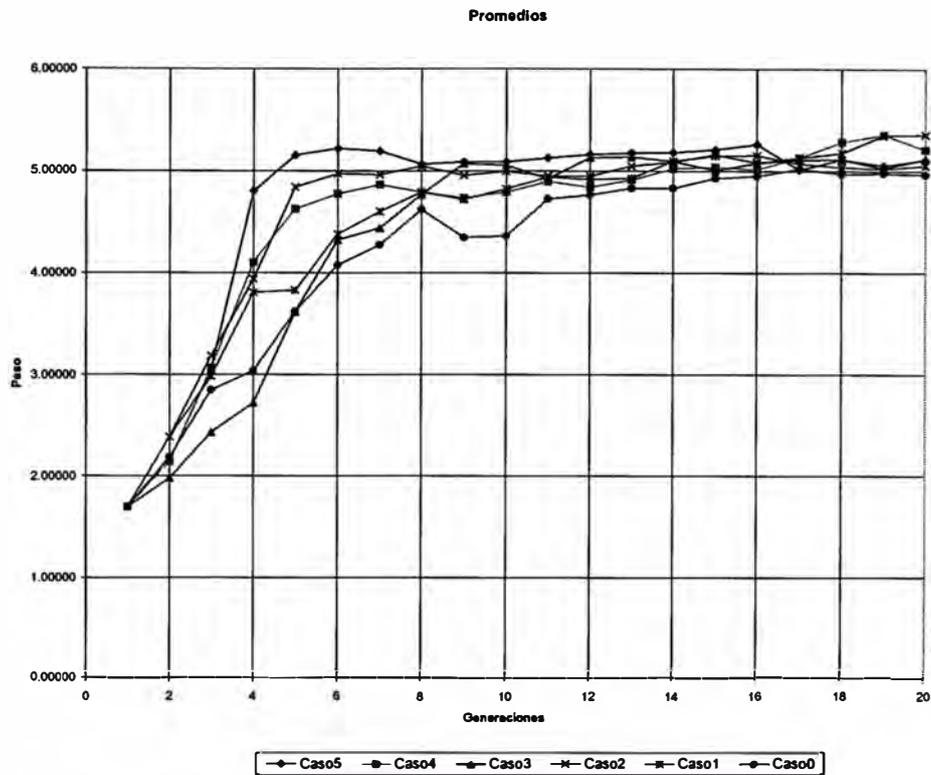
Se va a comentar por bloques de casos de acuerdo a los casos donde exista modificaciones similares, así podremos obtener conclusiones por la modificación de ciertos parámetros.

CUADRO 01

CASO	Longitud de cromosoma	The best	Preselección	Selección	Probabilidad de cruce	Numero de puntos	Tipo de Cruce	Probabilidad de mutación	Tipo de Mutación	Tipo de Reemplazo	Tipo de niching	Prevención de Incesto	Tipo de Ruido
0	30	Ninguno	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
1	30	Elitismo	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
2	30	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
3	30	Ninguno	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
4	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
5	30	Copiar el mejor	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
6	30	Copiar el mejor	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
7	30	Elitismo	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
8	30	Copiar el mejor	Ninguna	Ruleta	0,9	1	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
9	30	Elitismo	Escalamiento lineal	Ruleta	0,9	1	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
10	30	Copiar el mejor	Ninguna	Ruleta	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
11	30	Elitismo	Escalamiento lineal	Ruleta	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
12	30	Copiar el mejor	Ninguna	Ruleta	0,9	3	De n puntos	0,005	Simple inversión	Agregar	Ninguno	-----	Ninguno
13	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,005	Simple inversión	Agregar	Ninguno	-----	Ninguno
14	30	Copiar el mejor	Ninguna	Ruleta	0,9	3	De n puntos	0,02	Simple inversión	Agregar	Ninguno	-----	Ninguno
15	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,02	Simple inversión	Agregar	Ninguno	-----	Ninguno
16	30	Copiar el mejor	Ninguna	Ruleta	0,9	3	De n puntos	0,01	Simple inversión	El más débil	Ninguno	-----	Ninguno
17	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	El más débil	Ninguno	-----	Ninguno
18	30	Copiar el mejor	Ninguna	Ruleta	0,9	2	Mezcla con tabla	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
19	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	Mezcla con tabla	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
20	30	Copiar el mejor	Ninguna	Ruleta	0,9	-----	Uniforme	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
21	30	Elitismo	Escalamiento lineal	Ruleta	0,9	-----	Uniforme	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
22	30	Copiar el mejor	Ninguna	Ruleta	0,9	-----	Uniforme modificado	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
23	30	Elitismo	Escalamiento lineal	Ruleta	0,9	-----	Uniforme modificado	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
24	30	Copiar el mejor	Ninguna	Estocastica sobrante	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
25	30	Elitismo	Ninguna	Estocastica sobrante	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
26	30	Copiar el mejor	Ninguna	Estocastica universal	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
27	30	Elitismo	Ninguna	Estocastica universal	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
28	30	Copiar el mejor	Ninguna	Torneo	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
29	30	Elitismo	Ninguna	Torneo	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
30	30	Copiar el mejor	Ninguna	Estocastica sobrante	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
31	30	Elitismo	Ninguna	Estocastica sobrante	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
32	30	Copiar el mejor	Ninguna	Estocastica universal	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
33	30	Elitismo	Ninguna	Estocastica universal	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
34	30	Copiar el mejor	Ninguna	Torneo	0,9	3	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
35	30	Elitismo	Ninguna	Torneo	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
36	30	Copiar el mejor	-----	-----	0,9	-----	-----	0,01	Simple inversión	-----	Apiñamiento	NO	Ninguno
37	30	Elitismo	-----	-----	0,9	-----	-----	0,01	Simple inversión	-----	Apiñamiento	NO	Ninguno
38	30	Copiar el mejor	-----	-----	0,9	-----	-----	0,01	Simple inversión	-----	Apiñamiento	SI	Ninguno
39	30	Elitismo	-----	-----	0,9	-----	-----	0,01	Simple inversión	-----	Apiñamiento	SI	Ninguno
40	30	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
41	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
42	30	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme
43	30	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme
44	15	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
45	15	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
46	15	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
47	15	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
48	15	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme
49	15	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme

Tabla 7.1 Casos para optimización hidráulica

Análisis de elitismo: Caso 0, caso 1, caso 2, caso 3, caso 4 y caso 5

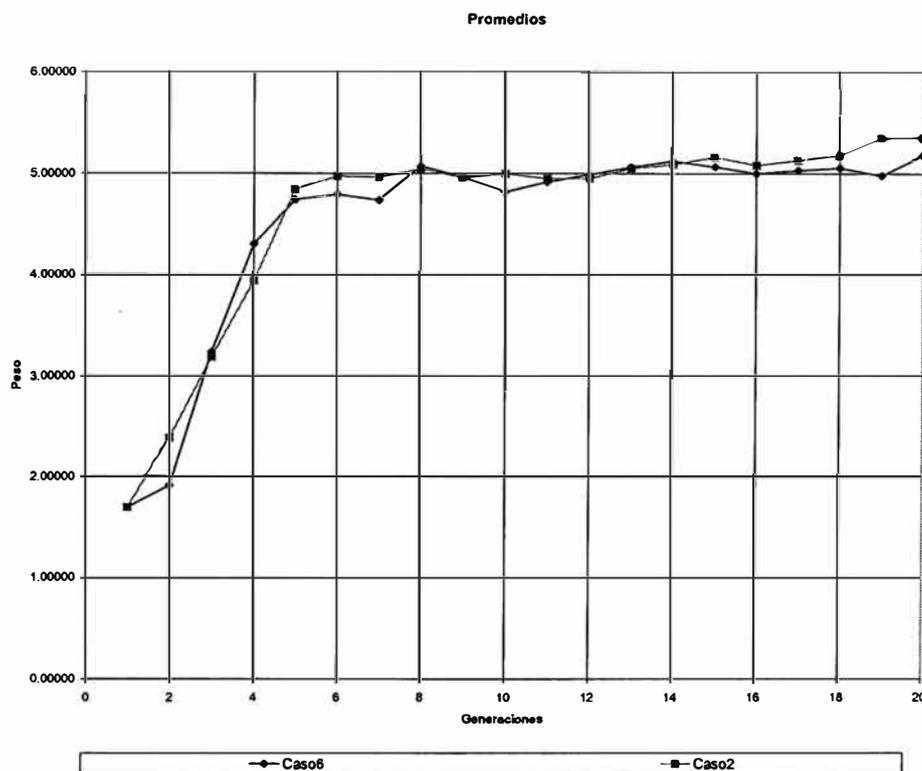


En el caso 0 y en el caso 3 como no existe elitismo se observa que el peso aumenta y disminuye y demora en converger. El que llega a converger más rápidamente es el caso 4,

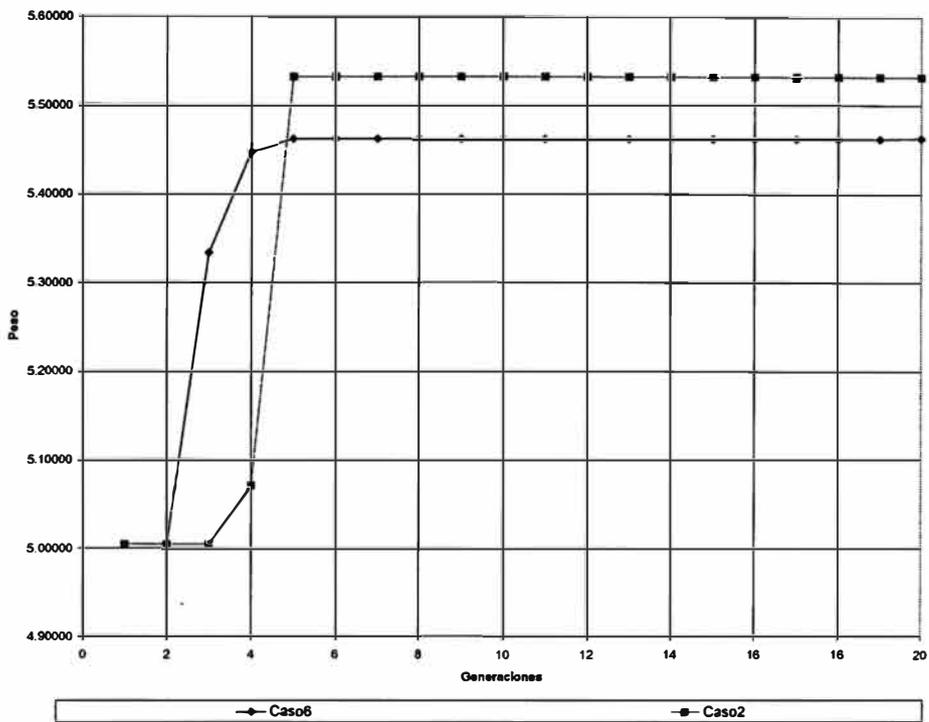
pero se mantiene constante en las generaciones siguientes. El caso 2 es el que alcanza el mayor peso con un valor de 5.53157.

Por otro lado se observa que el promedio en el caso 4 demora en converger, esto evita la convergencia prematura y a la larga produce mejores individuos, en el resto de casos se ve que el promedio aumenta rápidamente, esto produce una convergencia prematura y a la larga estanca el mejor en un valor. Por lo encontrado, se ha decidido probar en adelante con los casos 2 y 4 como base para la modificación de sus parámetros.

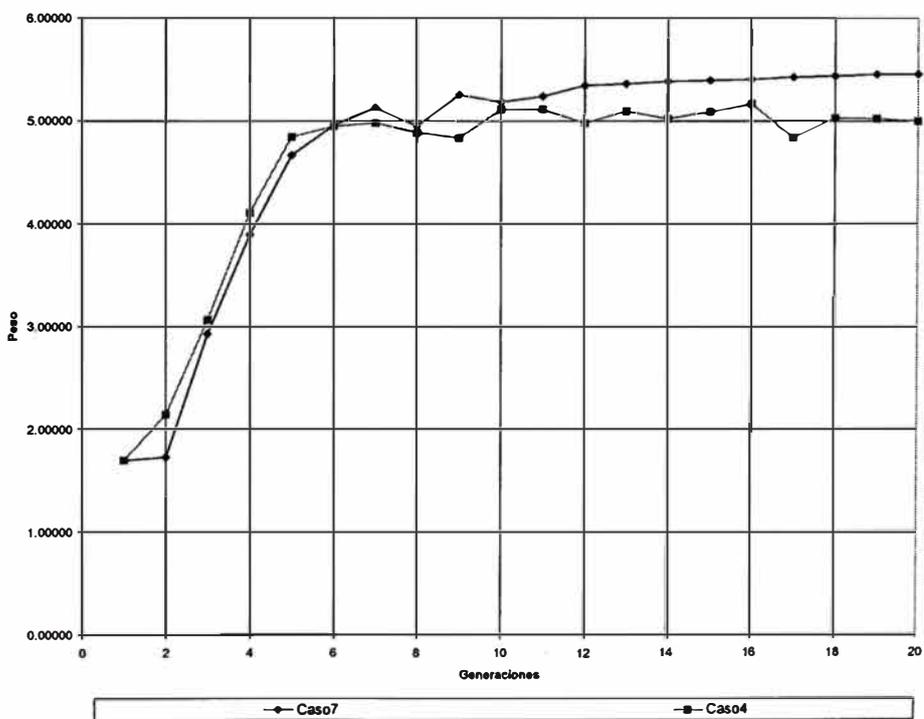
Análisis de probabilidad de cruce: Caso 2, caso 4, caso 6 y caso 7

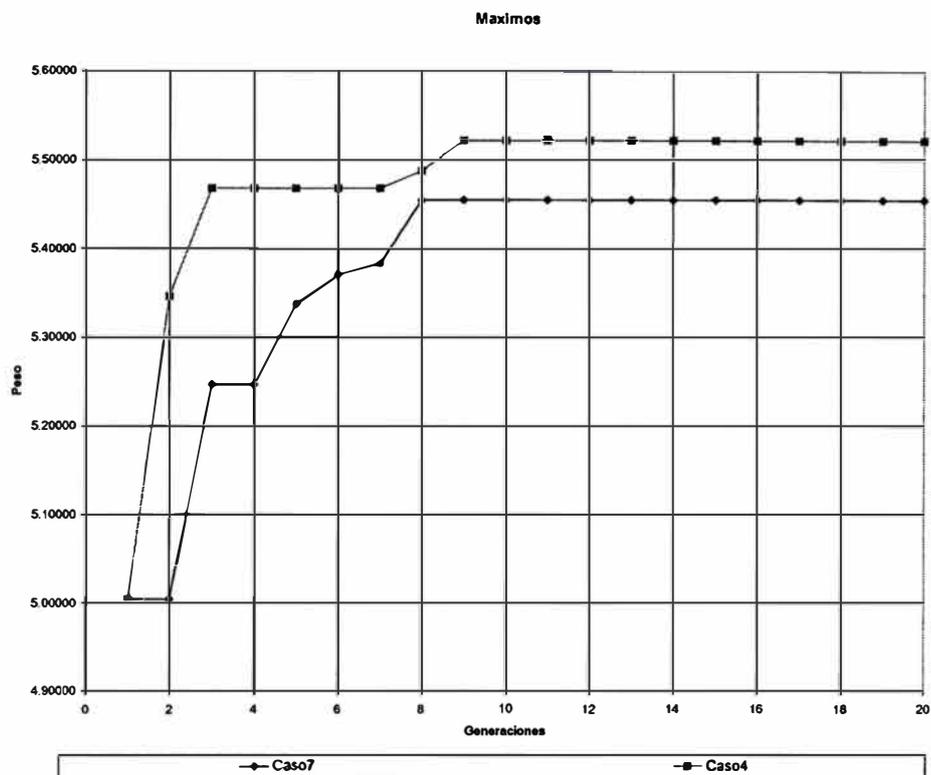


Maximos



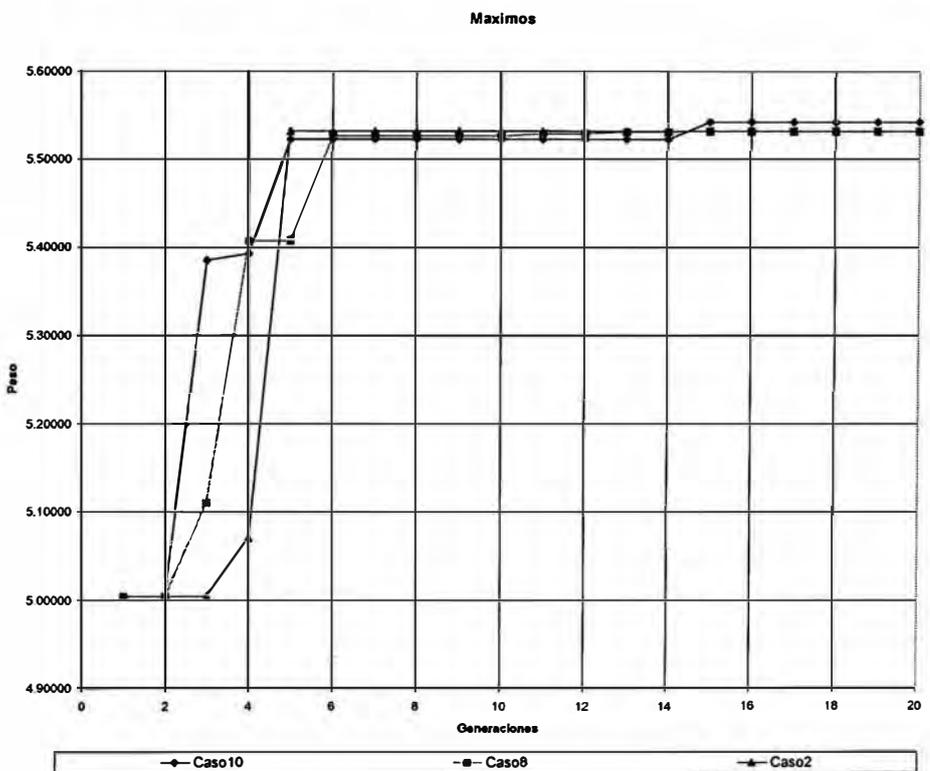
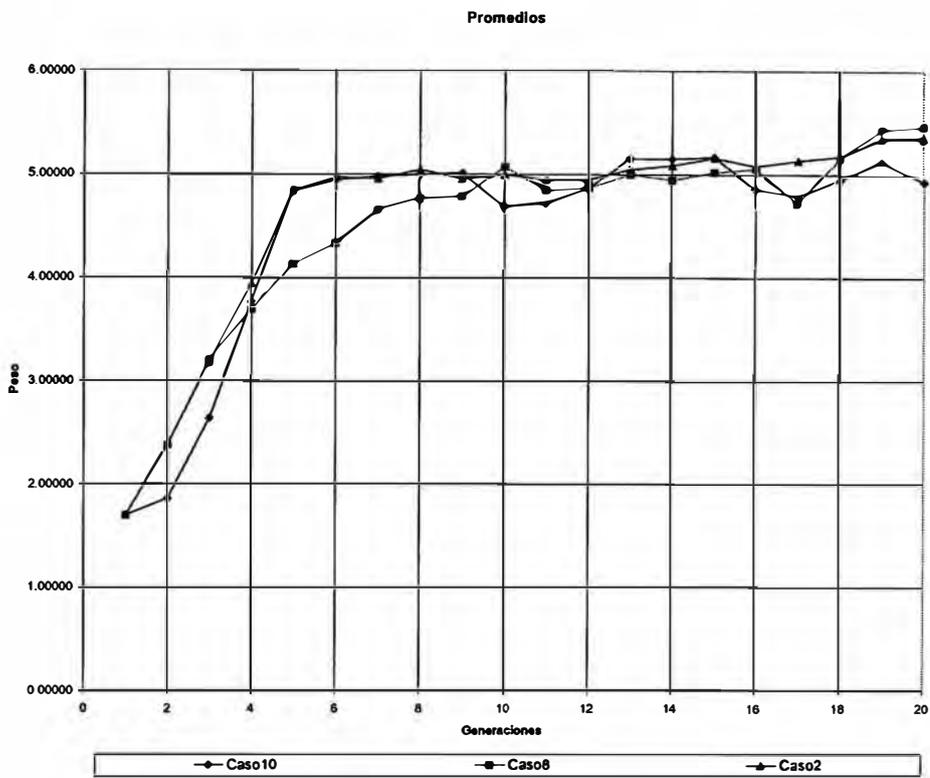
Promedios

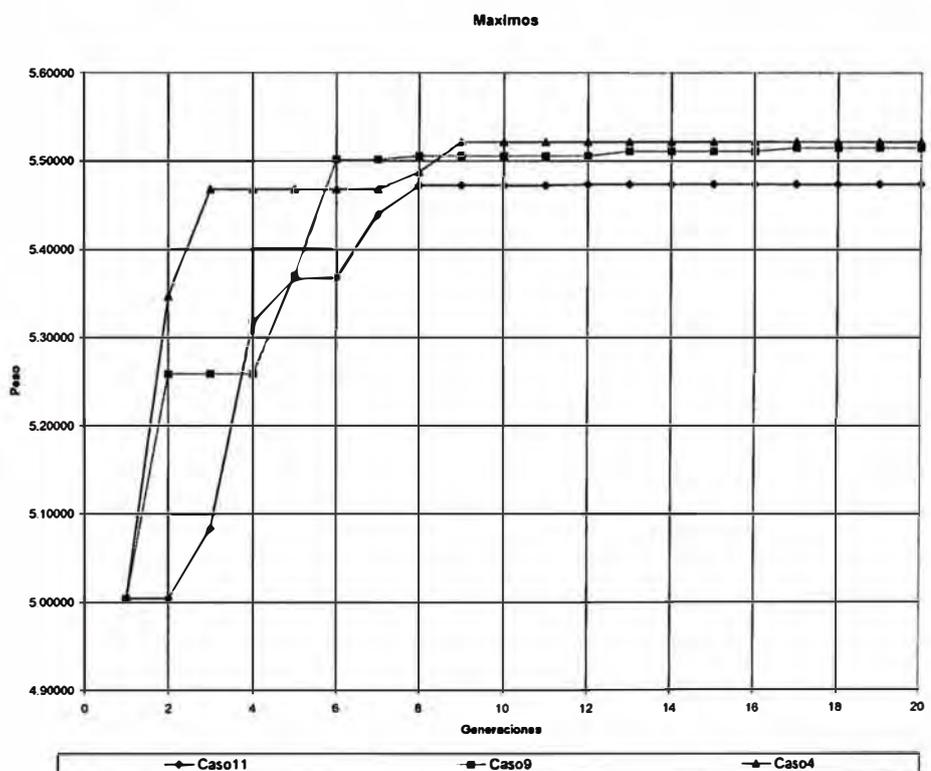
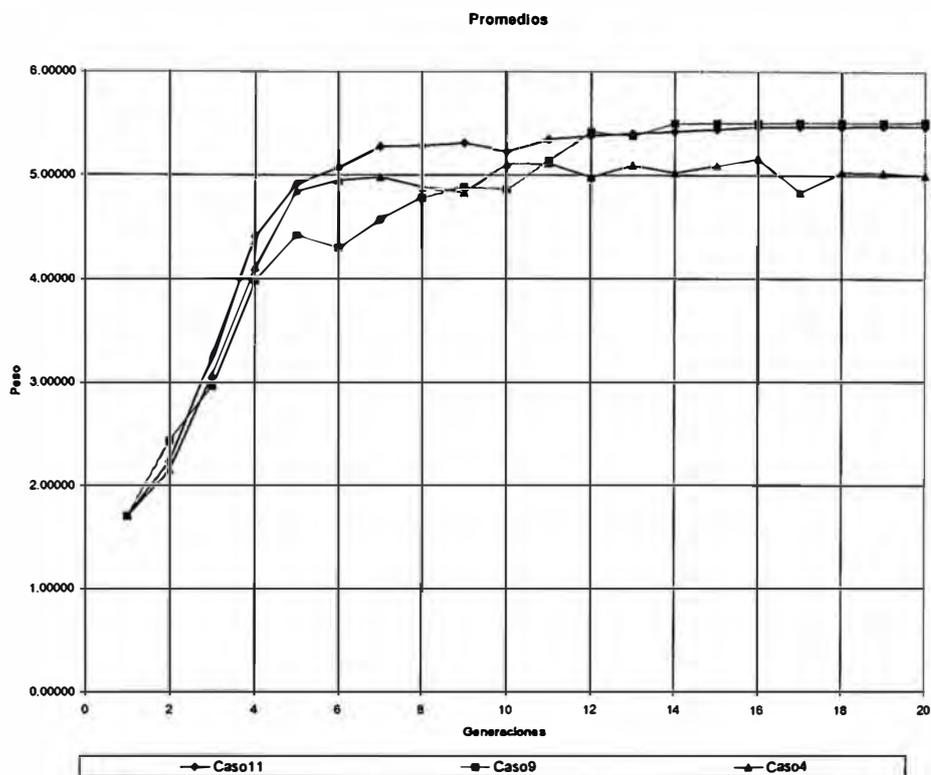




Se observa que tanto en el caso 6 y en el caso 7 (la probabilidad de cruce de 0.75) no se alcanza el óptimo que se alcanza en el caso 2 y en el caso 4 (la probabilidad de cruce de 0.9). Por otro lado los promedios son similares, aunque en el caso 7 se ve un mejor promedio lo que significa que será muy difícil conseguir un mejor individuo en ese caso, por lo tanto se tomará la probabilidad de cruce de 0.9, con esto los casos 2 y 4 continúan como base.

Análisis de los puntos de cruce: Caso 2, caso 4, caso 8, caso 9, caso 10 y caso 11

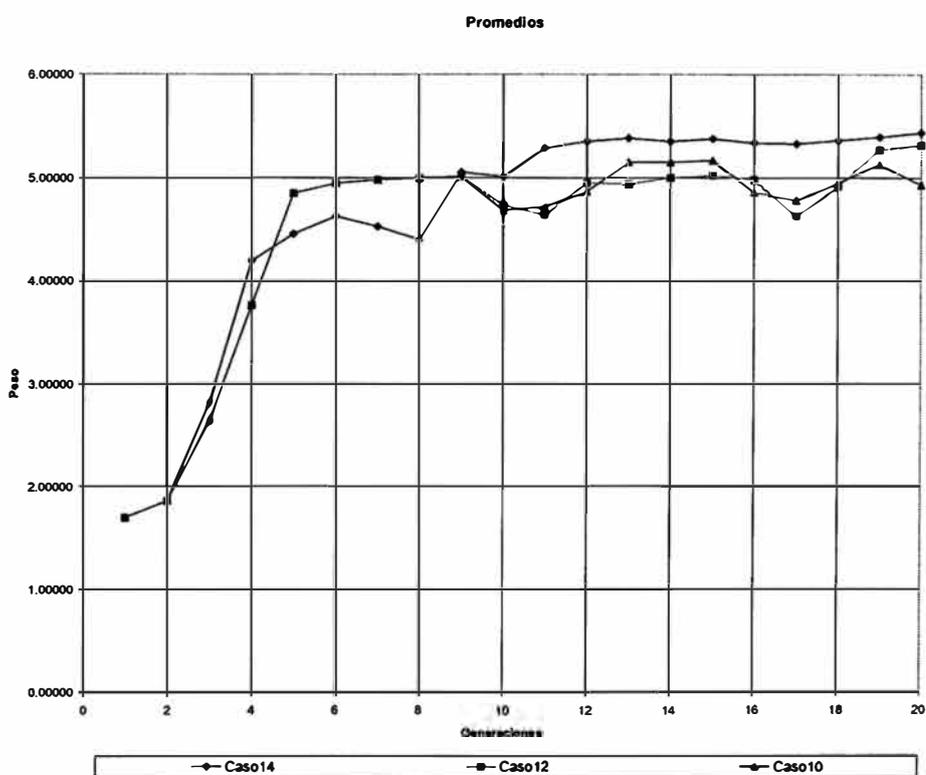




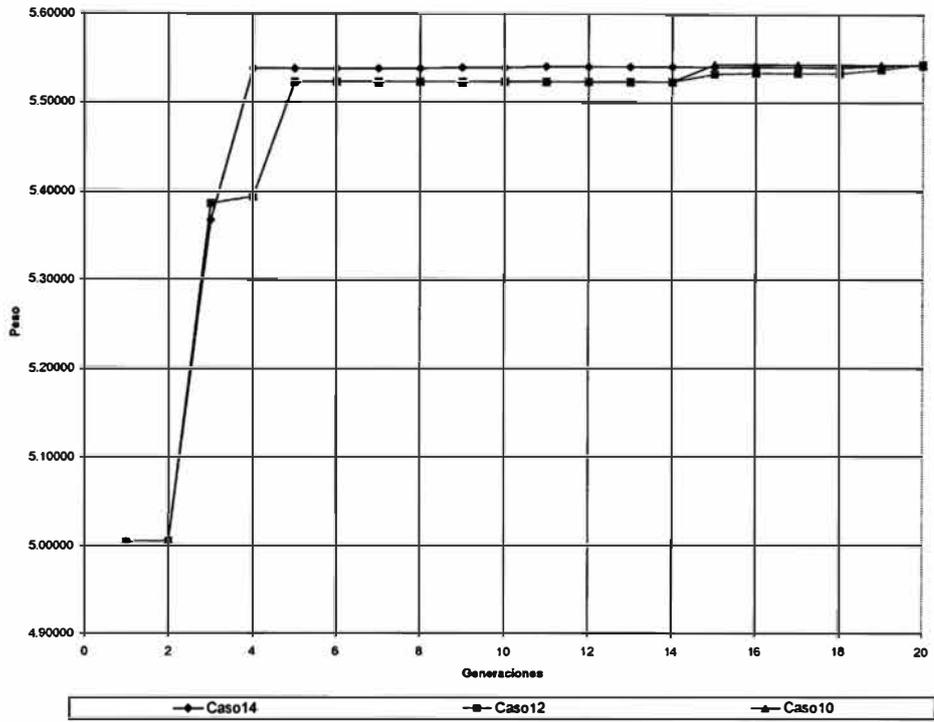
Quando se aplican diversos puntos de cruce al caso 2, se aprecia que cuando se tienen 3 puntos de cruce se consigue un mejor individuo y a su vez el promedio no es muy alto lo que

significa que se mantiene cierta diversidad, pudiendo conseguirse mejores resultados en futuras generaciones. Para el caso 4, el mejor número de puntos de cruce es 2, a diferencia del anterior caso con 3 puntos se obtiene un mal resultado. De lo anterior se deduce que los mejores casos con el caso 10 y el caso 4.

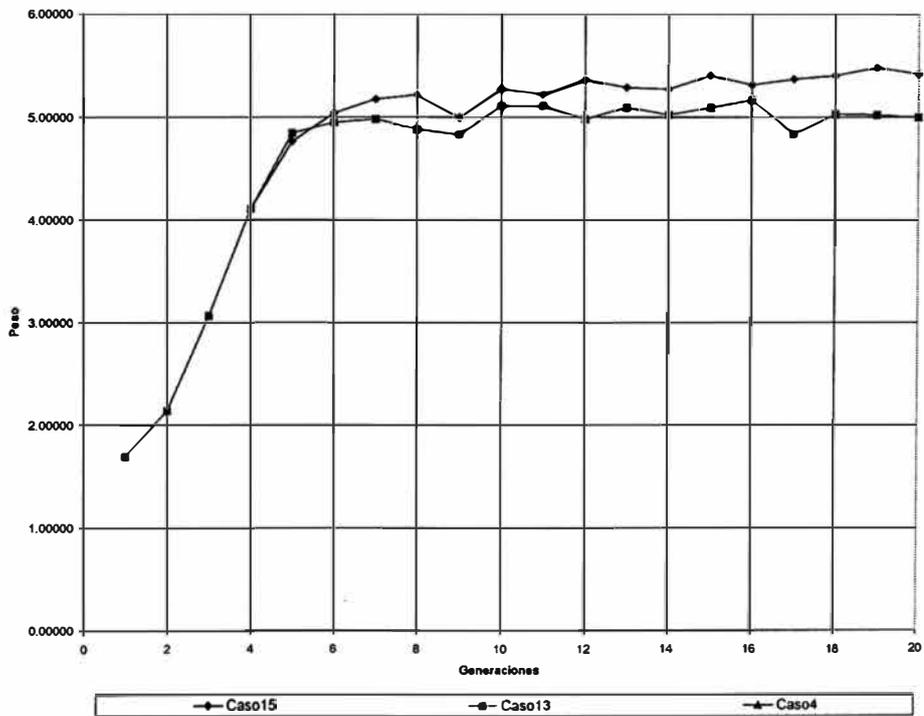
Análisis de la probabilidad de mutación: Caso 10, caso 4, caso 12, caso 13, caso 14 y caso 15

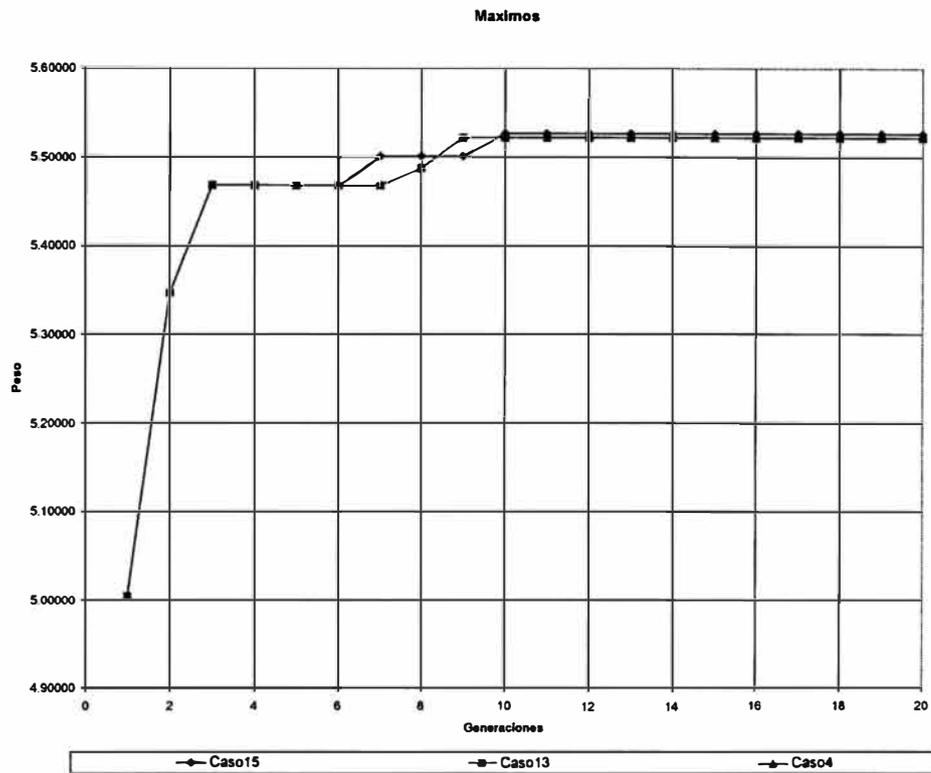


Maximos



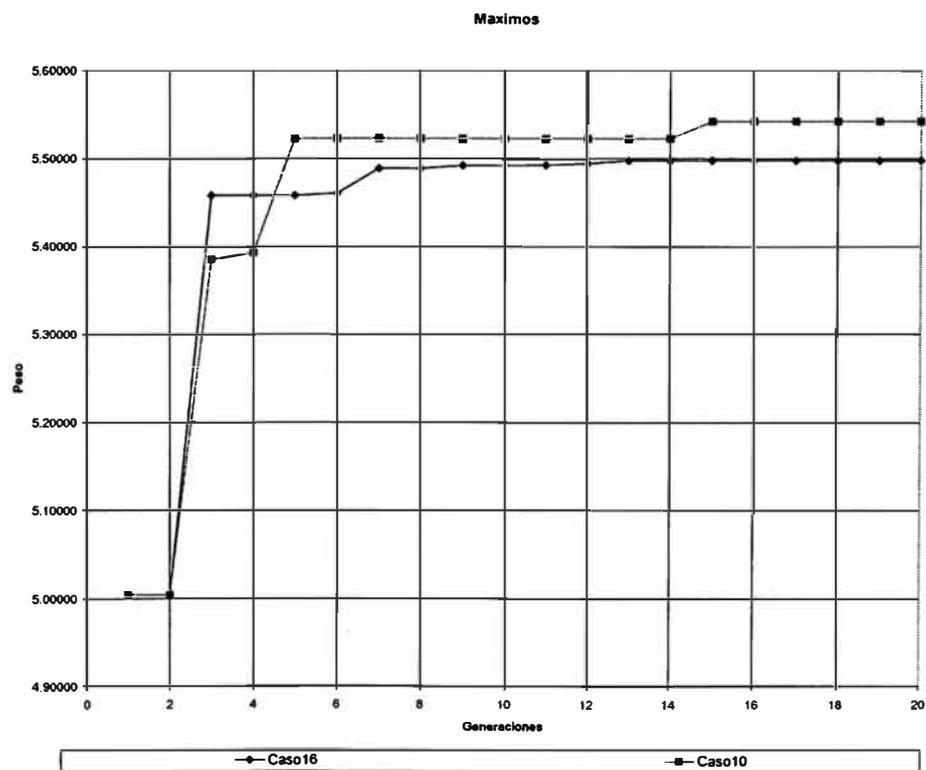
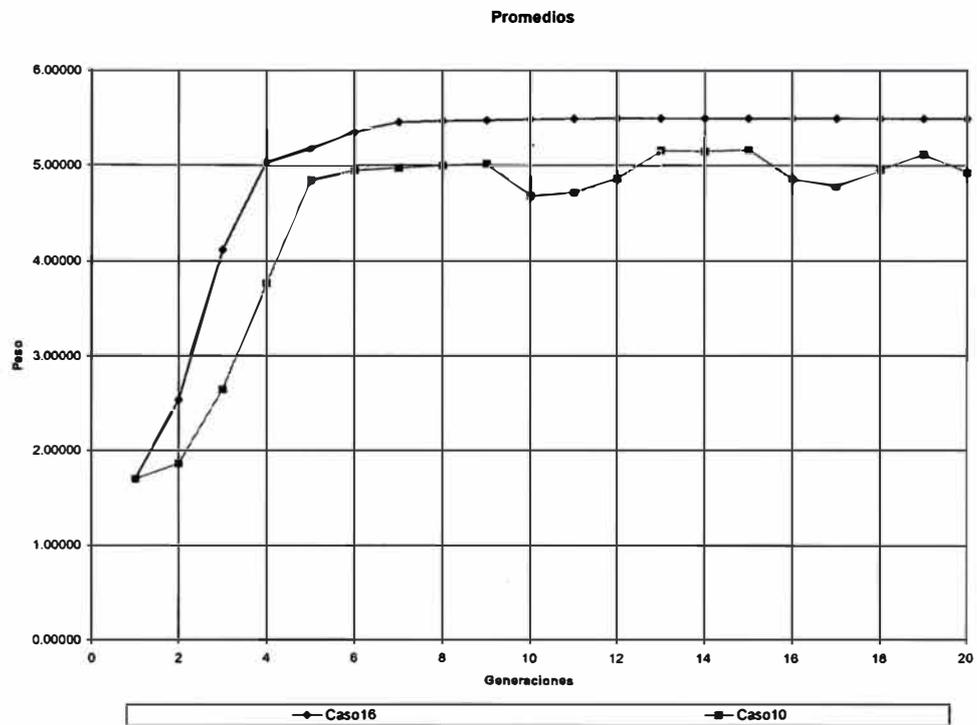
Promedios

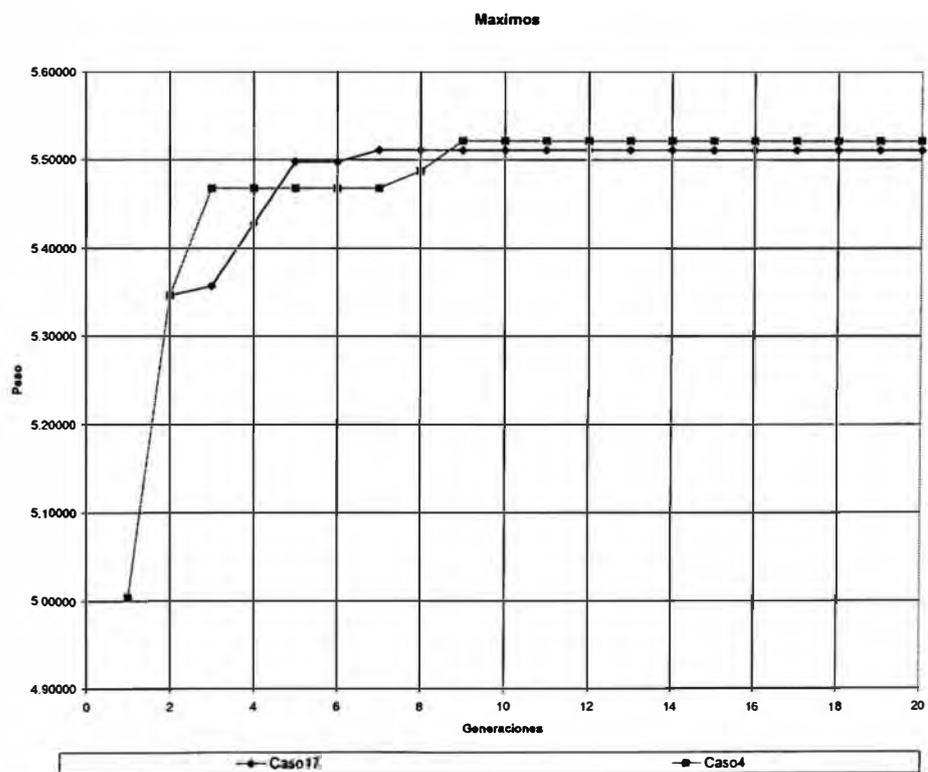
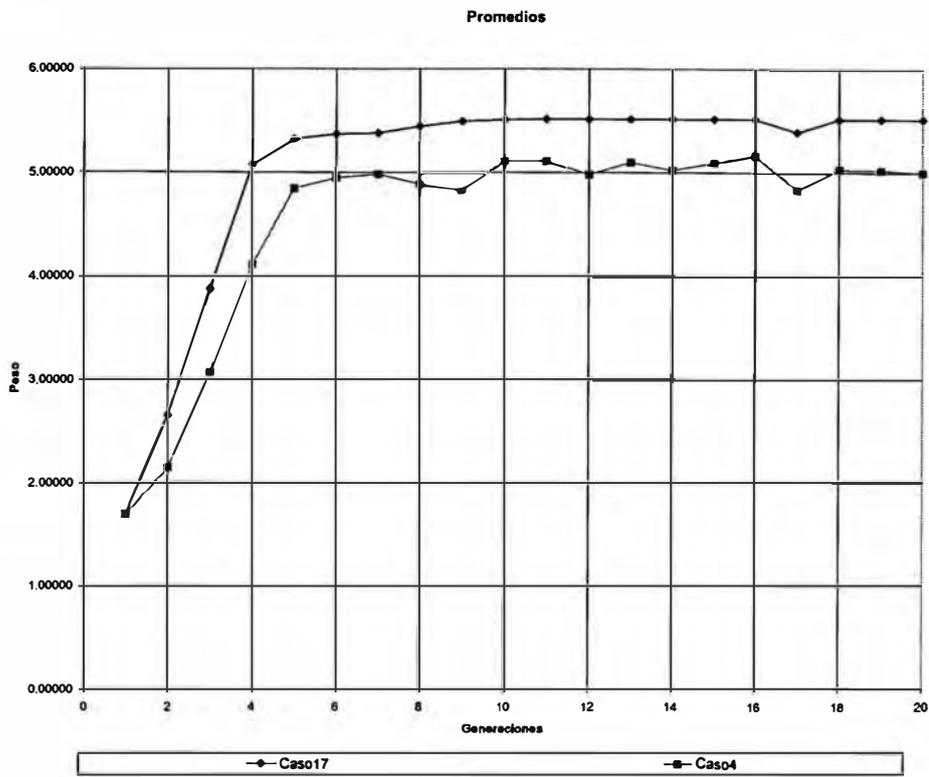


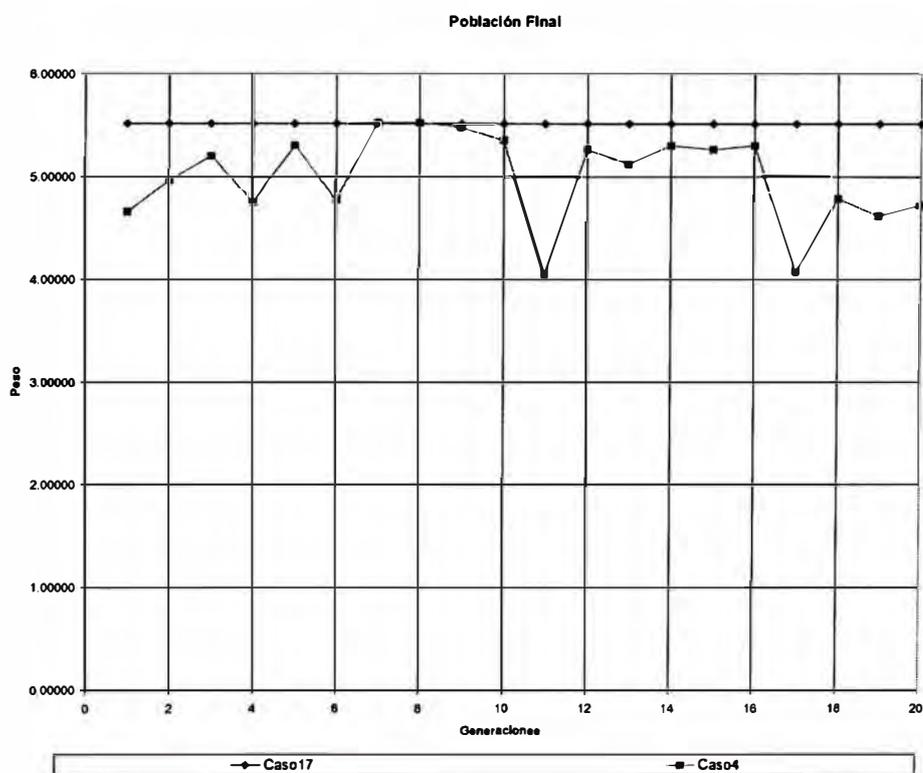
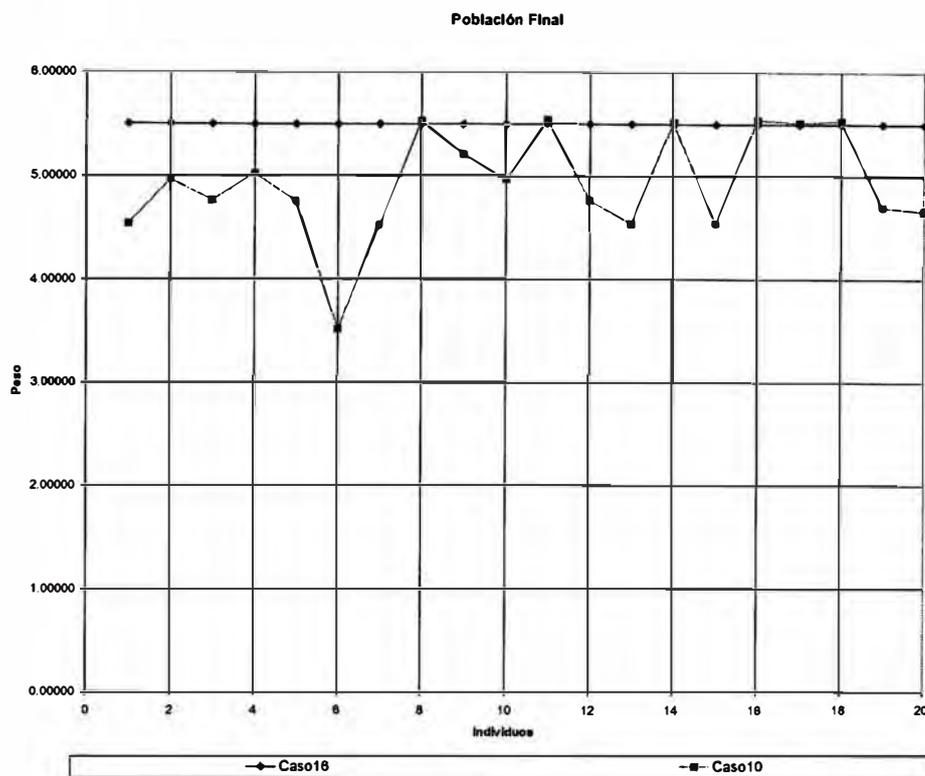


Se observa que el centro del problema de mutación no afecta notablemente al comportamiento del máximo; sin embargo si afecta al promedio de los individuos, aquí se comprueba que el operador de mutación en cierta forma crea un efecto de diversificación en la población, de acuerdo a eso, se puede mantener la misma probabilidad de mutación, es decir, se seleccionan los casos 4 y 10.

Análisis del reemplazo: Caso 10, caso 4, caso 16 y caso 17





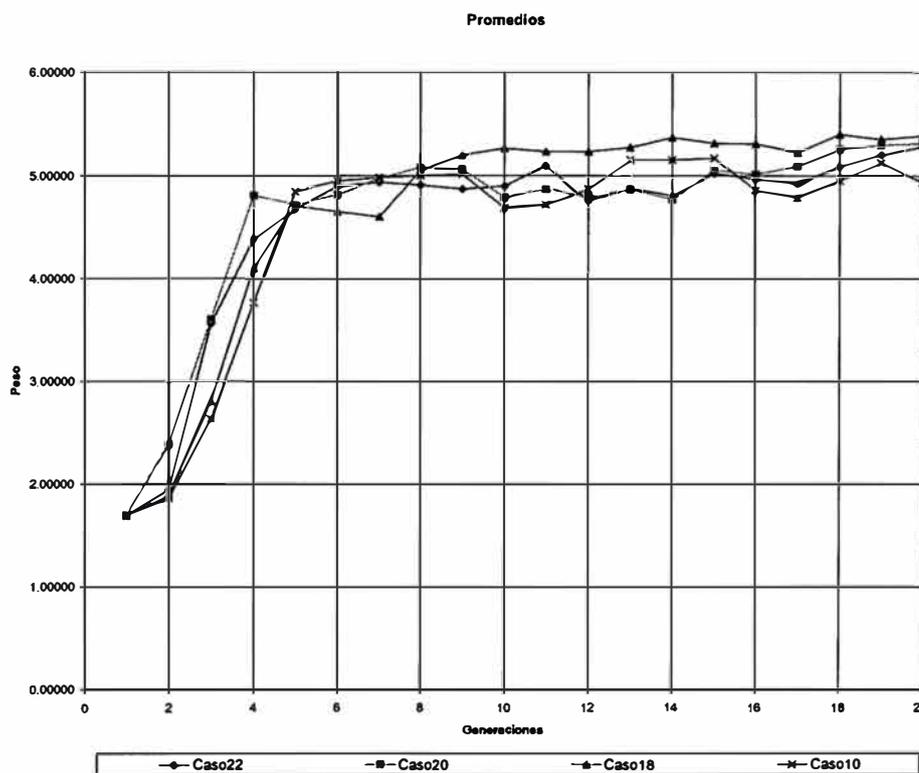


Al analizar el efecto del reemplazo, se obtiene que al reemplazar el más débil se consigue que solo exista un peso en la población, es decir, todos los individuos tienen el

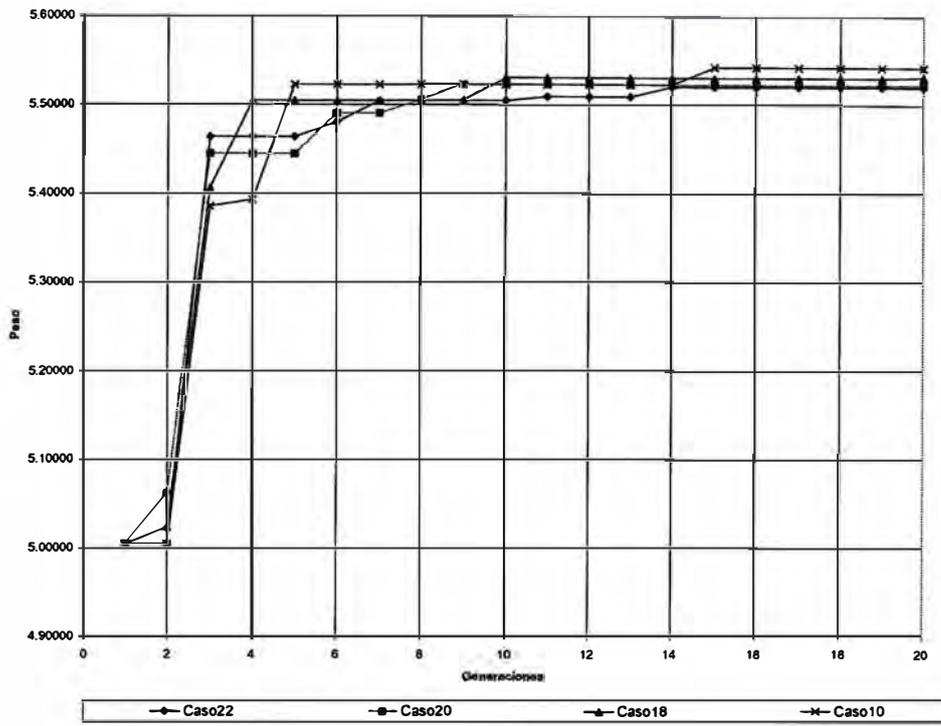
mismo peso; esto se puede ver en la población final, a pesar de que logra una convergencia acelerada, se limita a un óptimo no pudiendo continuar en la búsqueda en las futuras generaciones.

Análisis del cruce: Caso 10, caso 4, caso 18, caso 19, caso 20, caso 21, caso 22 y caso

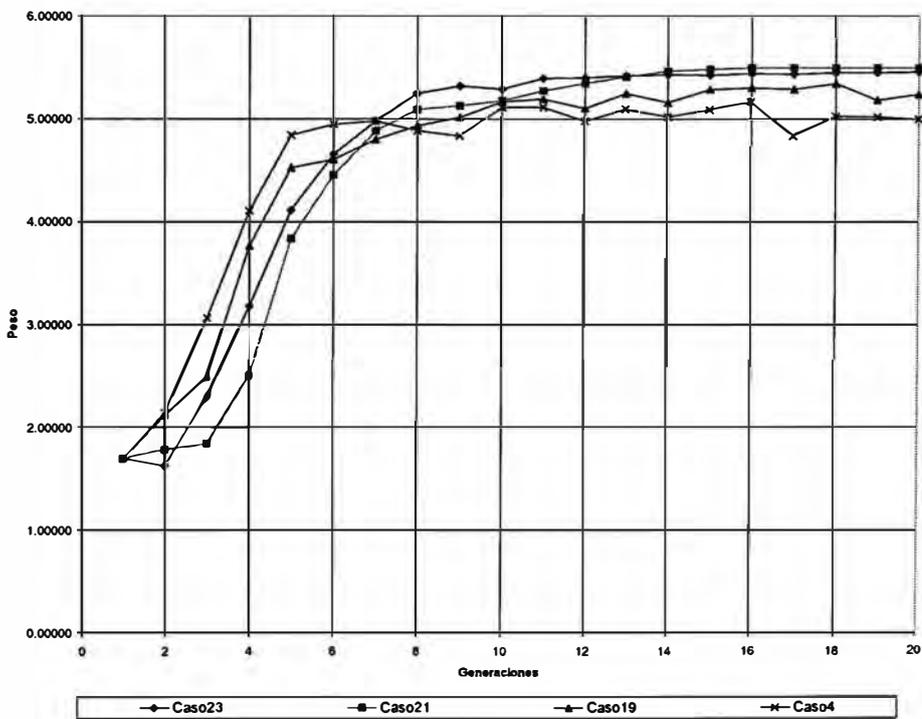
23

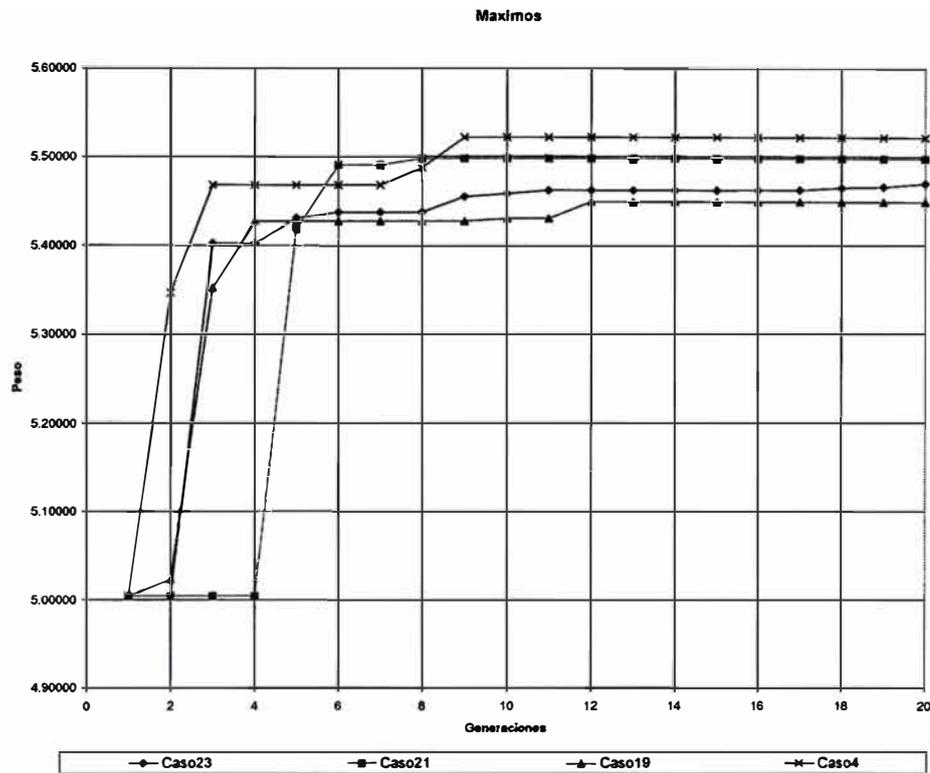


Maximos



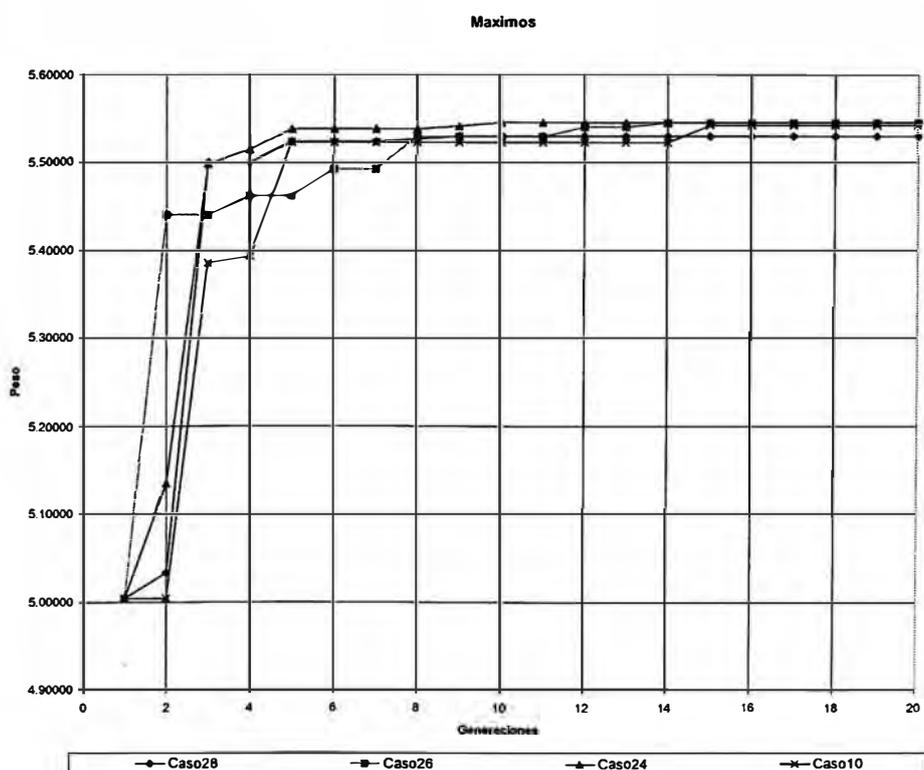
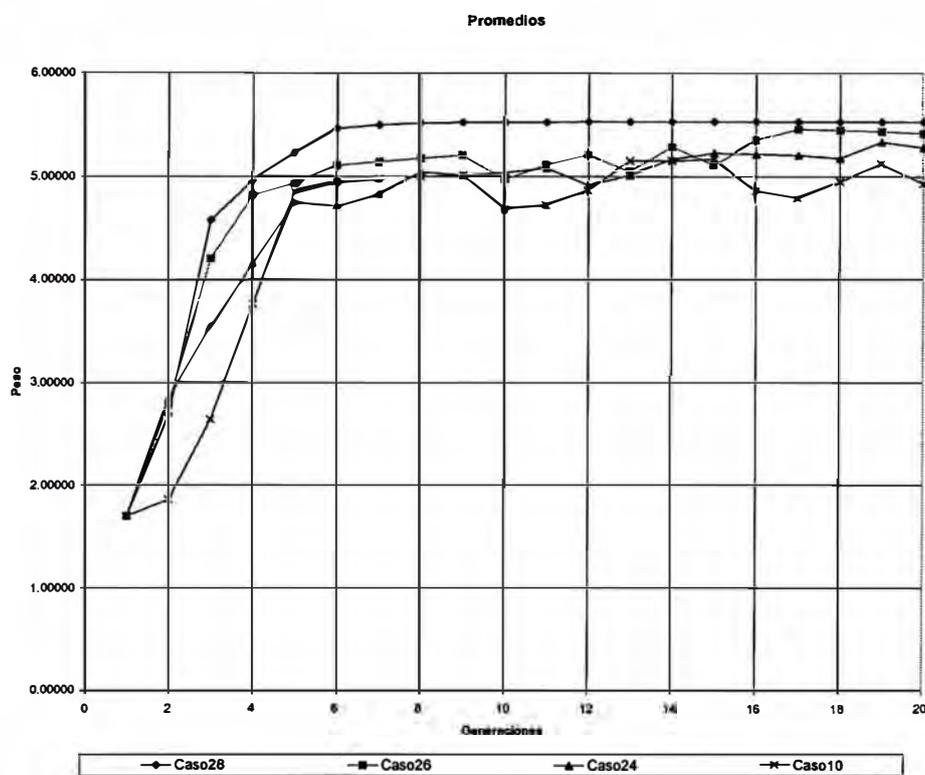
Promedios

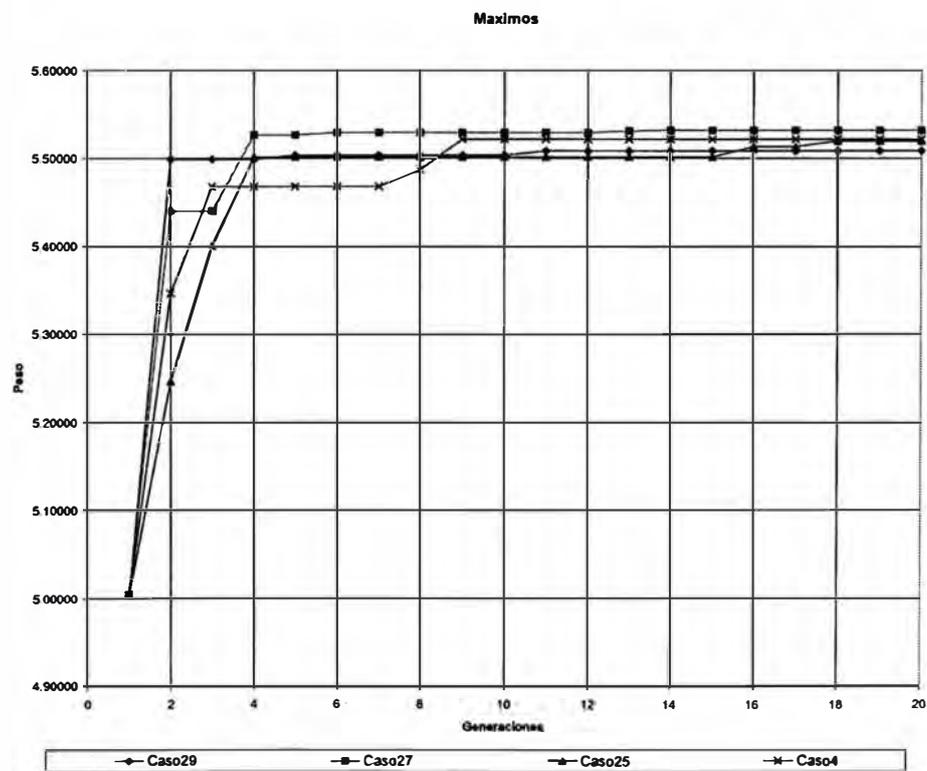
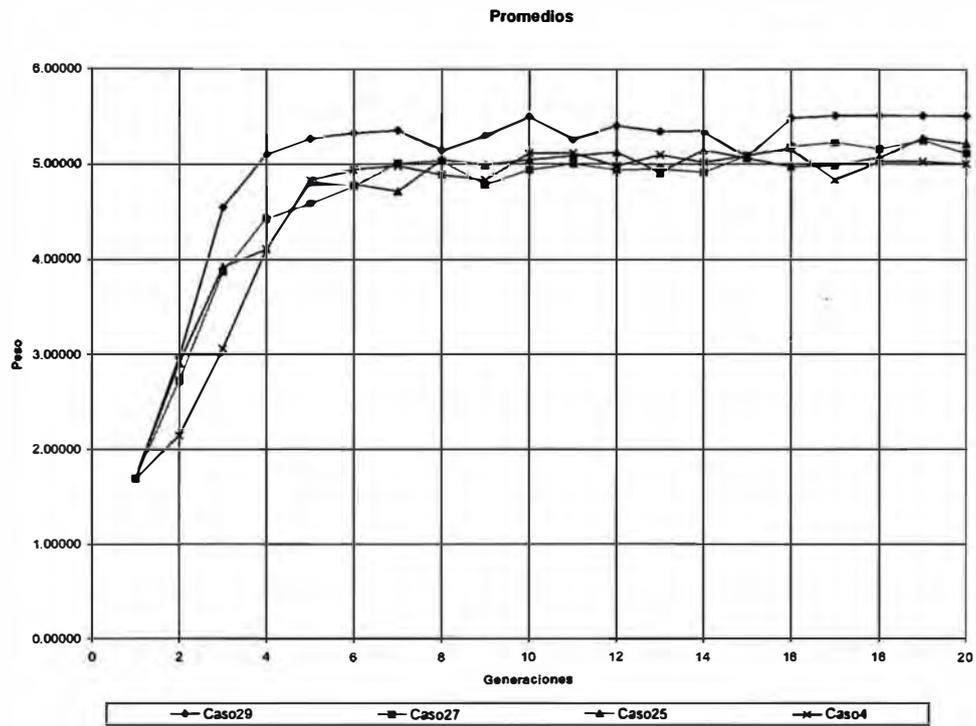




Al ver diferentes tipos de cruce, se observa que para este algoritmo los mejores individuos se obtienen en el cruce de n-puntos(caso 4 y 10), adicionalmente los promedios de estos casos son bajos, así que se podría conseguir mejores individuos en futuras generaciones. De esa manera se seleccionan los casos 4 y 10 como los mejores.

Análisis de la selección: Caso 10, caso 4, caso 24, caso 25, caso 26, caso 27, caso 28 y caso 29

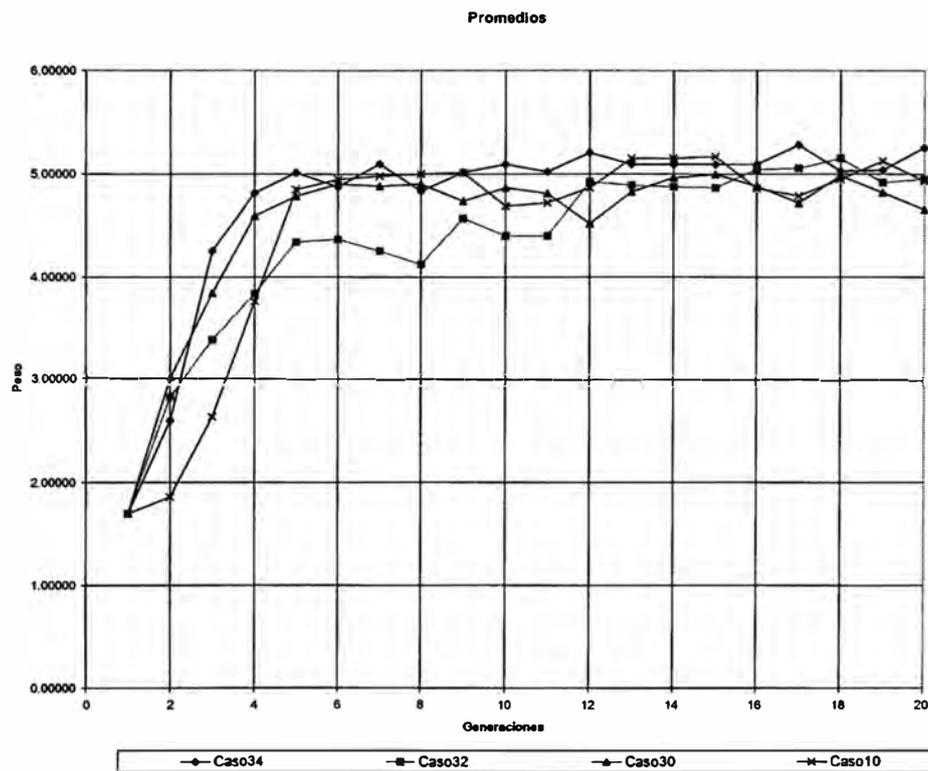




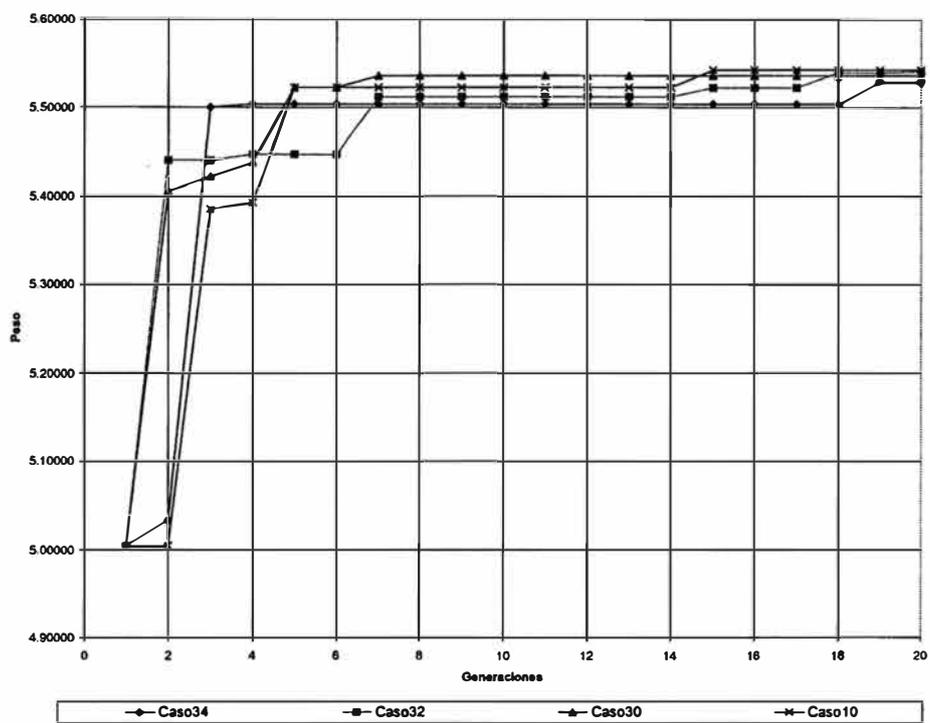
Se puede ver que con los métodos de selección estocástica universal y estocástica sobrante se obtienen los mejores resultados, a su vez sus promedios se comportan de manera

similar; pero sus promedios son más altos que en los casos base, con lo cual no se asegura mejores individuos, por lo que se opta por utilizar los casos base.

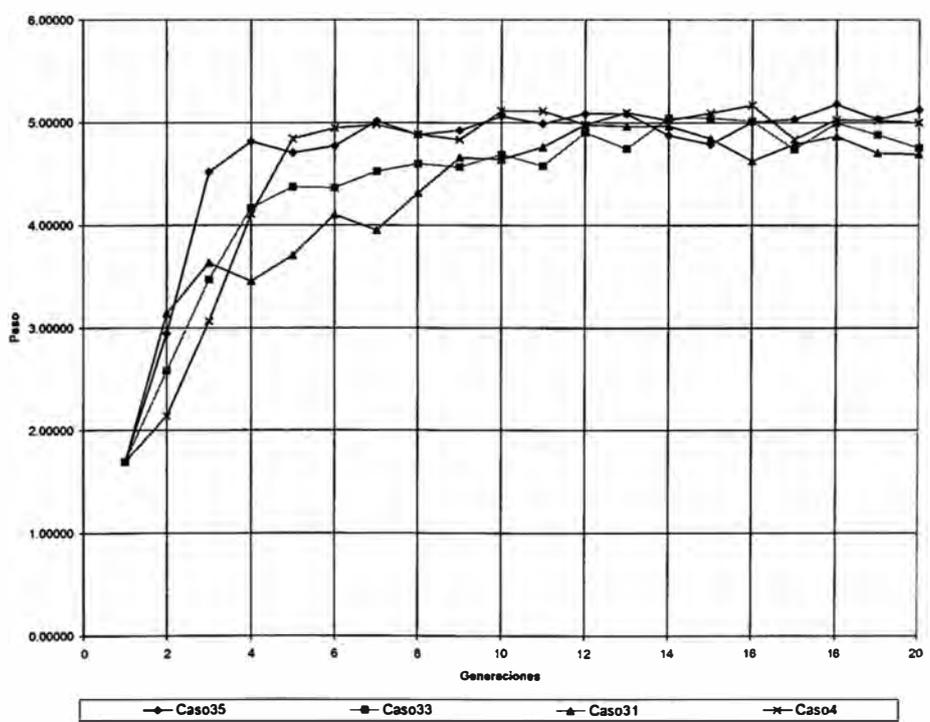
Análisis de la compartición: Caso 10, caso 4, caso 30, caso 31, caso 32, caso 33, caso 34 y caso 35

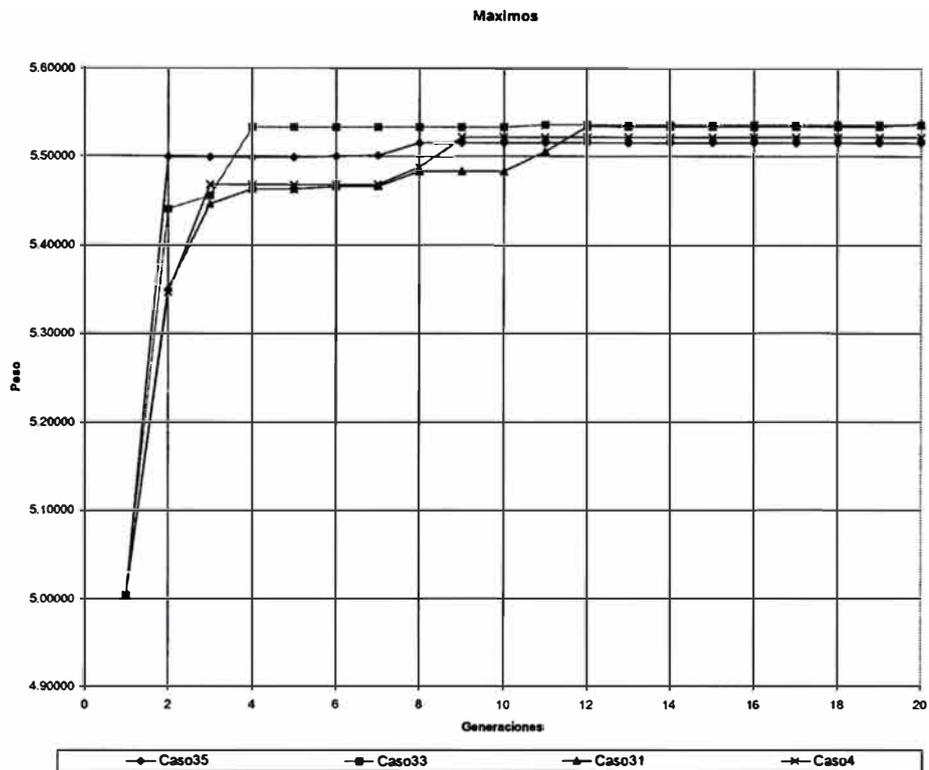


Máximos



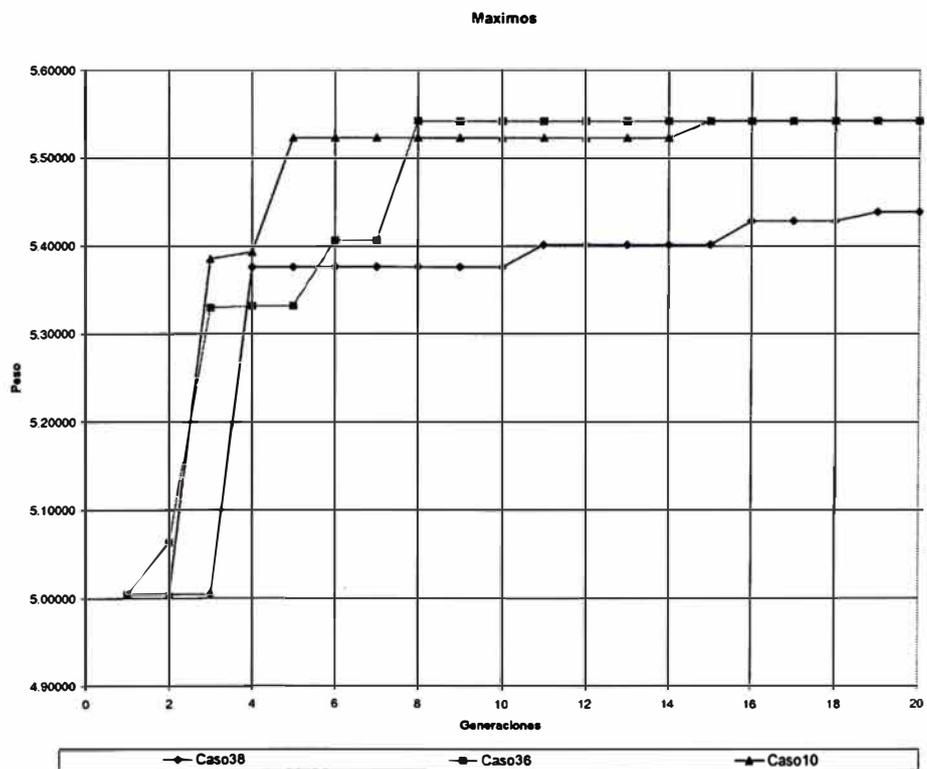
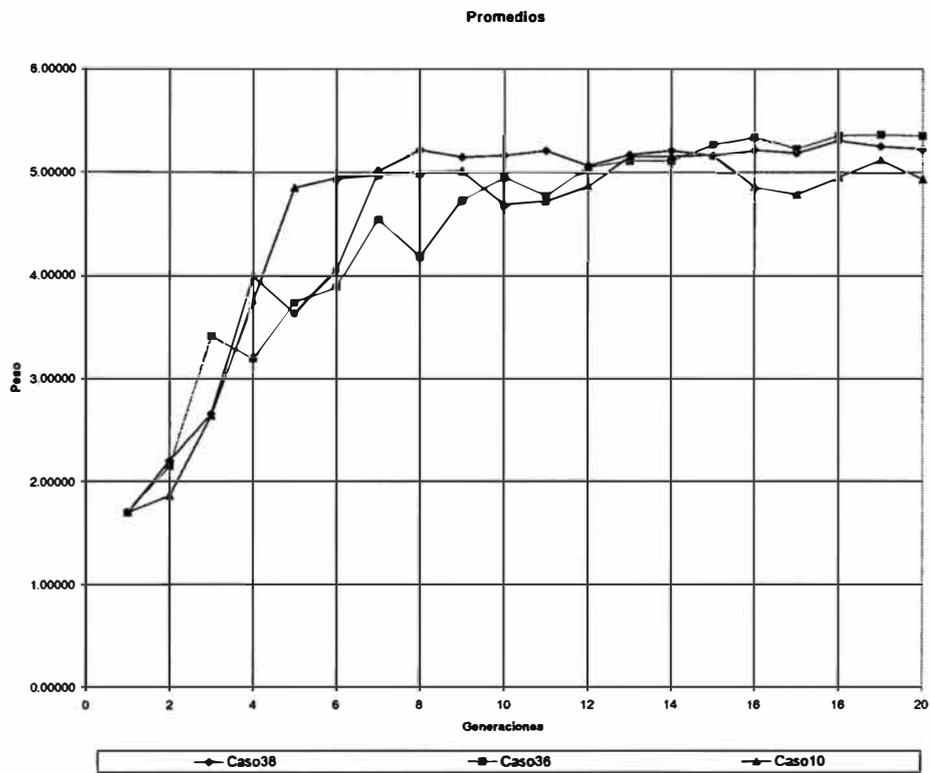
Promedios



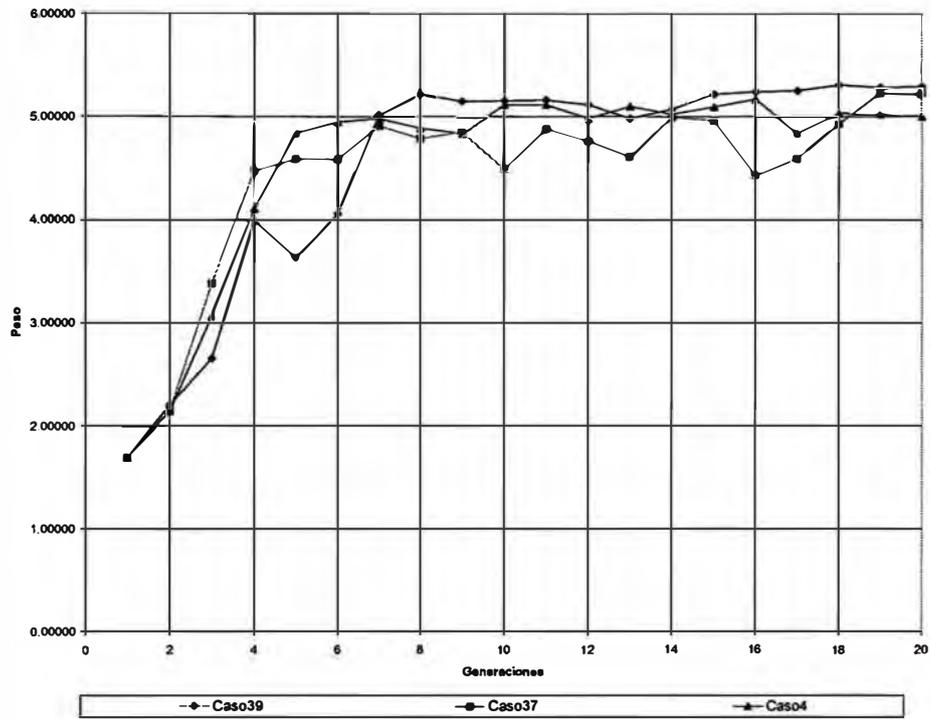


Ahora se aprecia un tipo de niching (compartición) con respecto a lo que se encuentra en los algoritmos genéticos con operadores básicos, se ve que los promedios bajan y mejores individuos, pero la inclusión del niching requiere algo más de tiempo, así que nos vamos a quedar con los casos base.

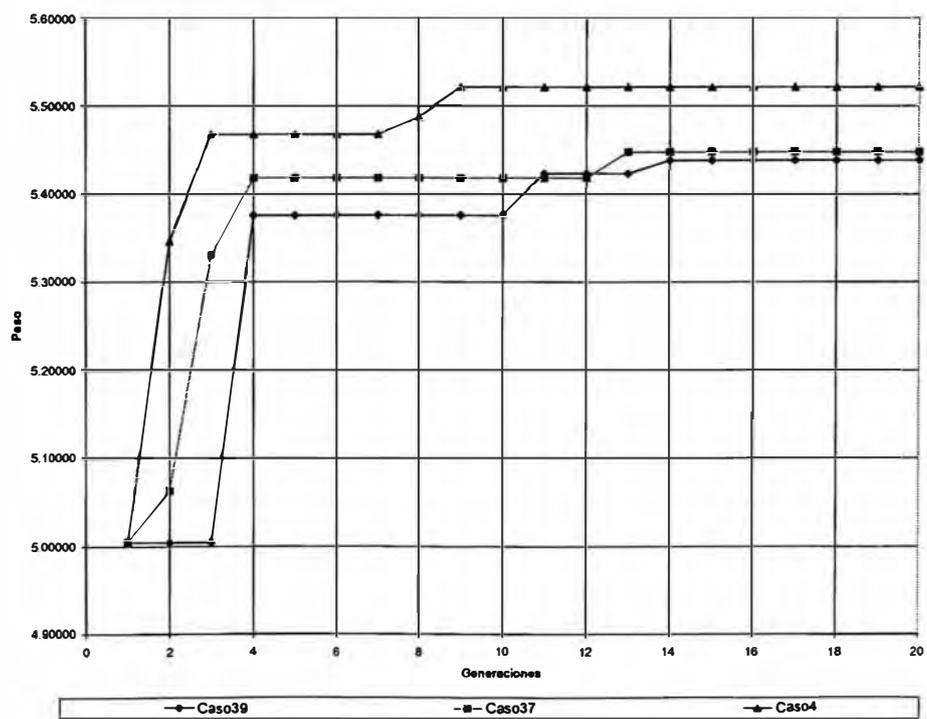
Análisis del apiñamiento: Caso 10, caso 4, caso 36, caso 37, caso 38 y caso 39



Promedios

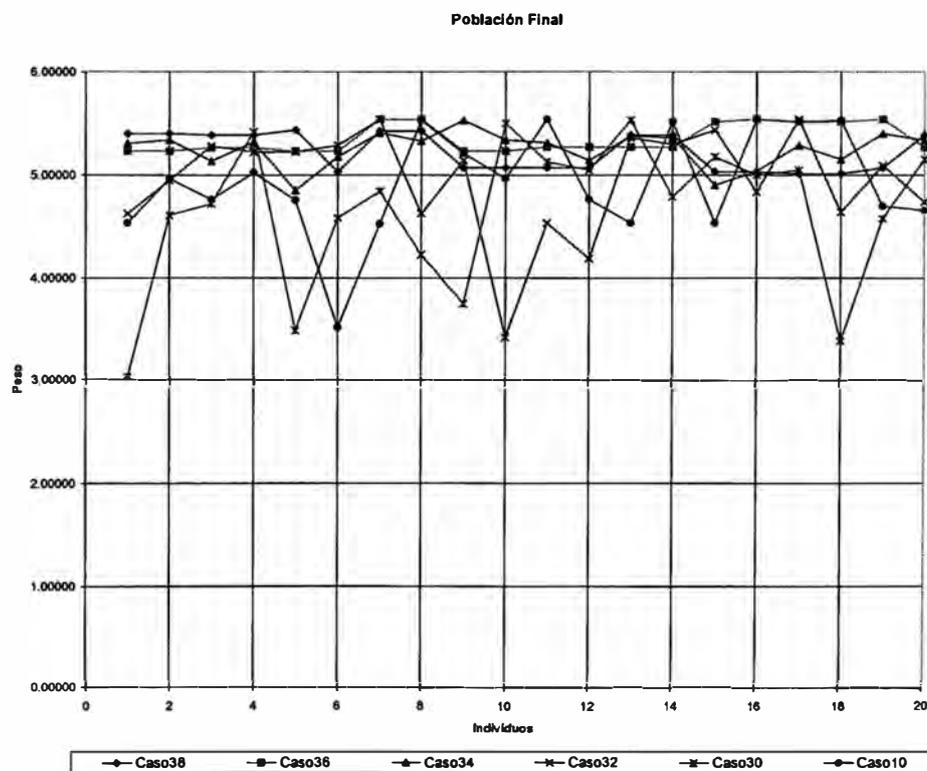


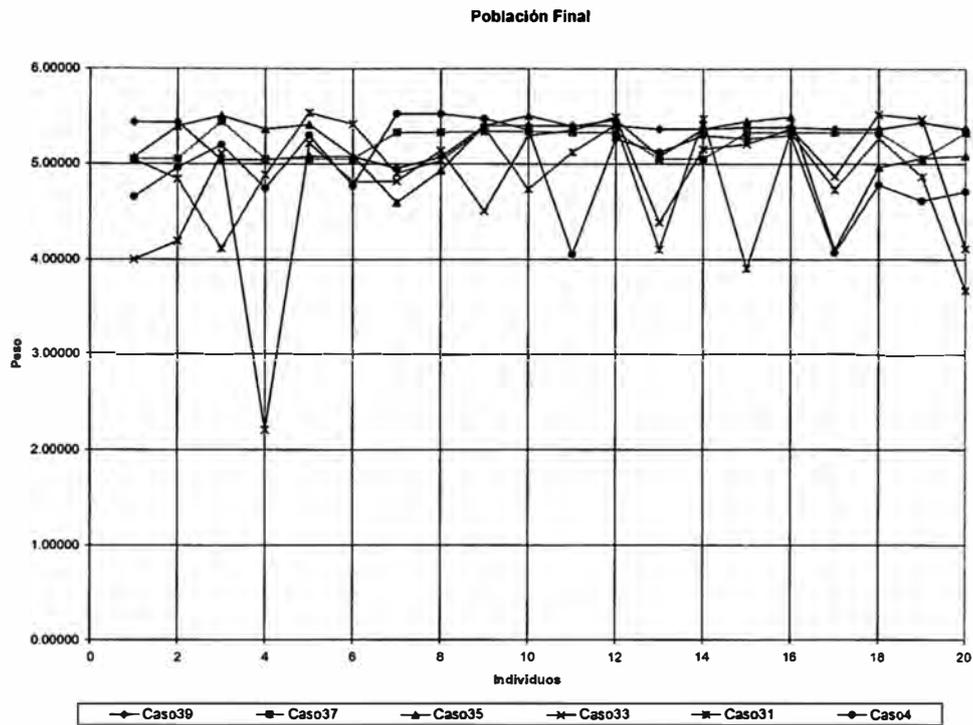
Maximos



Se observa que el apiñamiento determinístico no produce mejores individuos como en el caso anterior y los promedios crecen, por lo tanto en este algoritmo el apiñamiento determinístico no produce mayor diversificación. Entonces se concluye que los casos base son los mejores.

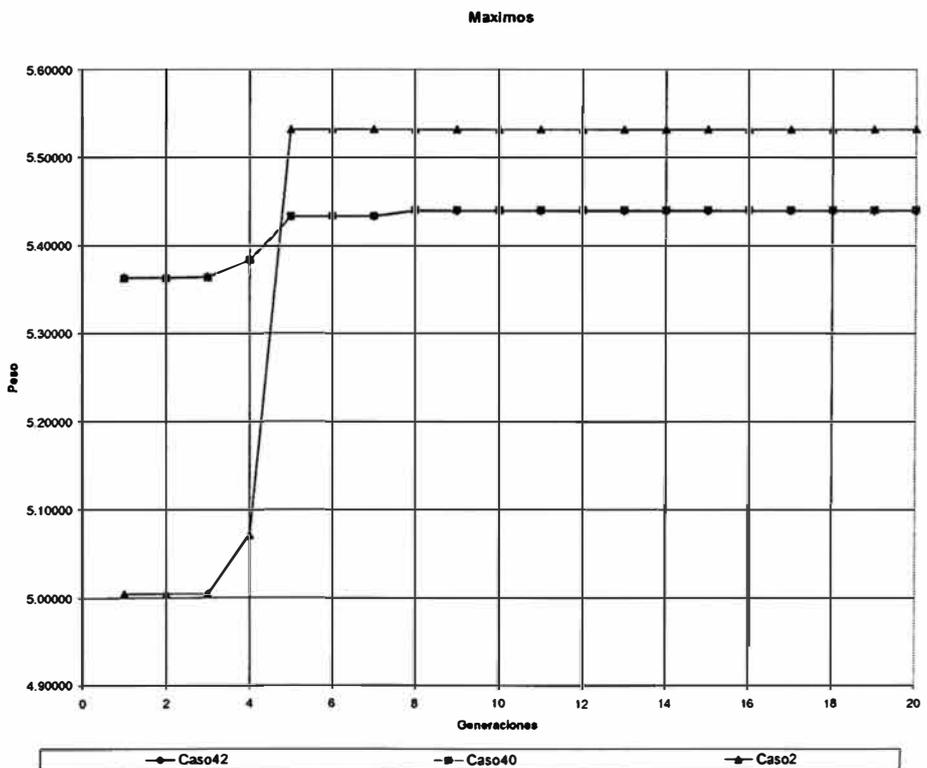
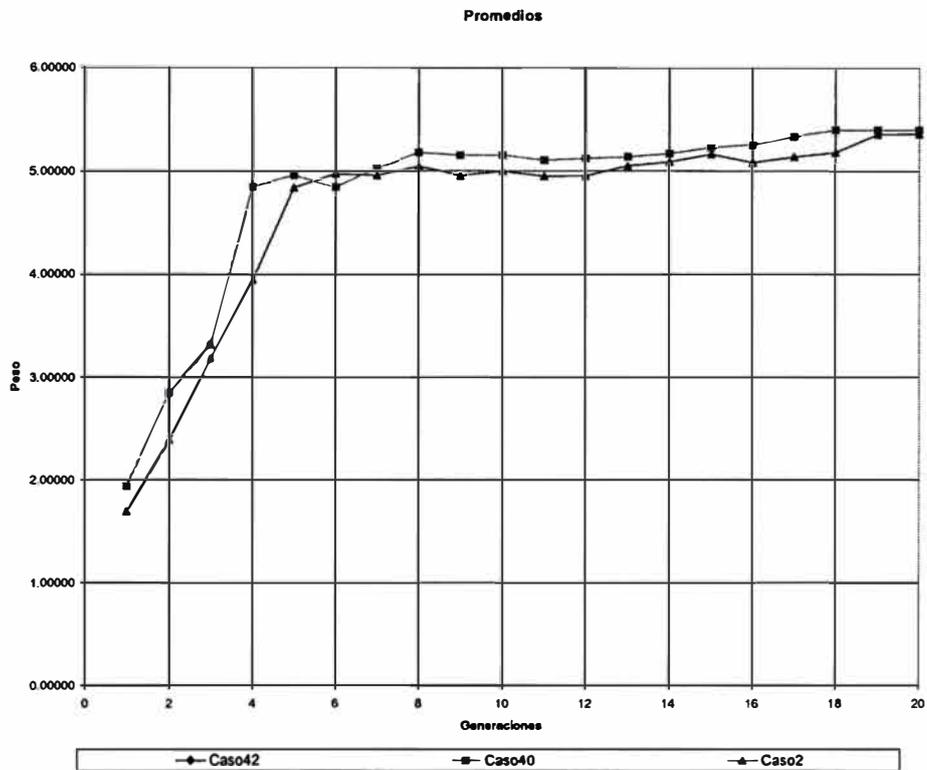
Análisis del niching(nicheo): Caso 10, caso 4, caso 30, caso 31, caso 32, caso 33, caso 34, caso 35, caso 36, caso 37, caso 38 y caso 39

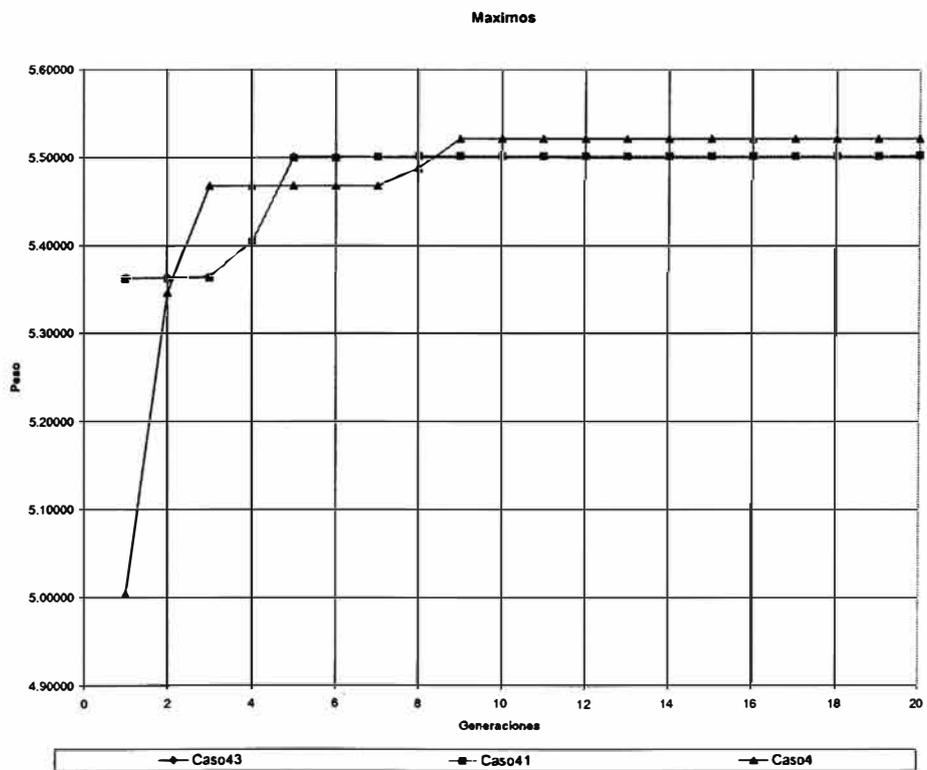
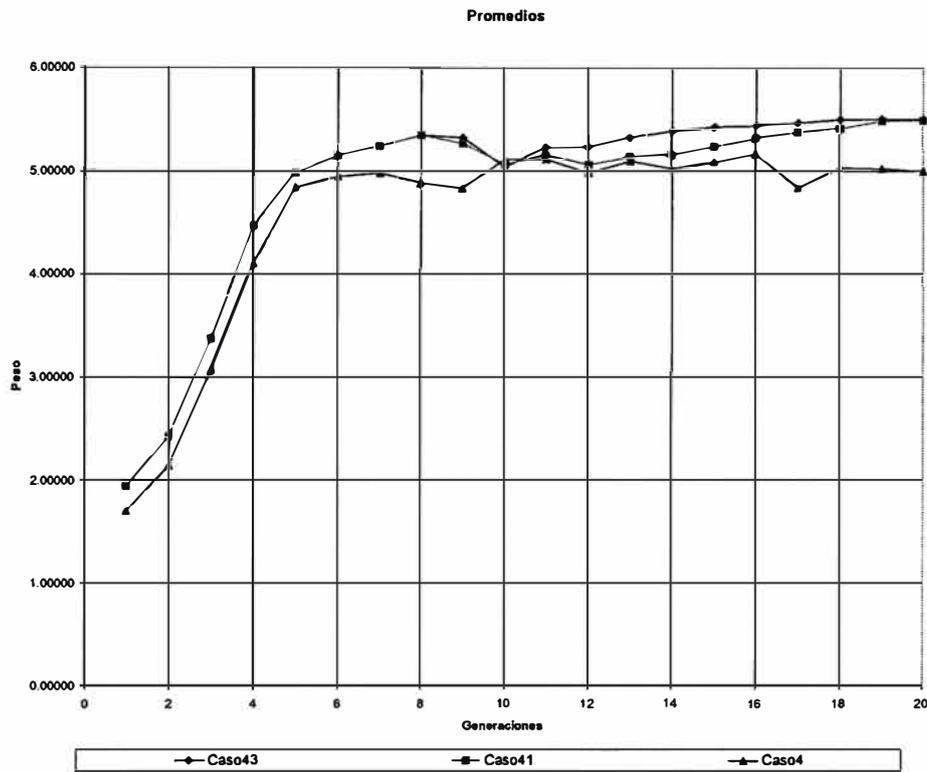




Aquí se comprueba lo que se veía en los casos anteriores, la mejor diversificación se produce con la compartición; como se vio en el capítulo III el niching se usa para encontrar varios puntos óptimos, al observar las gráficas se deduce que no habría otro punto óptimo ya que no se ve una acumulación de puntos alrededor de otro peso.

Análisis del ruido A: Caso 10, caso 4, caso 40, caso 41, caso 42 y caso 43

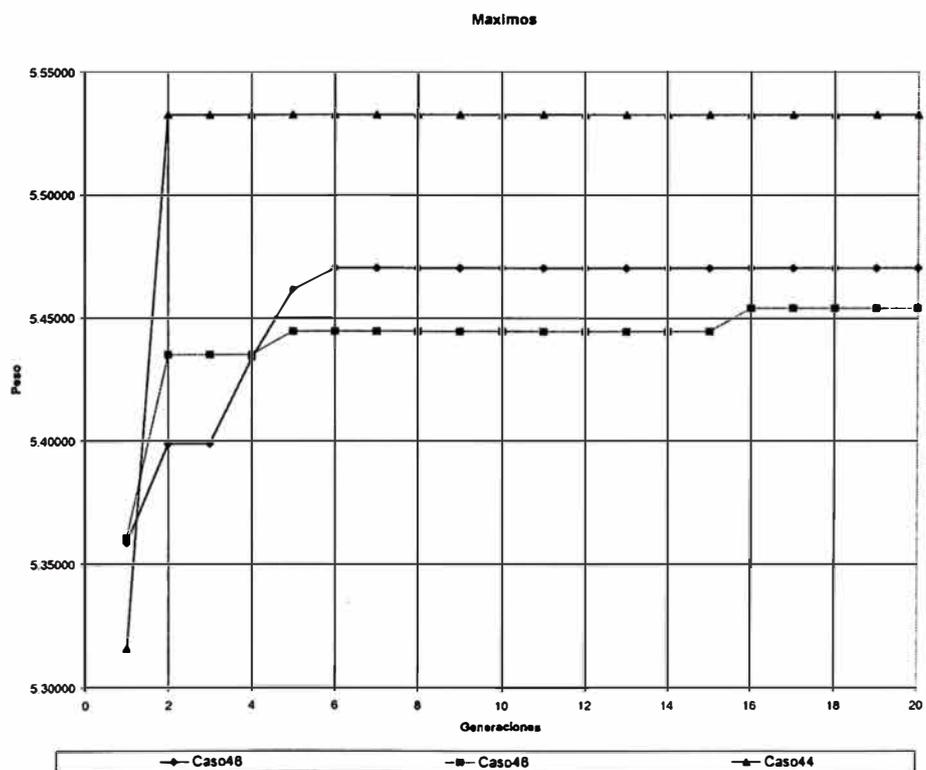
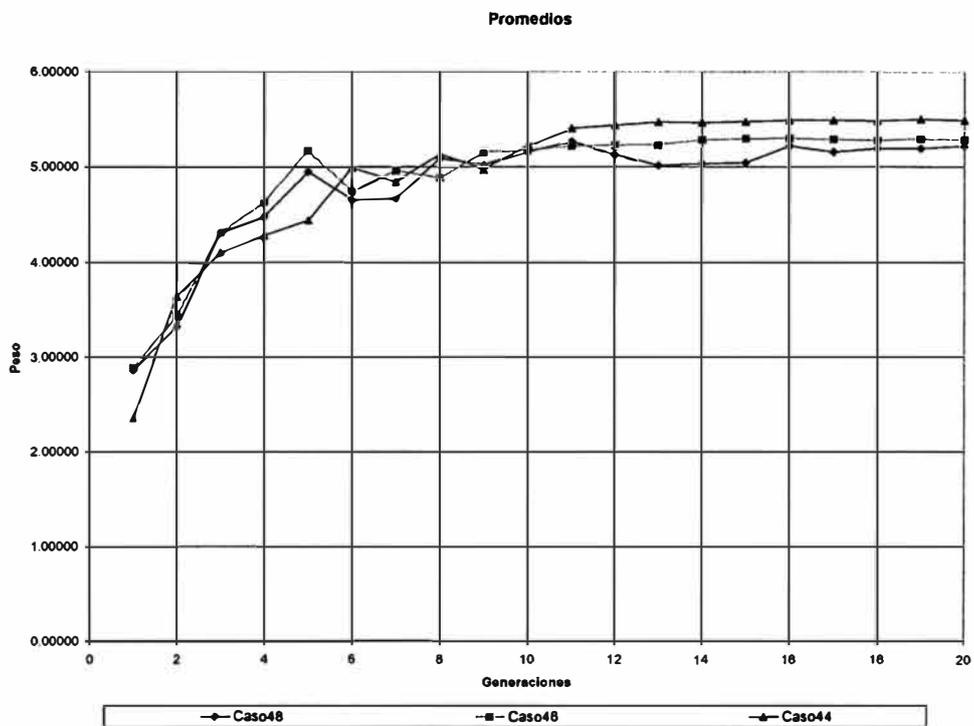


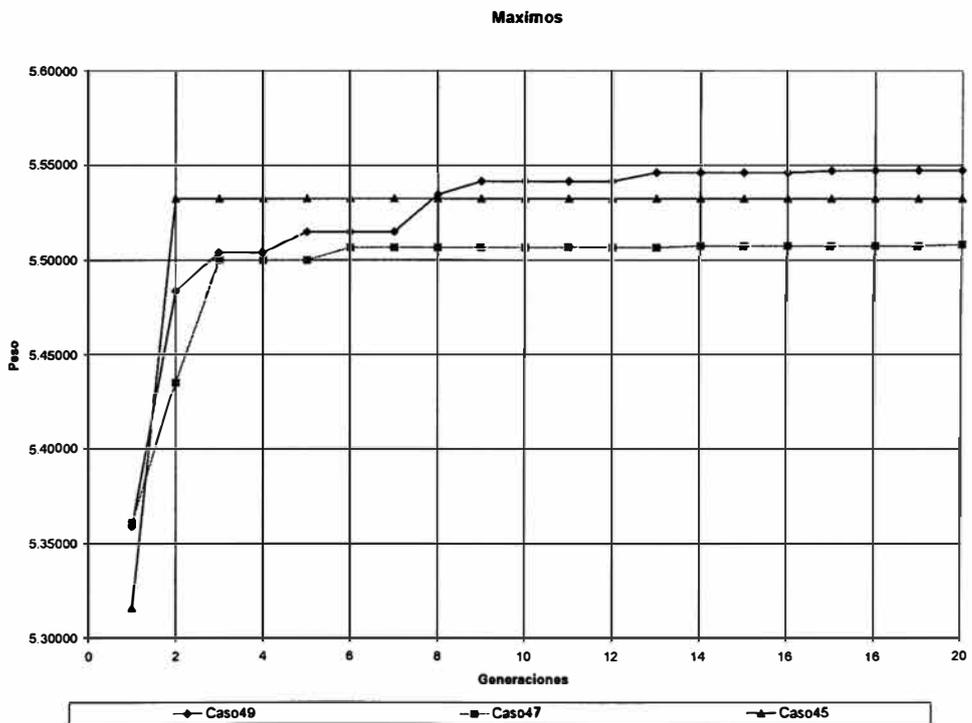
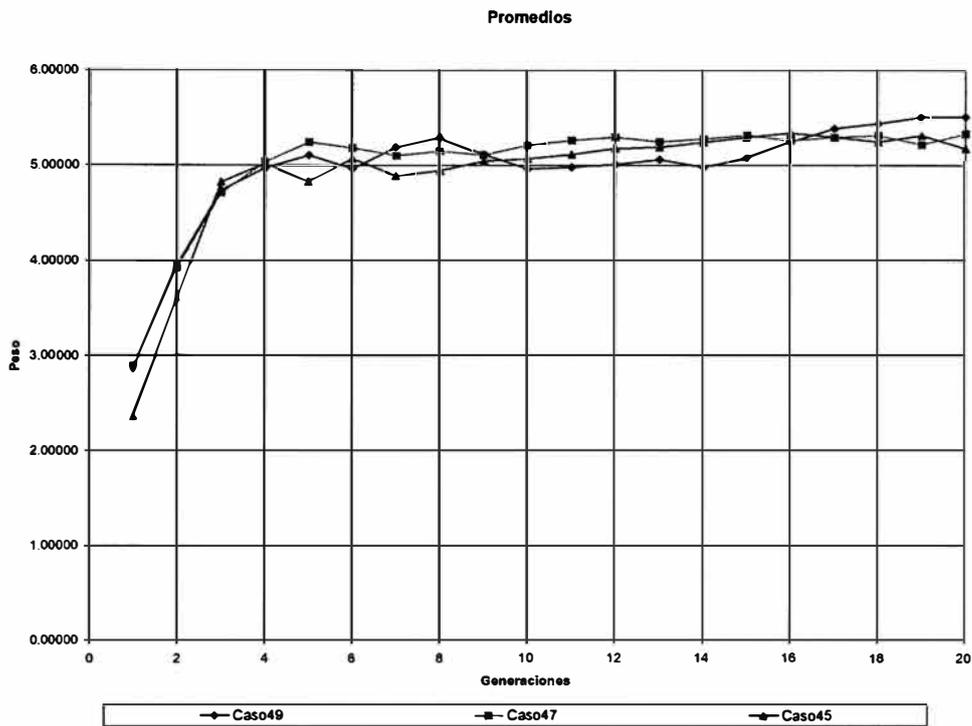


Se observa que con cromosomas de longitud 30, el ruido no produce ninguna mejora, se concluye que el ruido no mejora el resultado del algoritmo genético.

Análisis del ruido B: Caso 10, caso 4, caso 44, caso 45, caso 46, caso 47, caso 48 y caso

49

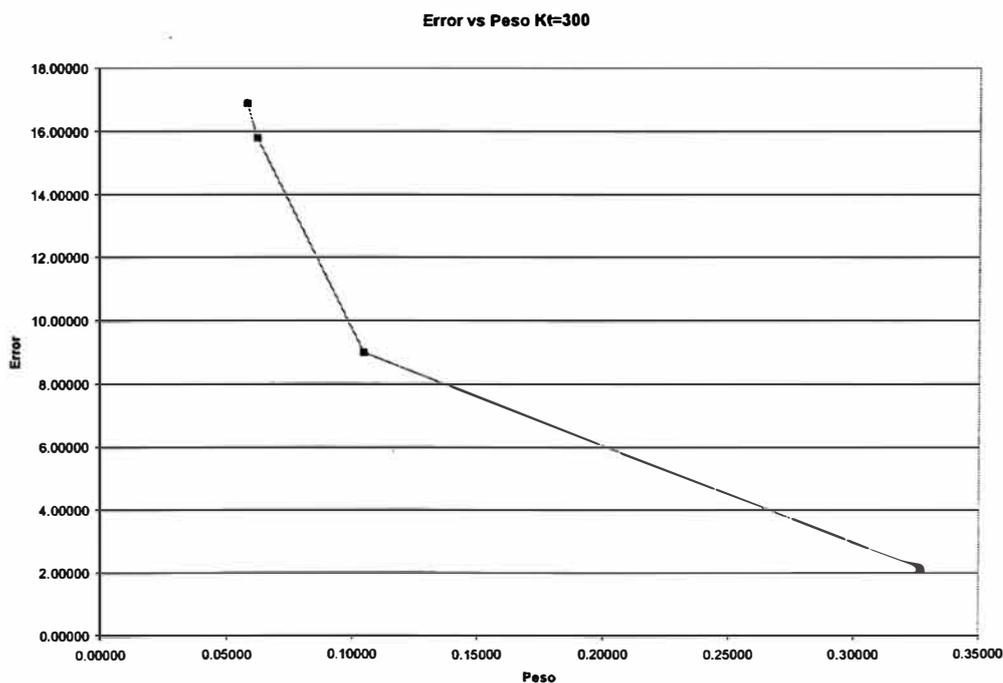
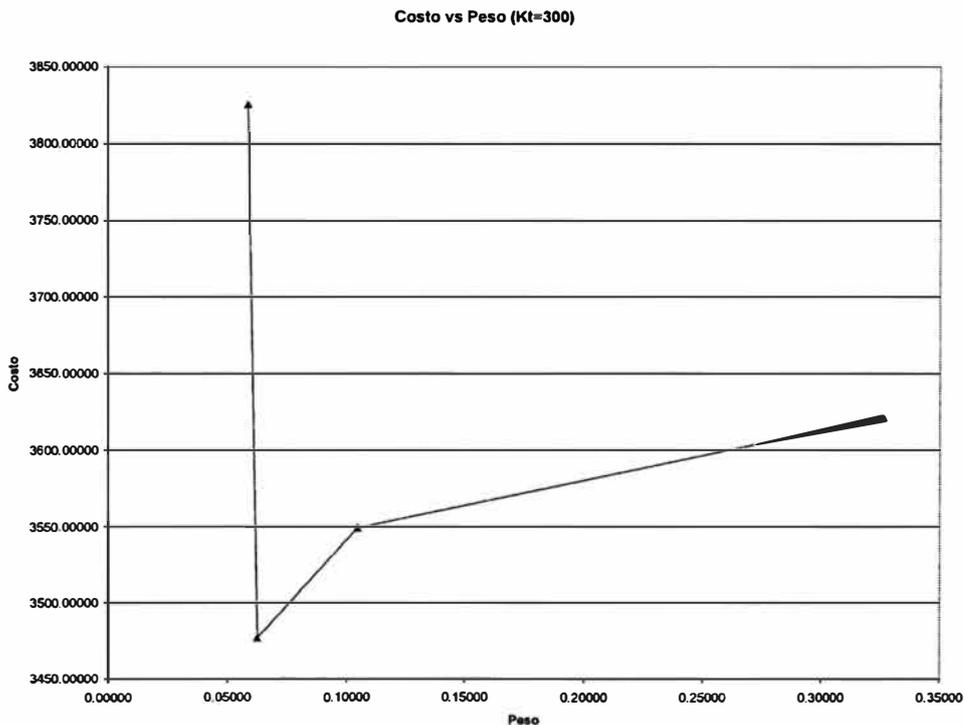


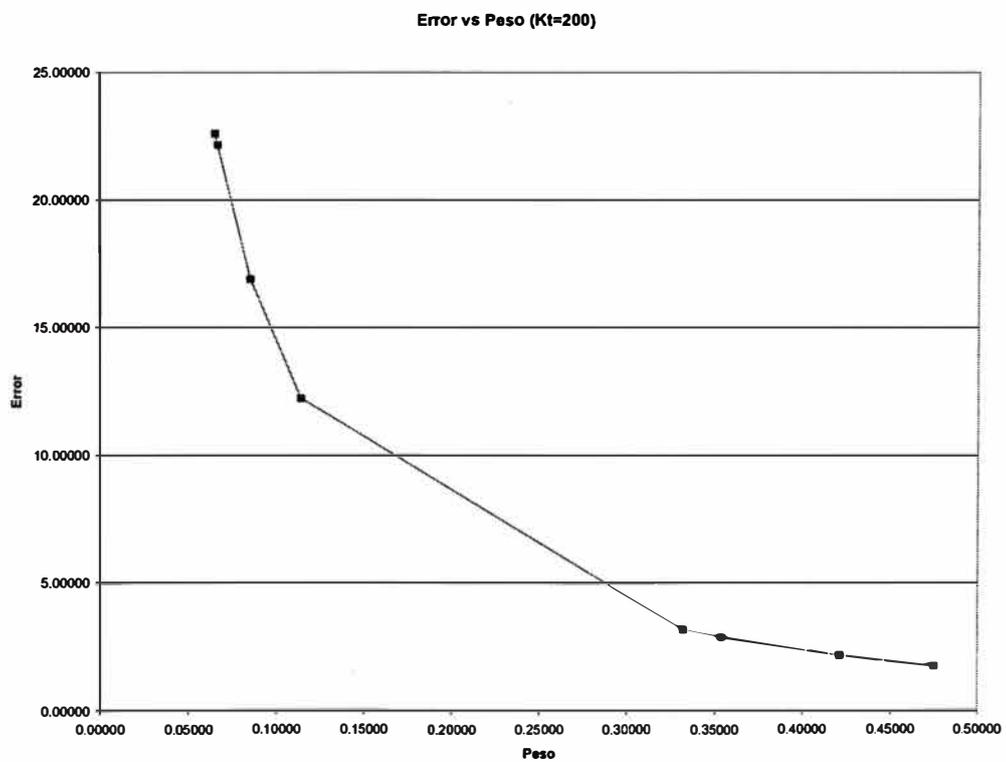
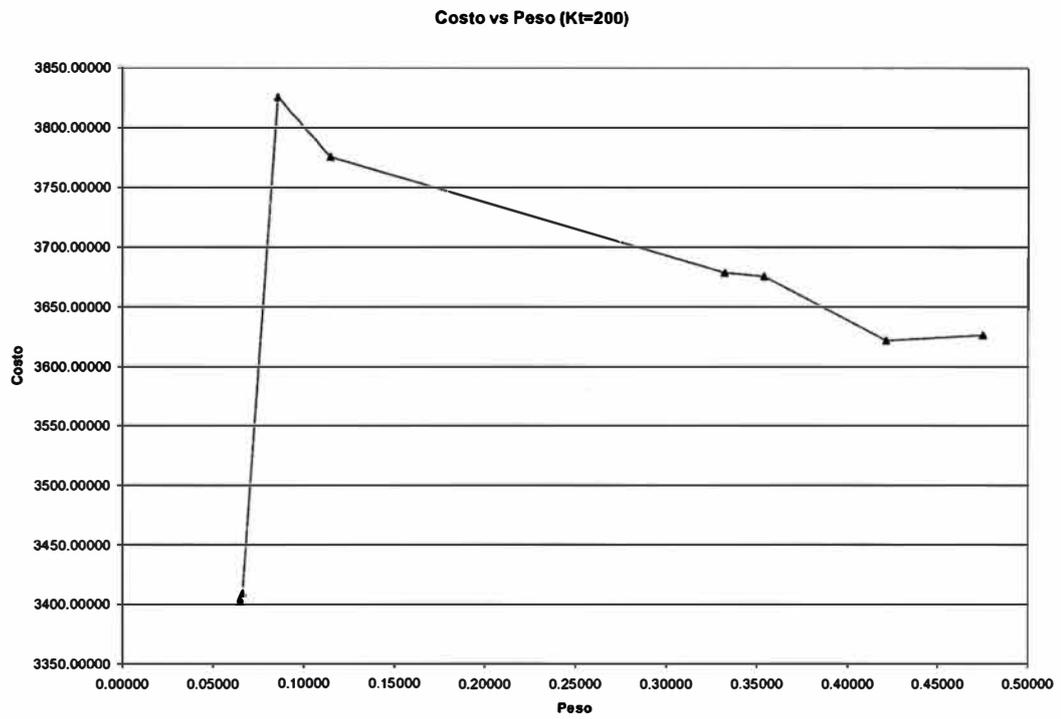


Aquí se observa que el ruido mejora un poco los resultados, esto se debe a que la longitud del cromosoma se ha reducido a la mitad, pero se ve que el mejor individuo es similar al promedio e inferior al obtenido con cromosomas de longitud 30, así que no se

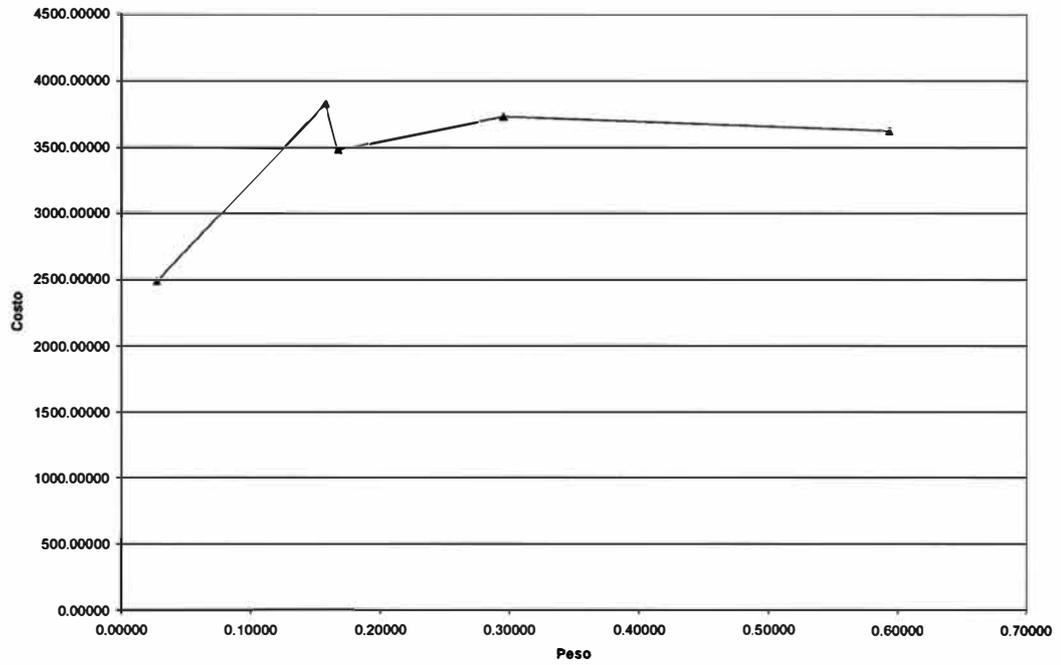
podría obtener mejores individuos con el ruido en el caso de las centrales hidráulicas.

En el caso de la *optimización térmica* se prueba el sistema sin pérdidas, es decir, realizando un despacho económico puro; primeramente se encuentra el valor de K_T de la ecuación (5.30). Se ha probado con 300, 200 y 100.

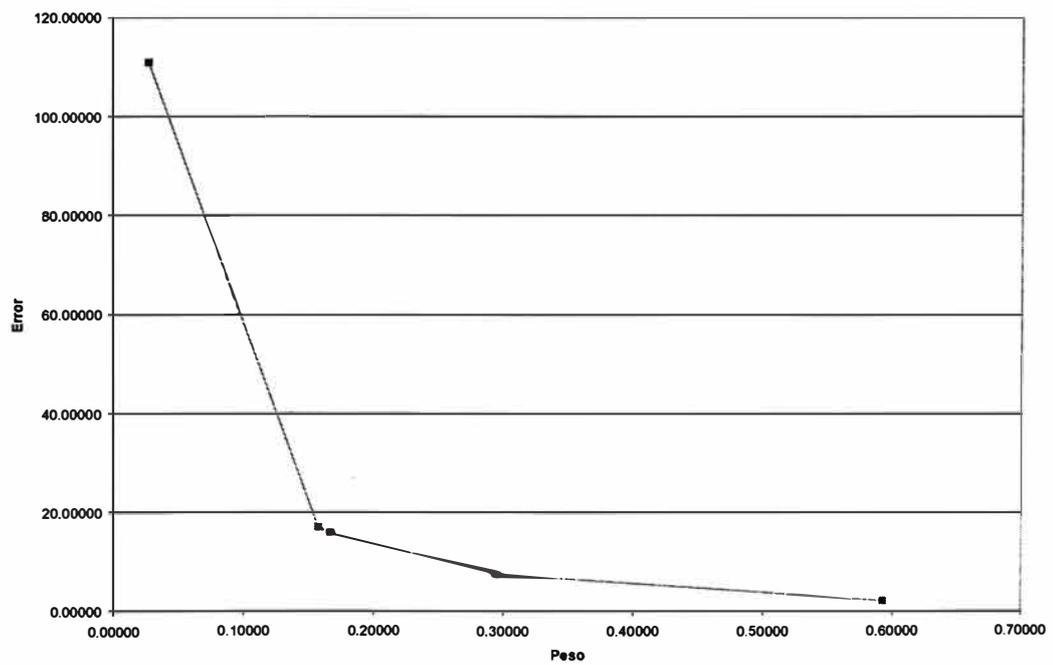




Costo vs Peso (Kt=100)



Error vs Peso (Kt=100)



Se observa en todos los casos que conforme aumenta el error el peso disminuye, esto se debe a la dependencia que existe entre el error y el peso, K_T funciona como una constante de proporcionalidad; en cambio el costo tiene un comportamiento diferente, pero se observa que con el valor de 200 el costo disminuye conforme aumenta el peso que es lo que se busca. Así que el valor de K_T que mejor se comporta es el valor de 200.

Para el análisis de la optimización térmica se ha utilizado la secuencia de operadores que se muestra en la tabla 7.2.

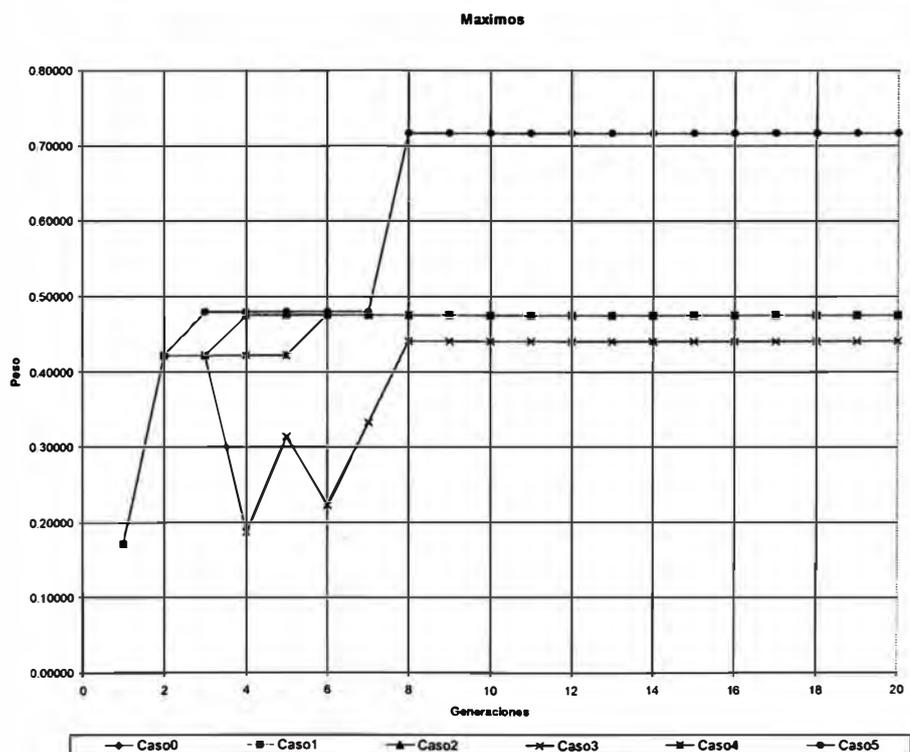
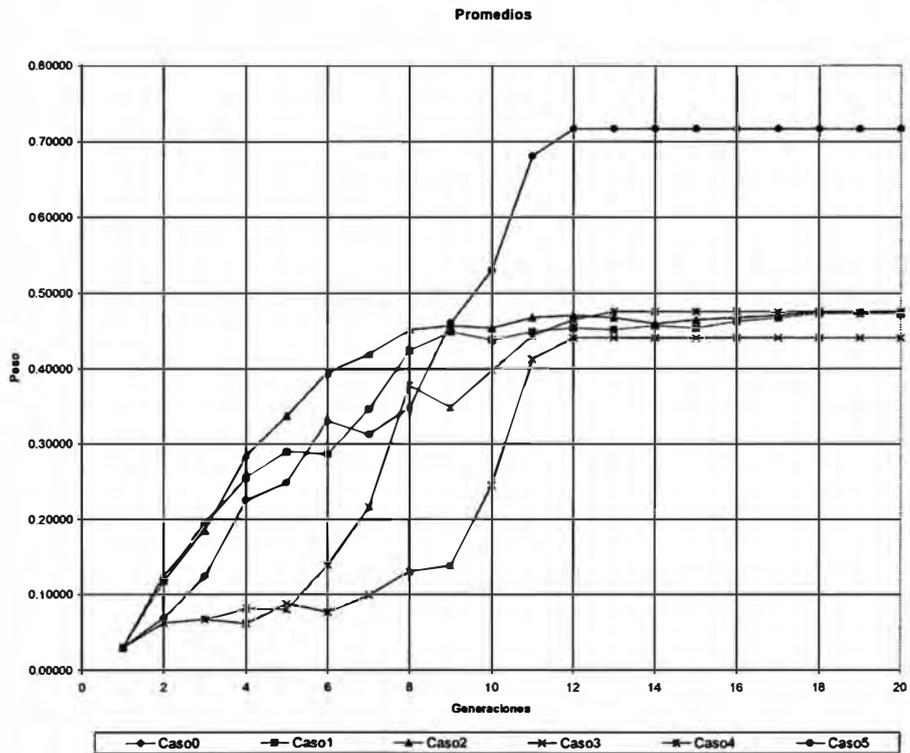
CUADRO 02

CASO	Longitud de cromosoma	The best	Preselección	Selección	Probabilidad de cruce	Numero de puntos	Tipo de Cruce	Probabilidad de mutación	Tipo de Mutación	Tipo de Reemplazo	Tipo de niching	Prevención de Incesto	Tipo de Ruido
0	10	Ninguno	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
1	10	Elitismo	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
2	10	Copiar el mejor	Ninguna	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
3	10	Ninguno	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
4	10	Elitismo	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
5	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,9	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
6	10	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
7	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
8	10	Elitismo	Ninguna	Ruleta	0,75	1	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
9	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	1	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
10	10	Elitismo	Ninguna	Ruleta	0,75	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
11	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	3	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
12	10	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,005	Simple inversión	Agregar	Ninguno	-----	Ninguno
13	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,005	Simple inversión	Agregar	Ninguno	-----	Ninguno
14	10	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,02	Simple inversión	Agregar	Ninguno	-----	Ninguno
15	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,02	Simple inversión	Agregar	Ninguno	-----	Ninguno
16	10	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	El más débil	Ninguno	-----	Ninguno
17	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	El más débil	Ninguno	-----	Ninguno
18	10	Elitismo	Ninguna	Ruleta	0,75	2	Mezcla con tabla	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
19	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	Mezcla con tabla	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
20	10	Elitismo	Ninguna	Ruleta	0,75	-----	Uniforme	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
21	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	-----	Uniforme	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
22	10	Elitismo	Ninguna	Ruleta	0,75	-----	Uniforme modificado	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
23	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	-----	Uniforme modificado	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
24	10	Elitismo	Ninguna	Estocastica sobrante	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
25	10	Copiar el mejor	Ninguna	Estocastica sobrante	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
26	10	Elitismo	Ninguna	Estocastica universal	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
27	10	Copiar el mejor	Ninguna	Estocastica universal	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
28	10	Elitismo	Ninguna	Torneo	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
29	10	Copiar el mejor	Ninguna	Torneo	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
30	10	Elitismo	Ninguna	Estocastica sobrante	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
31	10	Copiar el mejor	Ninguna	Estocastica sobrante	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
32	10	Elitismo	Ninguna	Estocastica universal	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
33	10	Copiar el mejor	Ninguna	Estocastica universal	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
34	10	Elitismo	Ninguna	Torneo	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
35	10	Copiar el mejor	Ninguna	Torneo	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Compartición	-----	Ninguno
36	10	Elitismo	-----	-----	0,75	-----	-----	0,01	Simple inversión	-----	Apilamiento	NO	Ninguno
37	10	Copiar el mejor	-----	-----	0,75	-----	-----	0,01	Simple inversión	-----	Apilamiento	NO	Ninguno
38	10	Elitismo	-----	-----	0,75	-----	-----	0,01	Simple inversión	-----	Apilamiento	SI	Ninguno
39	10	Copiar el mejor	-----	-----	0,75	-----	-----	0,01	Simple inversión	-----	Apilamiento	SI	Ninguno
40	10	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
41	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
42	10	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme
43	10	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme
44	5	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
45	5	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Ninguno
46	5	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
47	5	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Gauss
48	5	Elitismo	Ninguna	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme
49	5	Copiar el mejor	Escalamiento lineal	Ruleta	0,75	2	De n puntos	0,01	Simple inversión	Agregar	Ninguno	-----	Uniforme

Tabla 7.2 Casos para optimización térmica

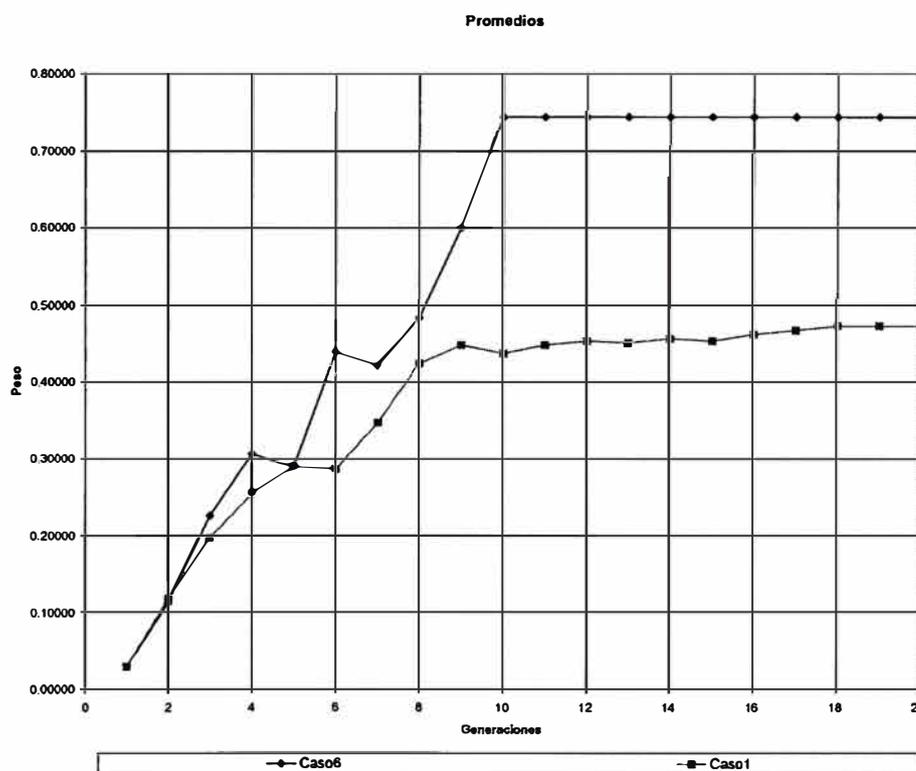
Se va a comentar por bloques de casos, para sacar nuestras conclusiones, al igual que se hizo para la optimización hidráulica.

Análisis de elitismo: Caso 0, caso 1, caso 2, caso 3, caso 4 y caso 5

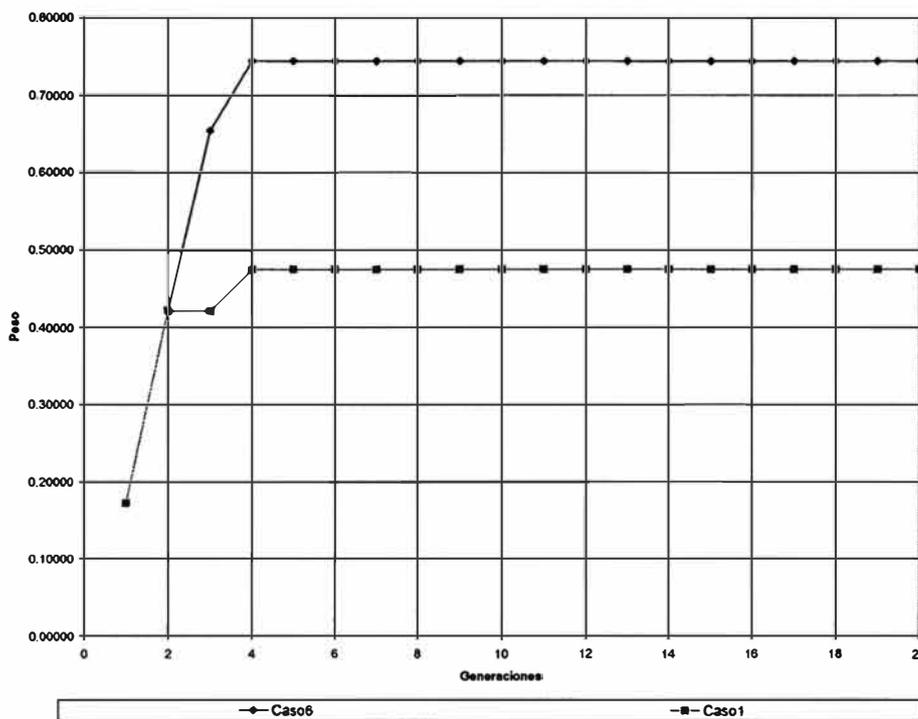


Se observa que el mejor resultado se logra en el caso 5, con la copia del mejor y escalamiento lineal, el segundo mejor es el caso 1. Se ve que el promedio en el caso 5 se iguala al máximo, es decir, la población tiene un único peso. Entonces, se toma los casos base, 5 y 1.

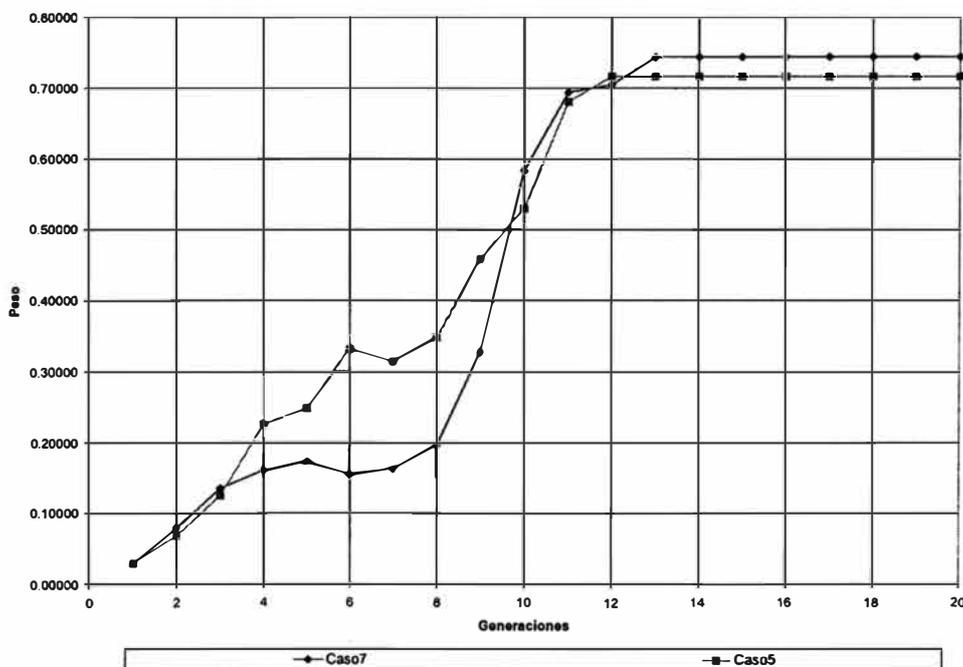
Análisis de probabilidad de cruce: Caso 2, caso 5, caso 6 y caso 7

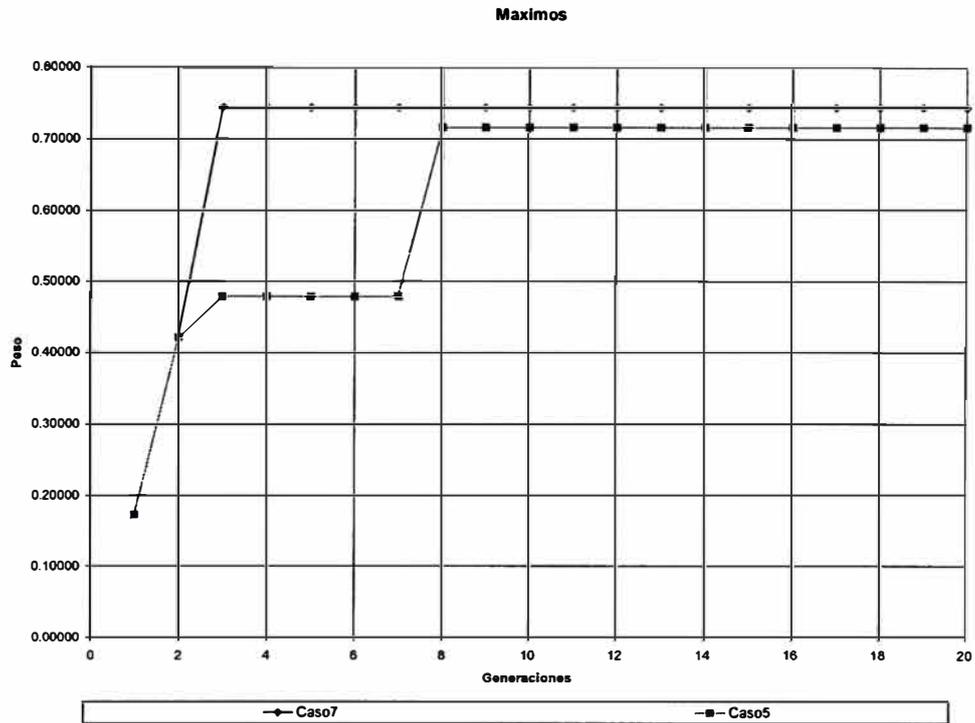


Maximos



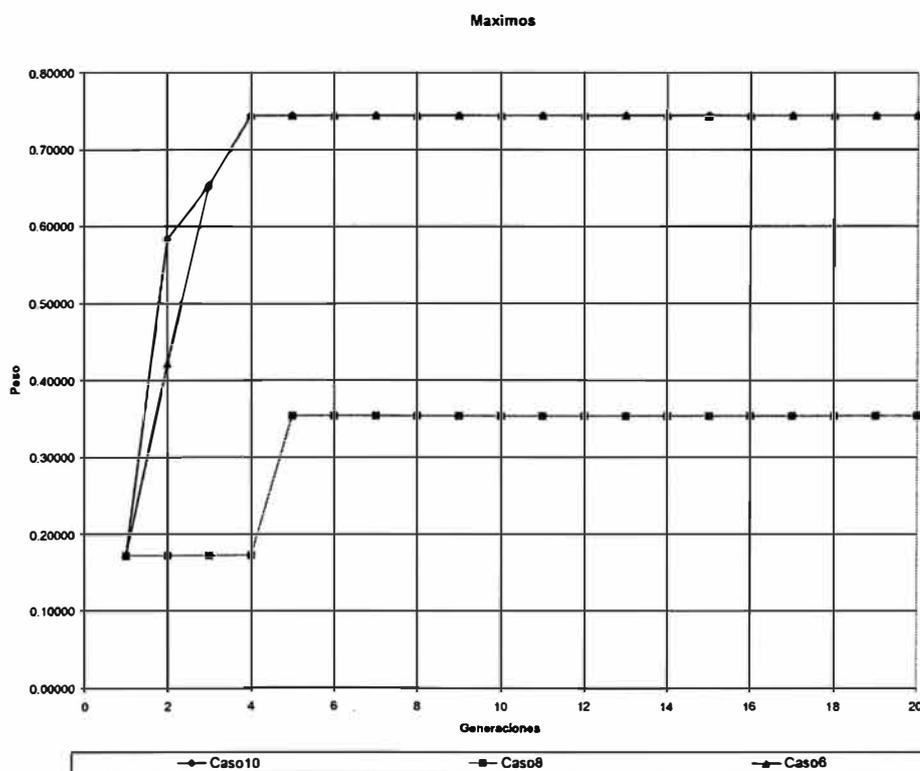
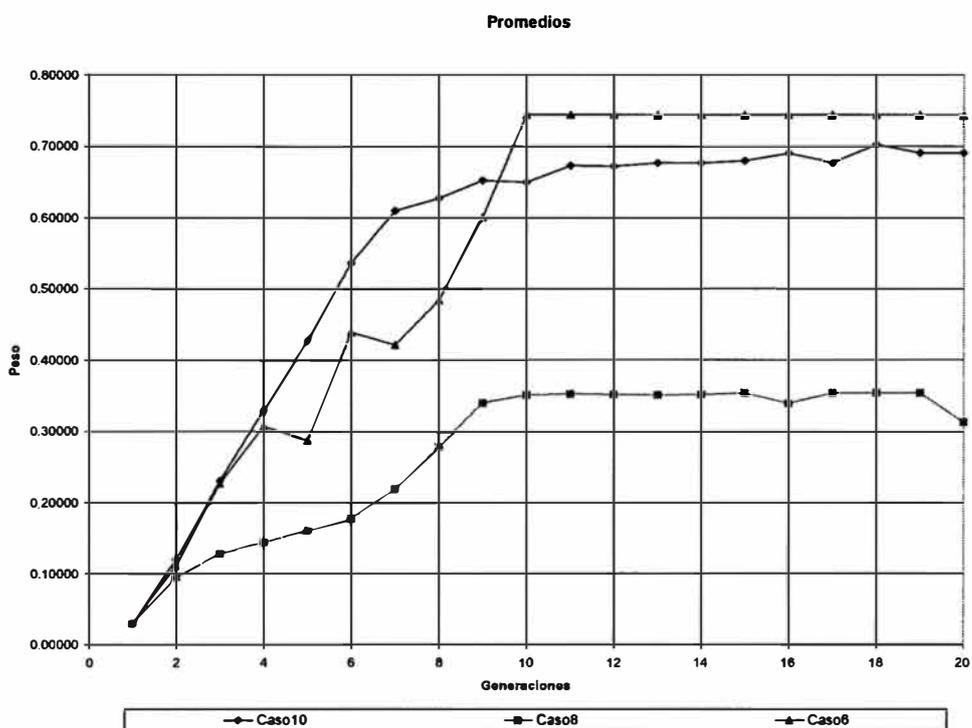
Promedios

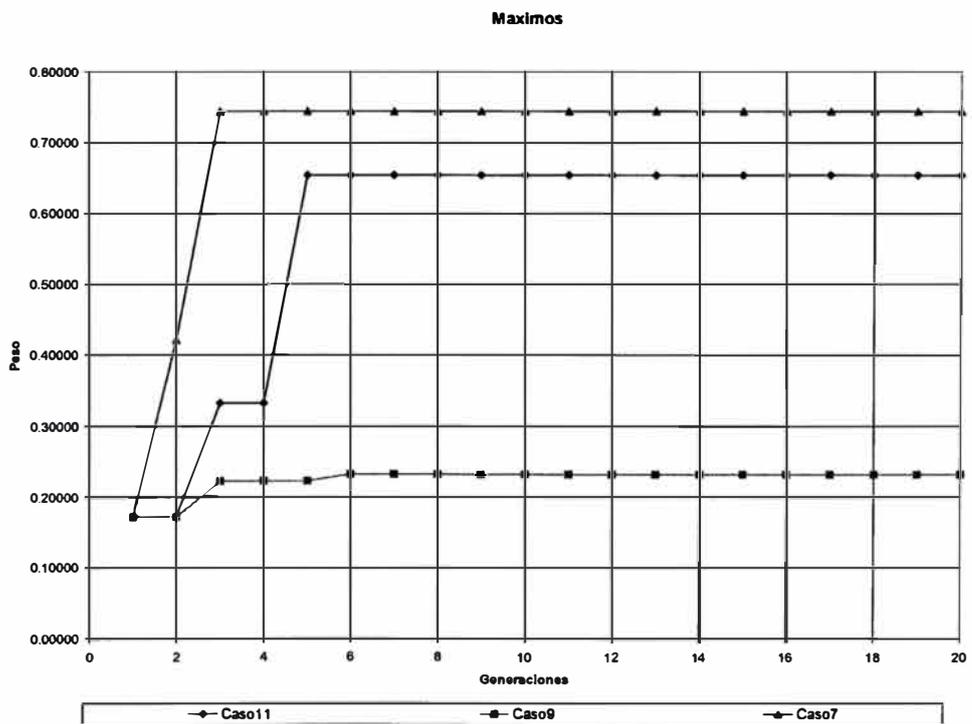
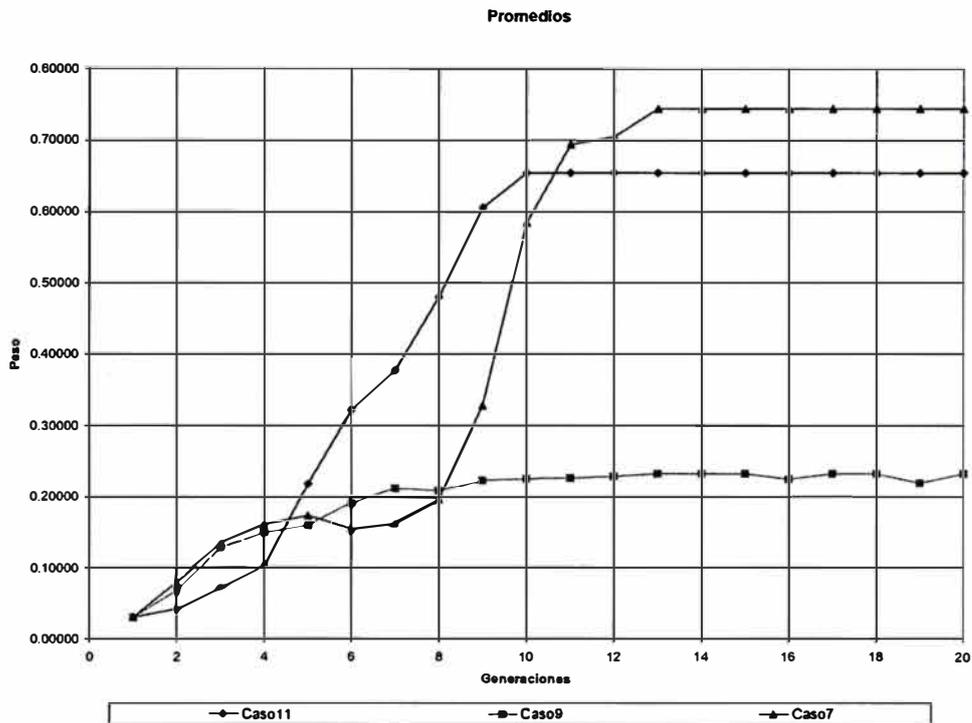




Se observa que tanto con la probabilidad de cruce de 0.75(casos 5 y 1) no se alcanza el óptimo que se alcanza con la probabilidad de cruce de 0.9. Por otro lado se ve que el promedio del caso 6 iguala a su máximo, igualmente ocurre en el caso 7, lo que significa que aumenta la velocidad de convergencia. Entonces se escoge nuevos casos base, 6 y 7.

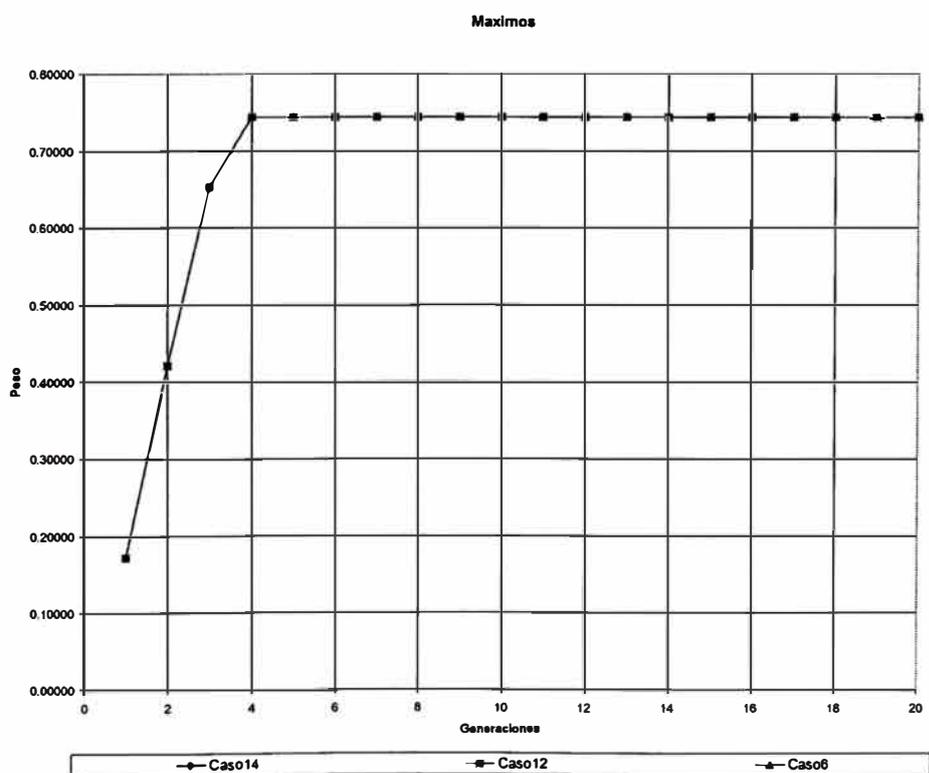
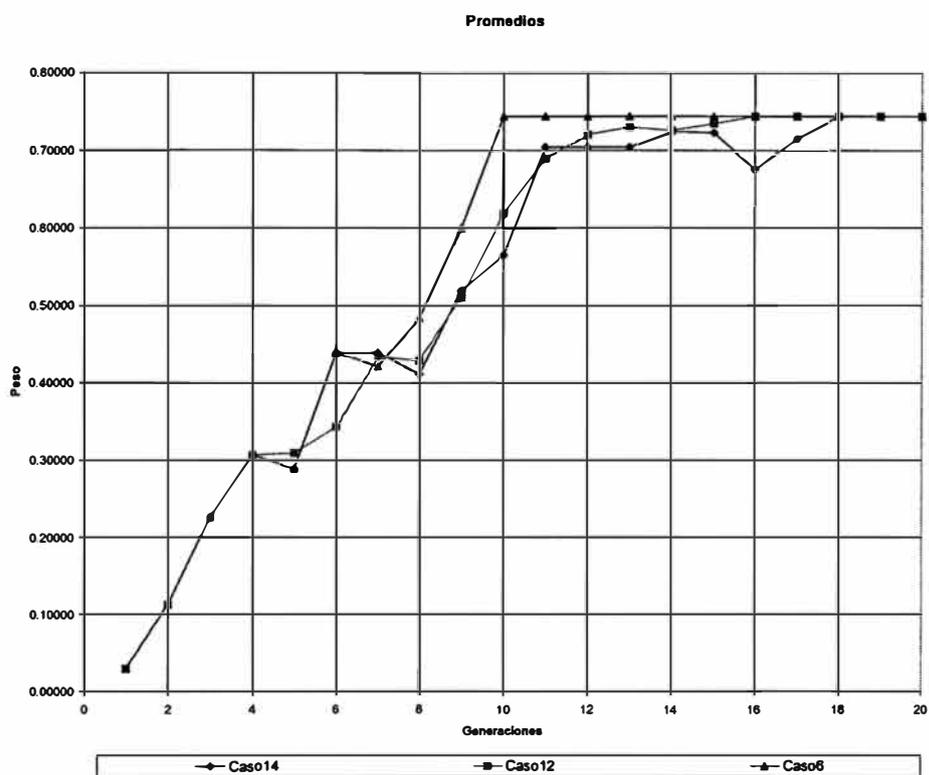
Análisis de los puntos de cruce: Caso 6, caso 7, caso 8, caso 9, caso 10 y caso 11

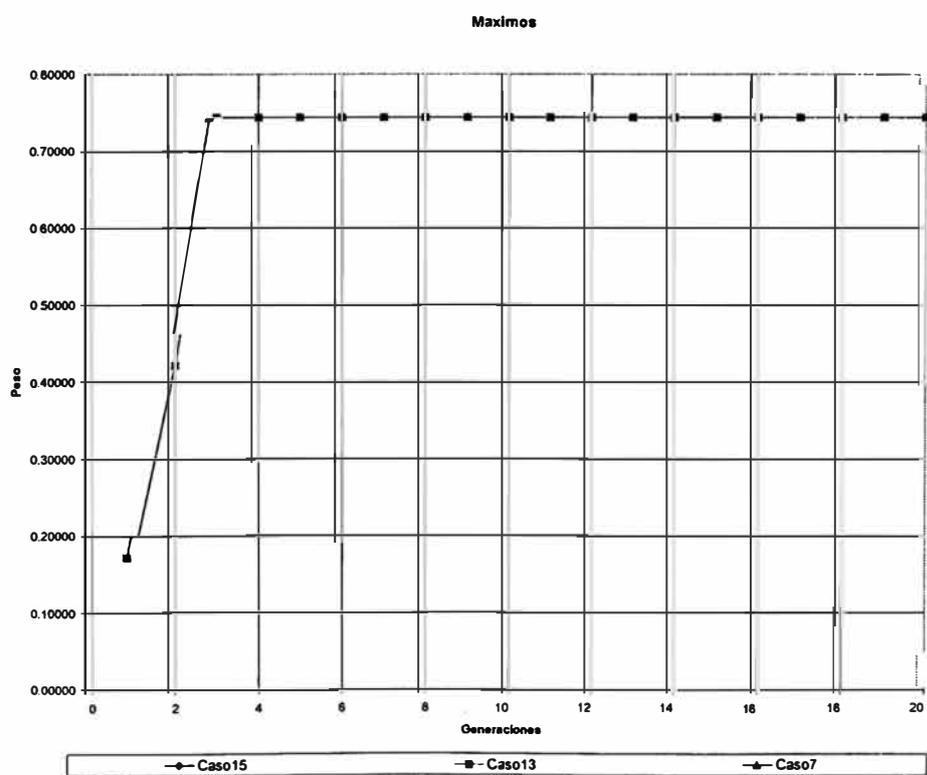
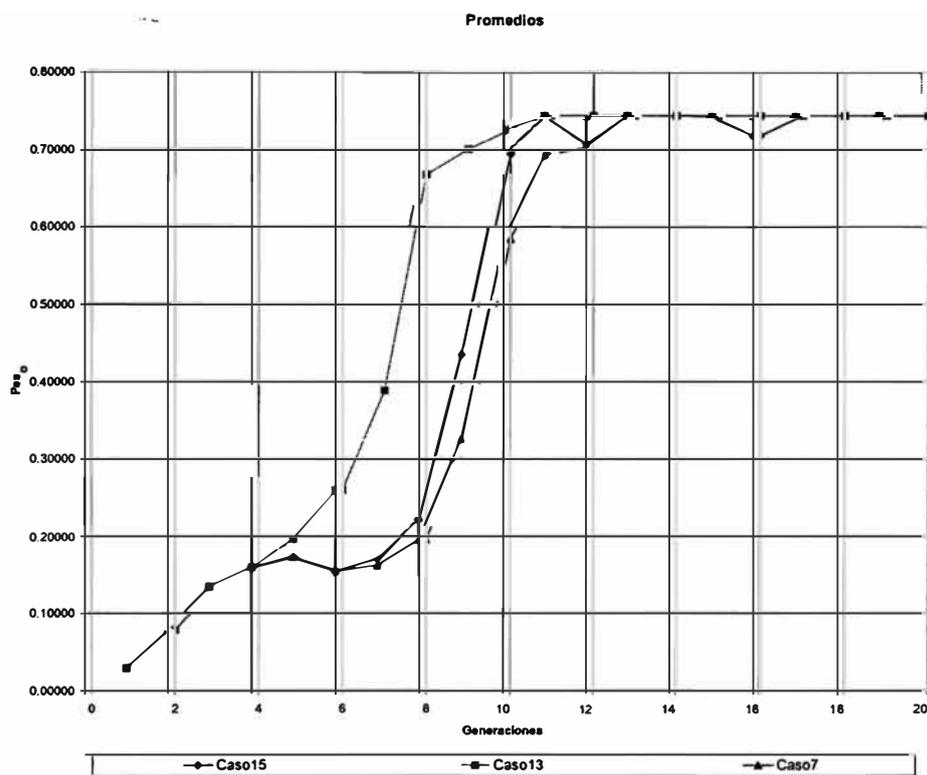




Se observa que con 1 punto de cruce se obtienen resultados malos, con 2 puntos de cruce se obtienen mejores individuos, a pesar que en el caso 10 (3 puntos de cruce), no hay mucha diferencia. Por la rapidez de convergencia se seleccionan dos puntos de cruce.

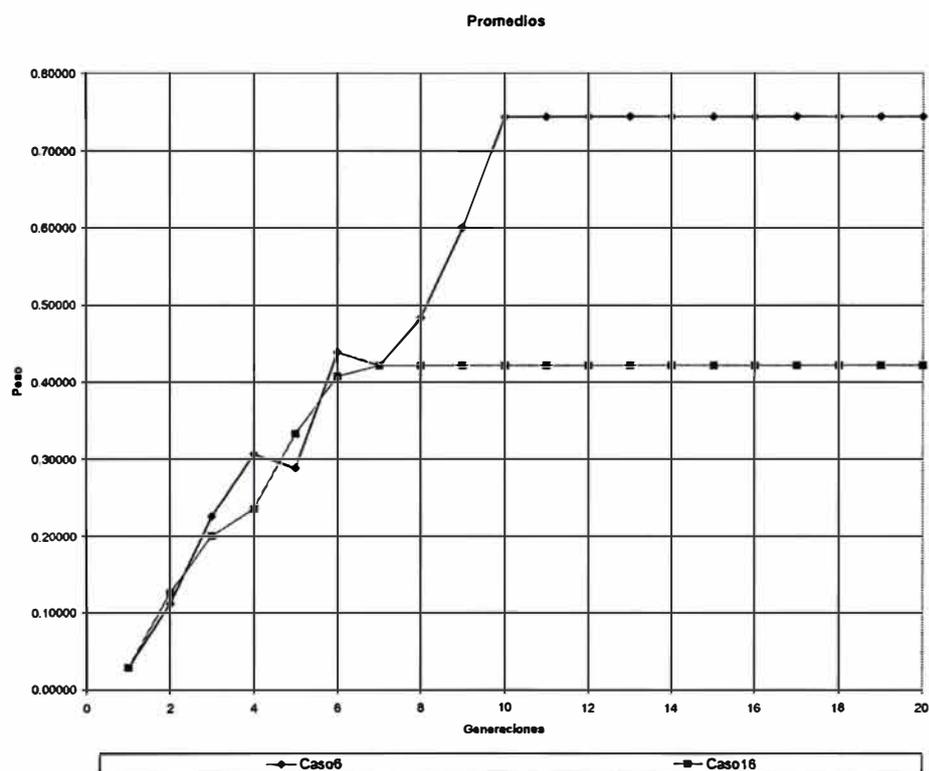
Análisis de la probabilidad de mutación: Caso 6, caso 7, caso 12, caso 13, caso 14 y caso 15



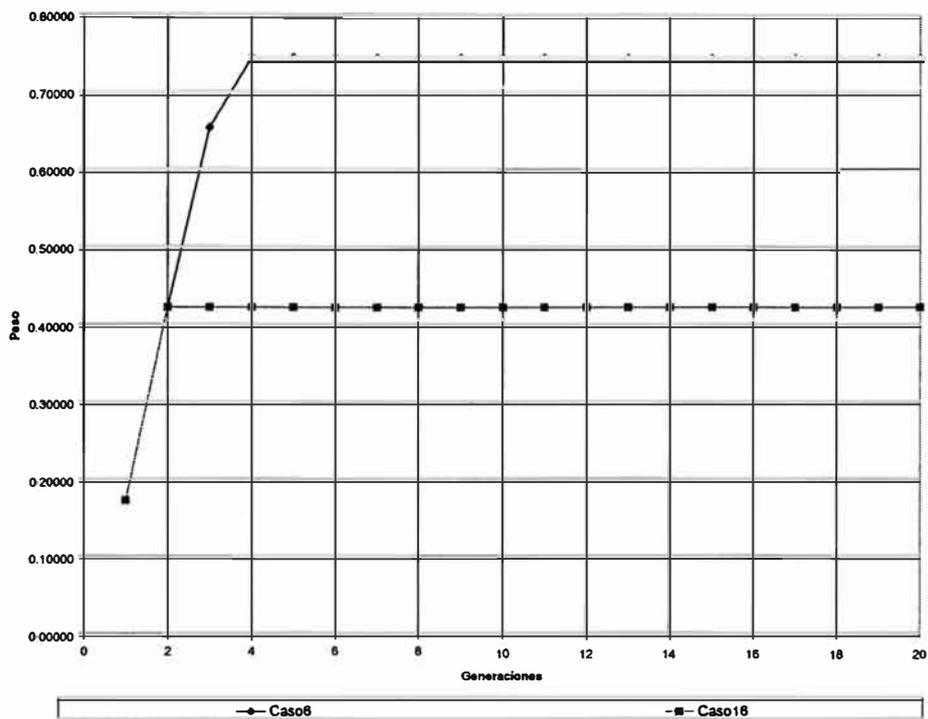


Al igual que con las hidráulicas se ve que la probabilidad de mutación actúa básicamente en el promedio de la población y no tiene efecto sobre los máximos, al final la población iguala su promedio con su máximo, de esta manera se aprecia que no tiene mucho sentido cambiar la probabilidad de mutación, entonces nos quedamos con los mismos casos base.

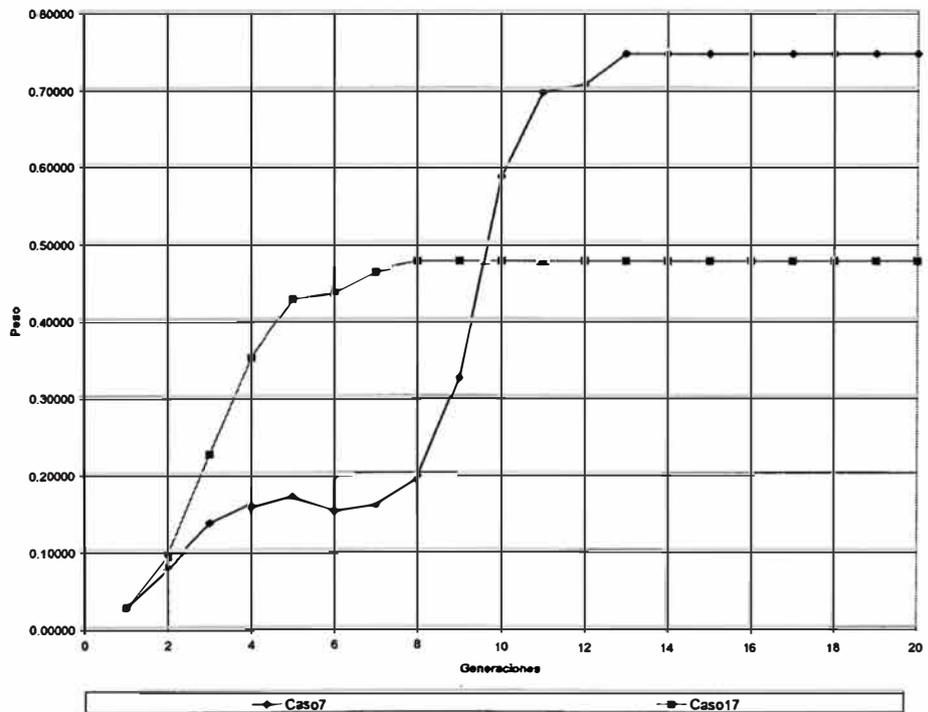
Análisis del reemplazo: Caso 6, caso 7, caso 16 y caso 17

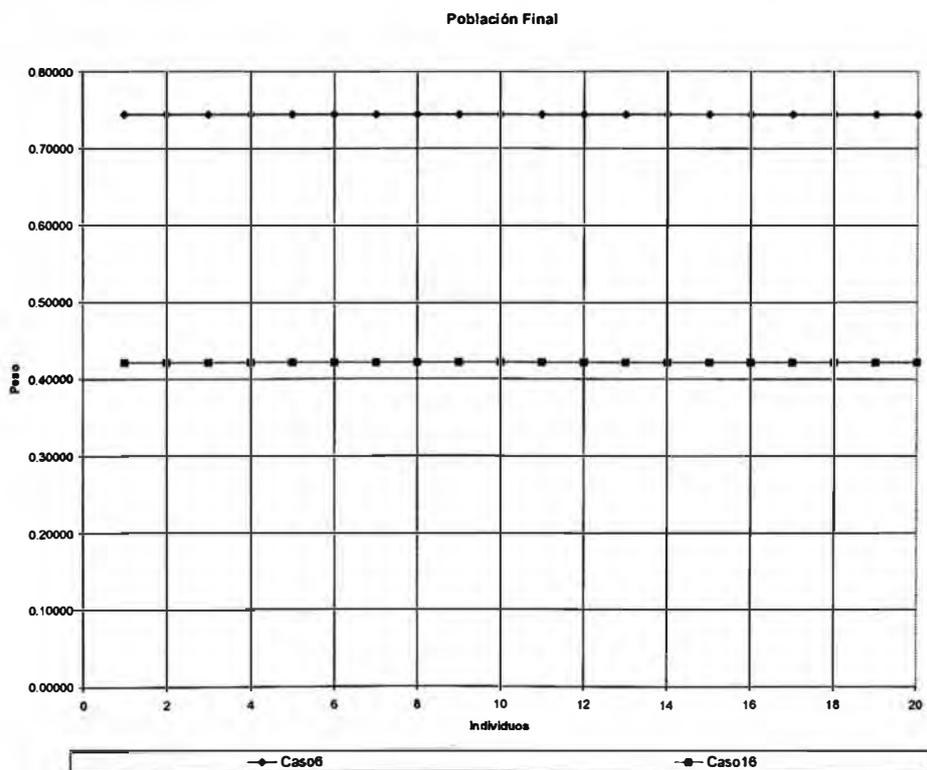
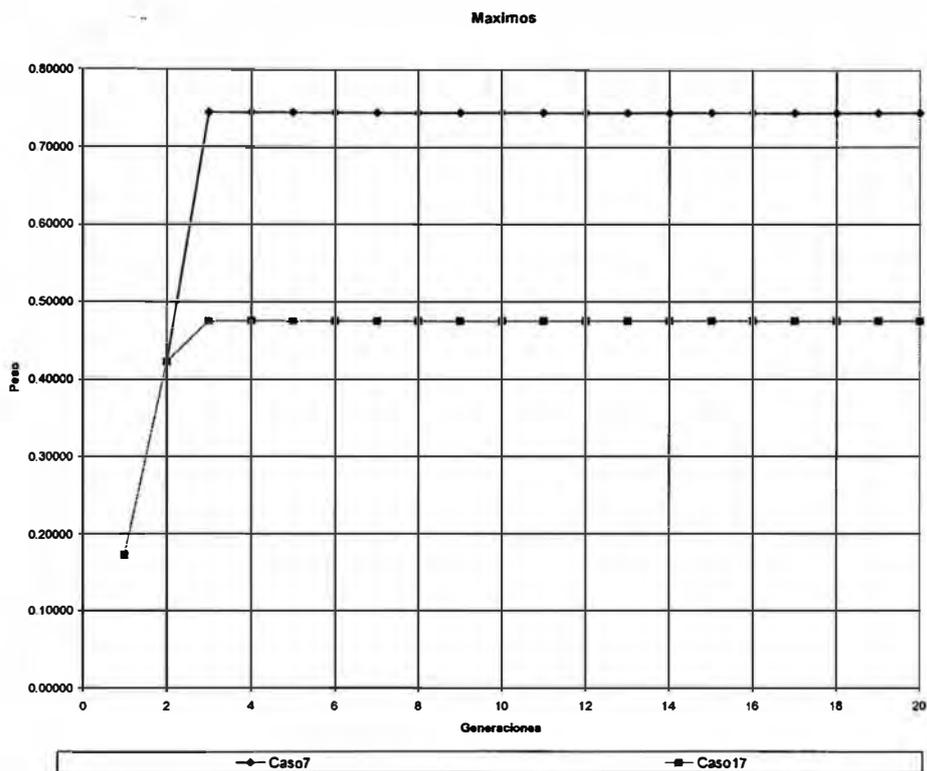


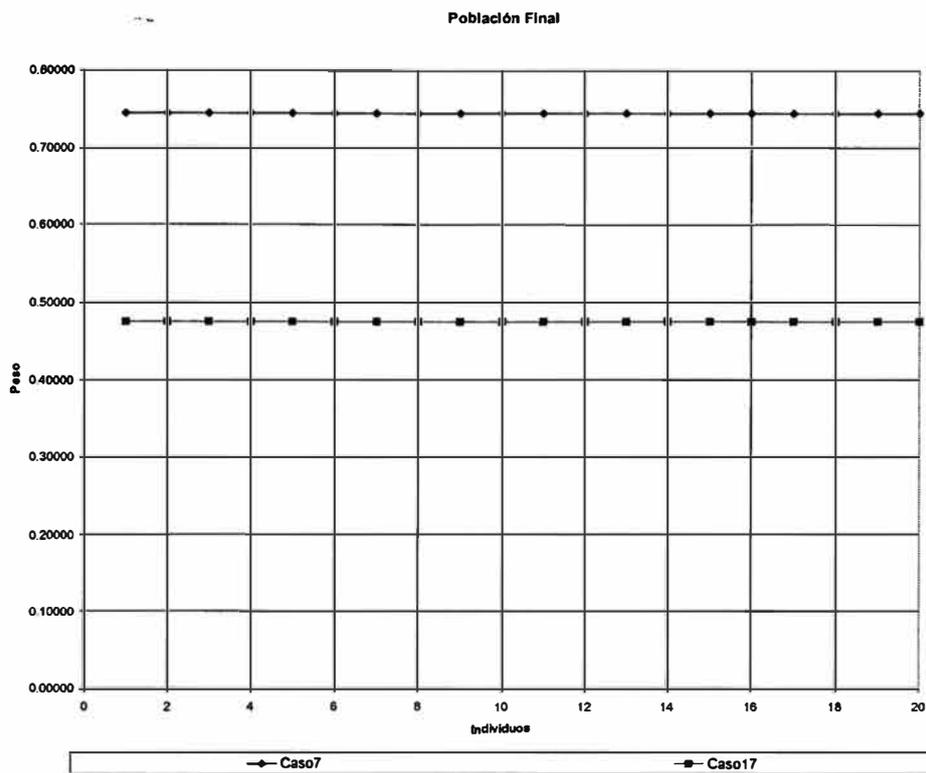
Maximos



Promedios

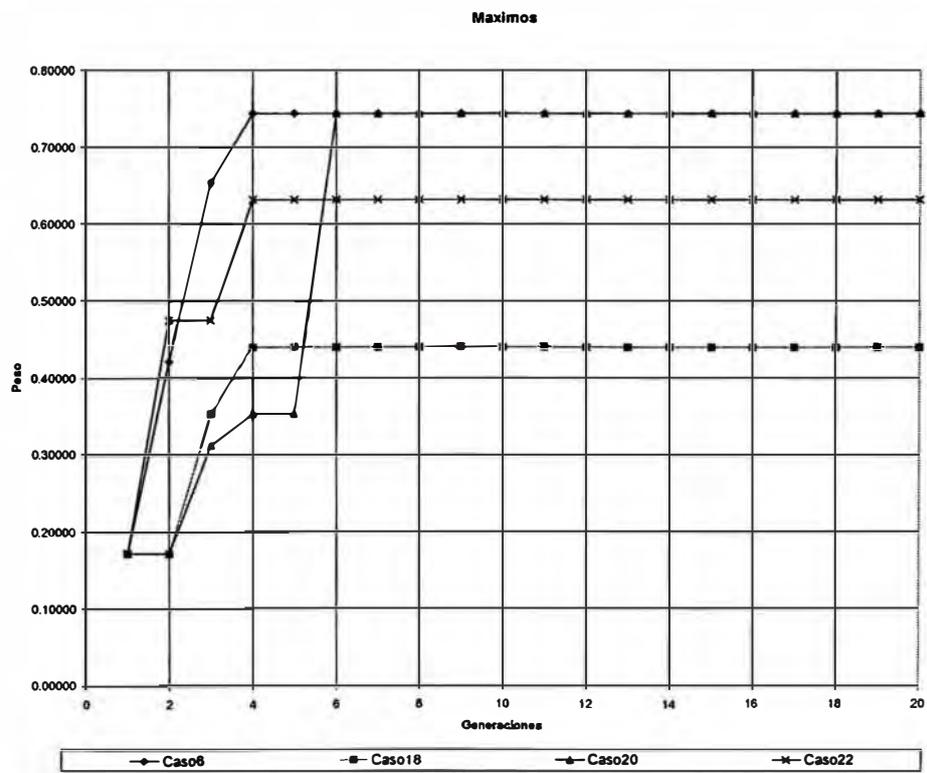
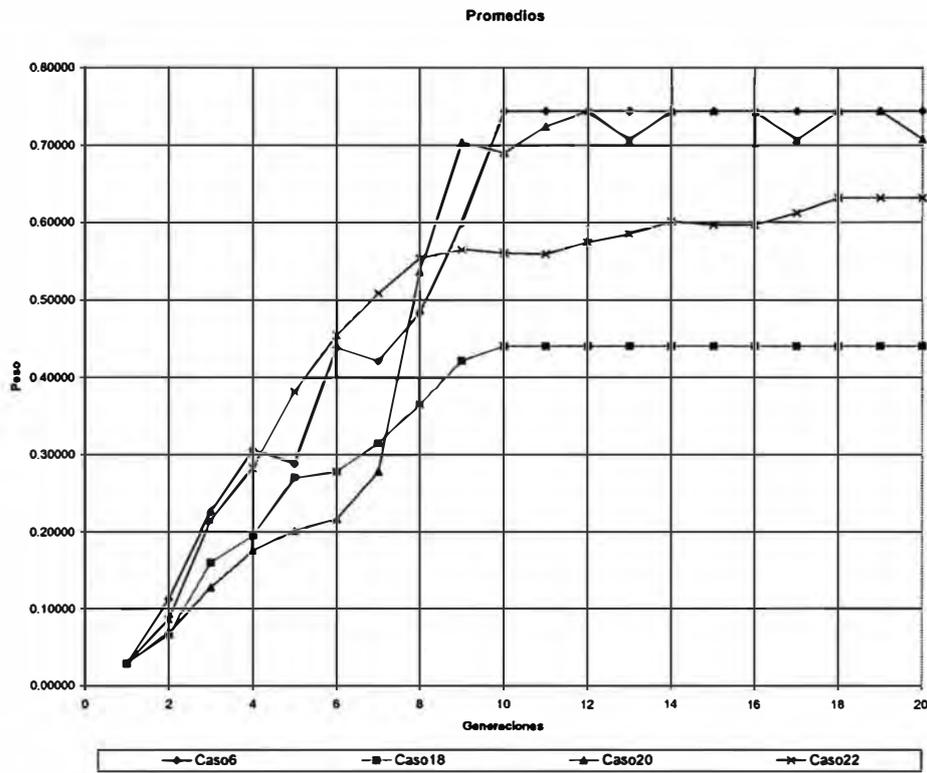


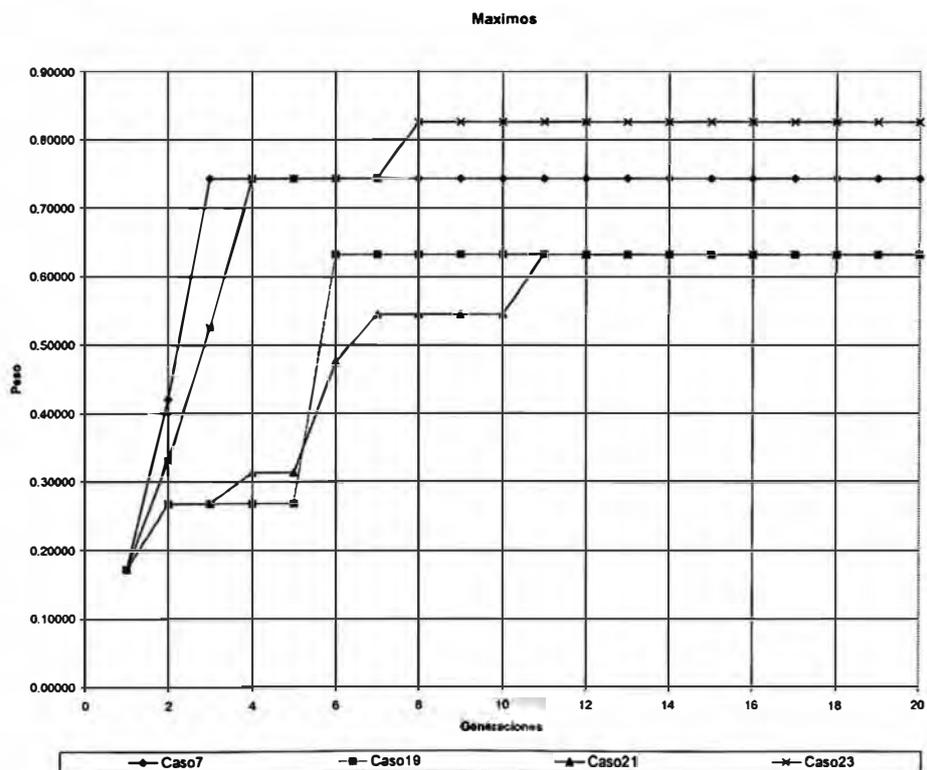
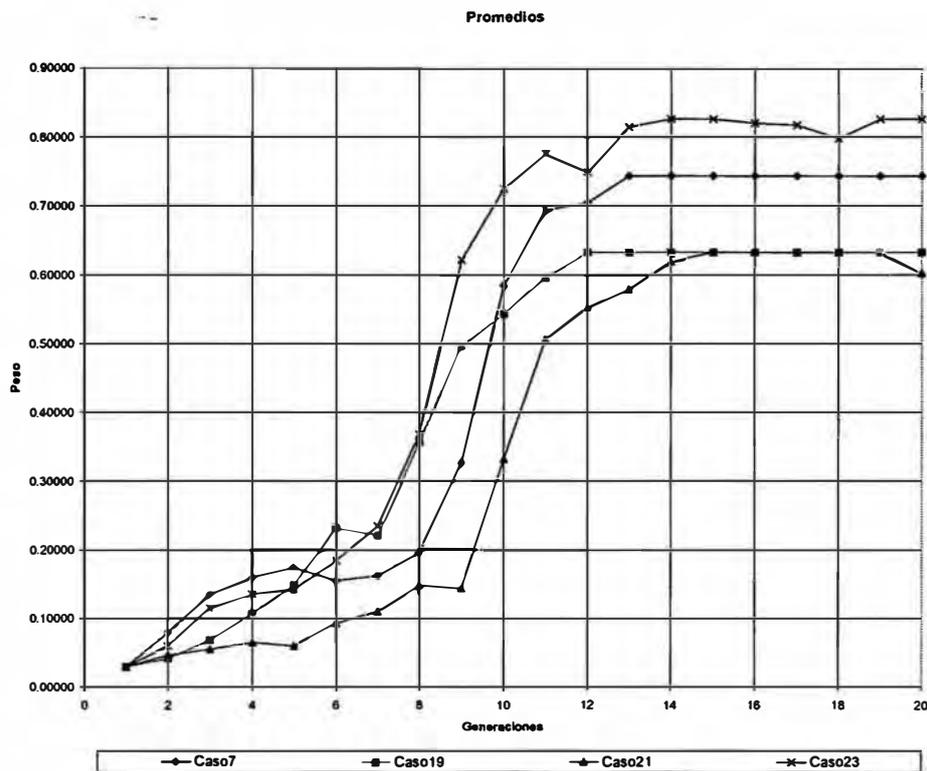




Se observa que al cambiar el tipo de reemplazo, el máximo rápidamente se detiene en un valor, por lo tanto, se continúa con los mismos casos base.

Análisis del cruce: Caso 6, caso 7, caso 18, caso 19, caso 20, caso 21, caso 22 y caso 23

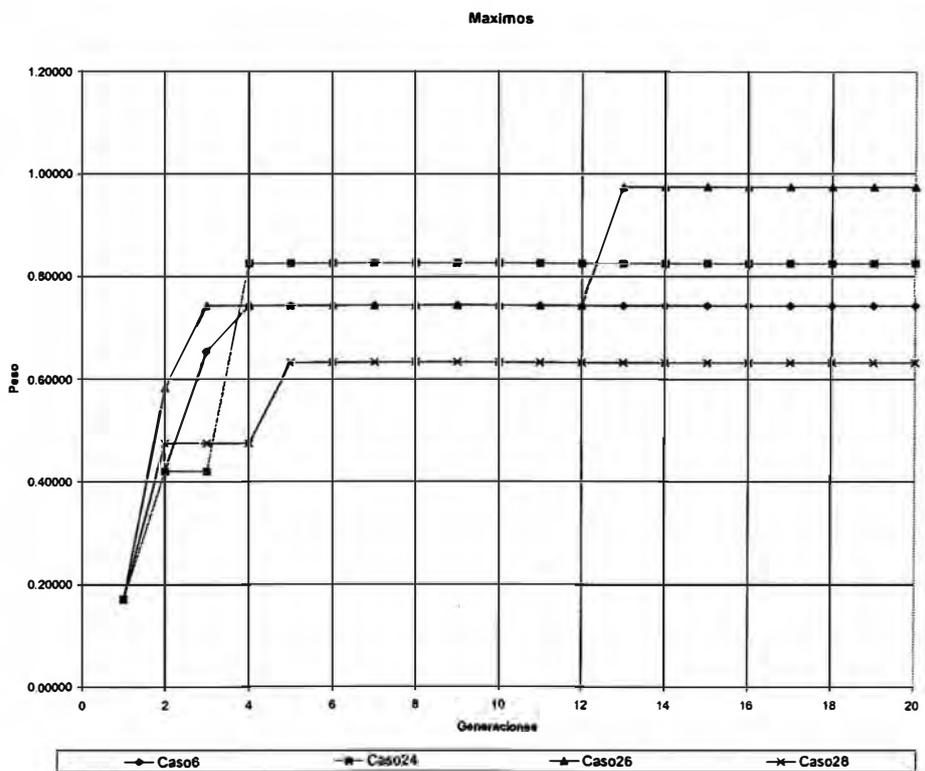
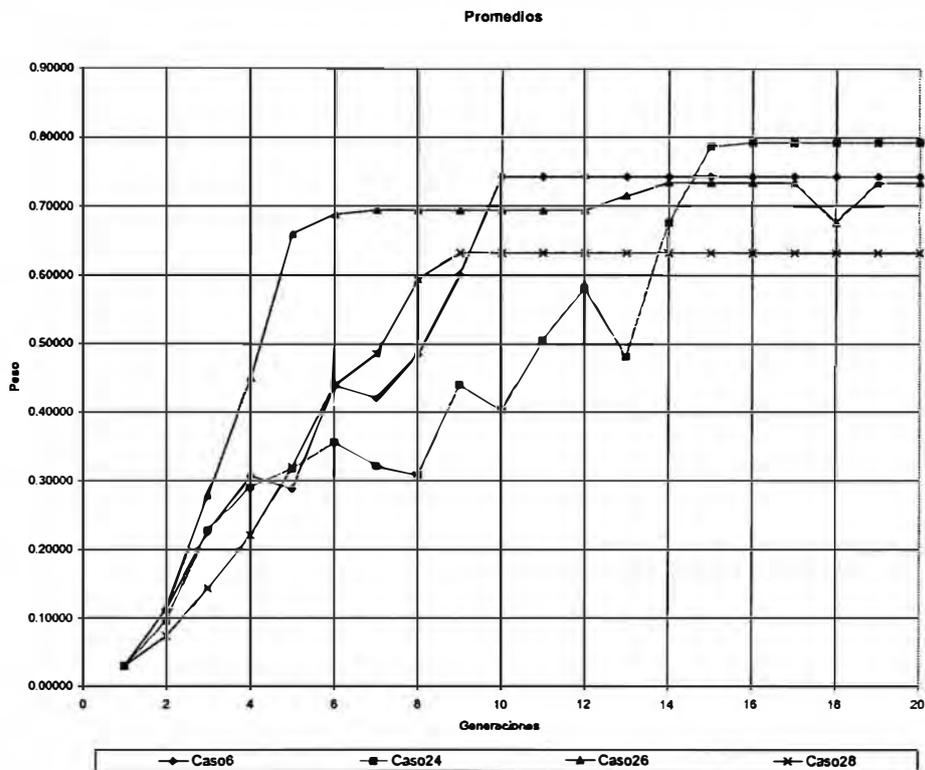




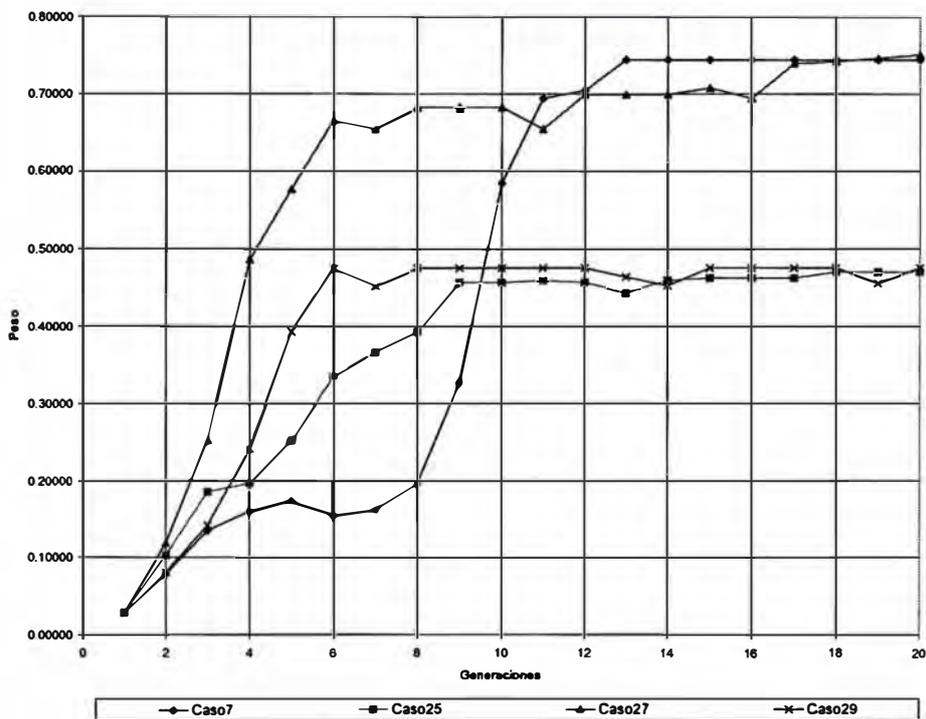
Se observa que al cambiar el tipo de cruce en el caso 6, no se logra mejores individuos, pero la convergencia es más rápida con el cruce uniforme. Al cambiar el tipo de cruce en el caso 7,

se obtienen un mejor individuo con el uniforme modificado.

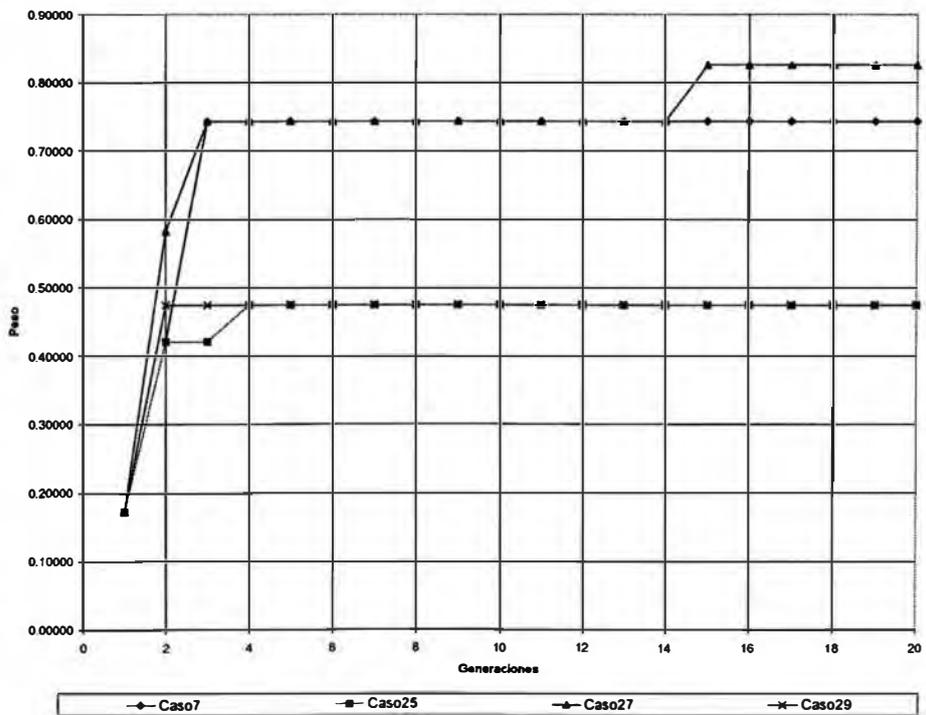
Análisis de la selección: Caso 6, caso 7, caso 24, caso 25, caso 26, caso 27, caso 28 y caso 29



Promedios

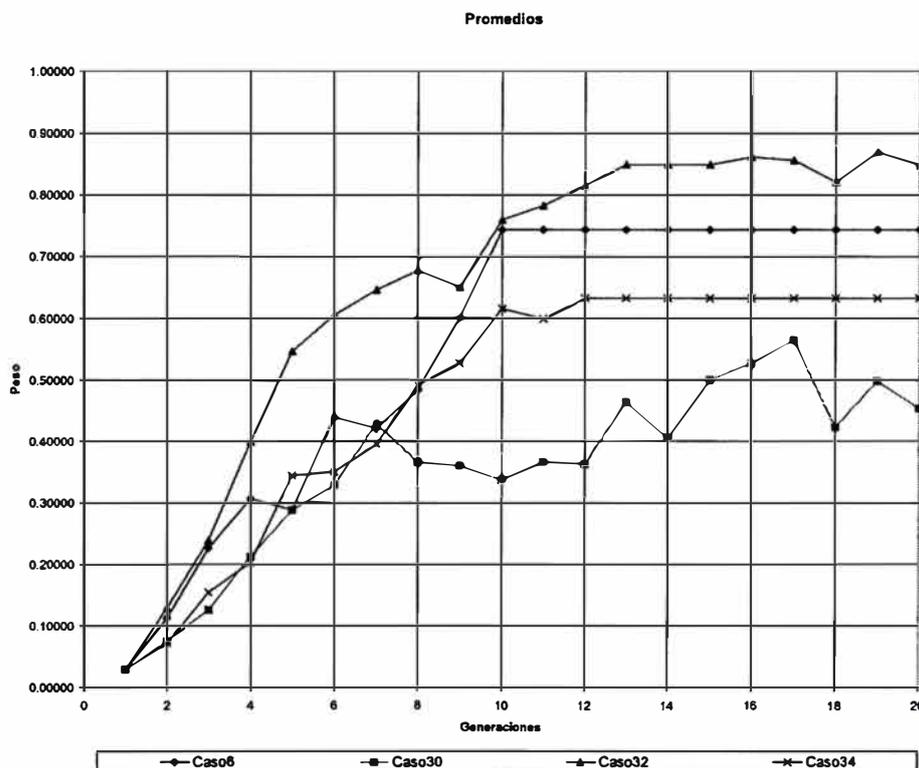


Maximos

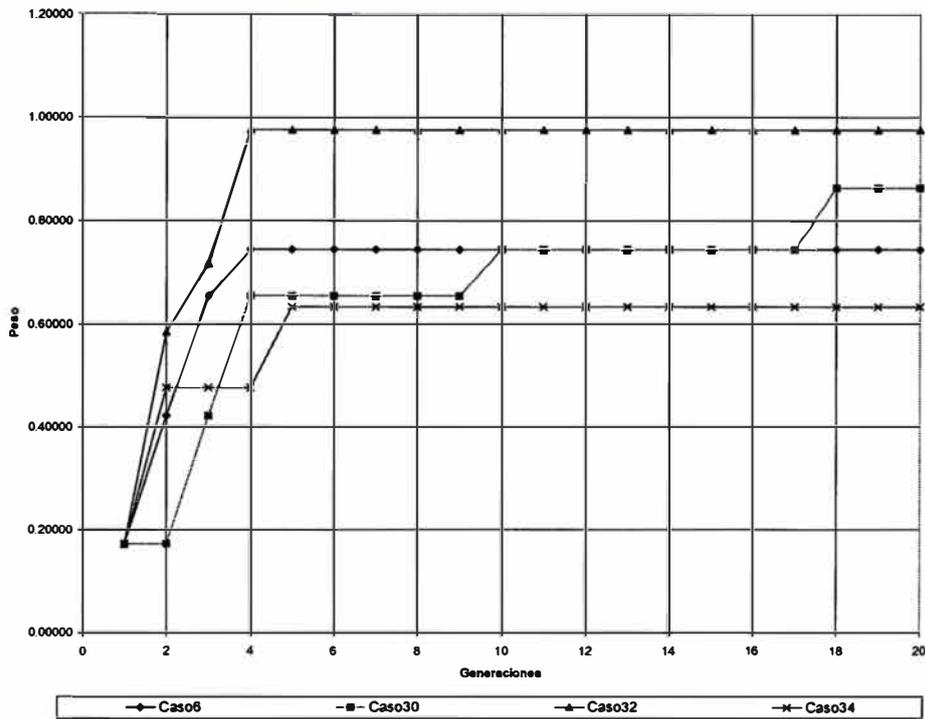


Se observa que el mejor tipo de selección es la estocástica universal, al cambiar el caso 7, prácticamente se logra el peso máximo, es decir, un valor cercano a uno. Se analizará el niching (compartición y apiñamiento) para ver si existe otro máximo.

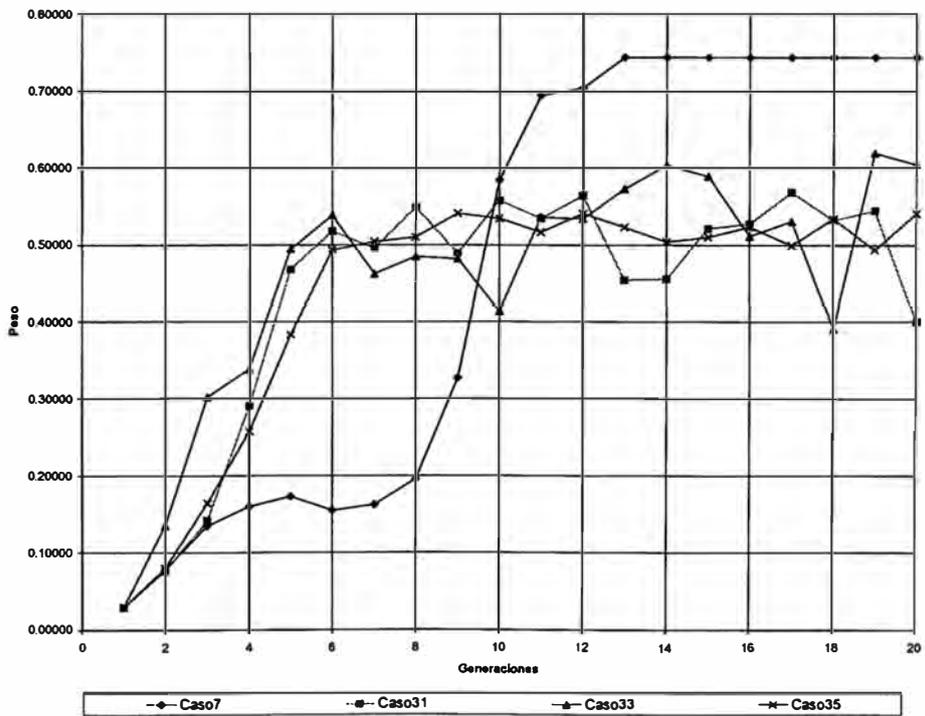
Análisis de la compartición: Caso 6, caso 7, caso 30, caso 31, caso 32, caso 33, caso 34 y caso 35

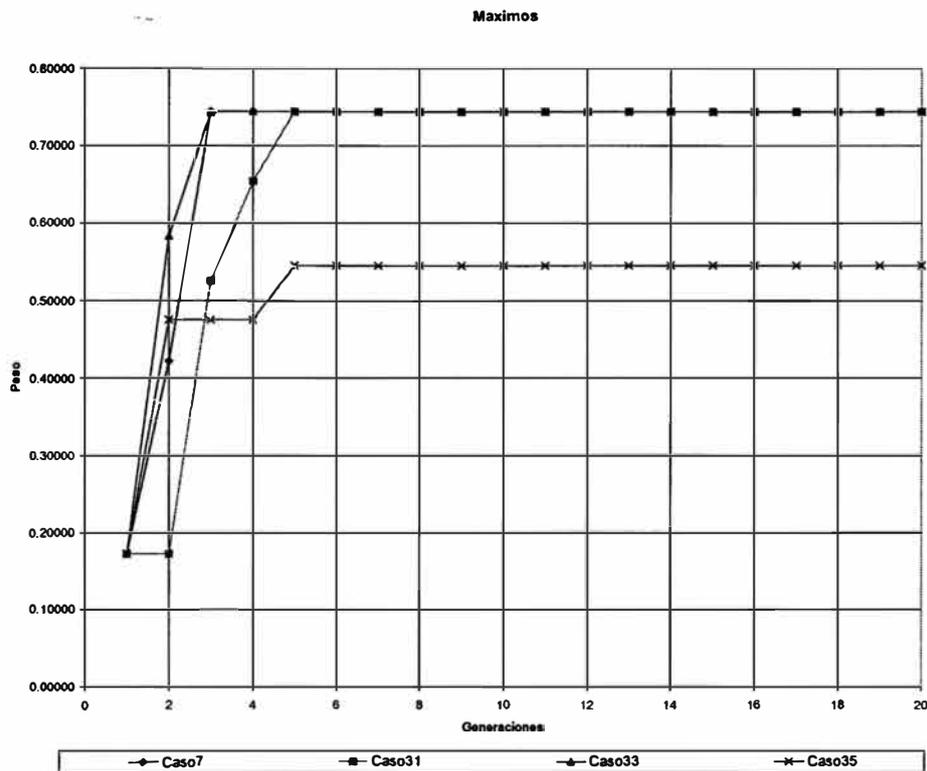


Maximos



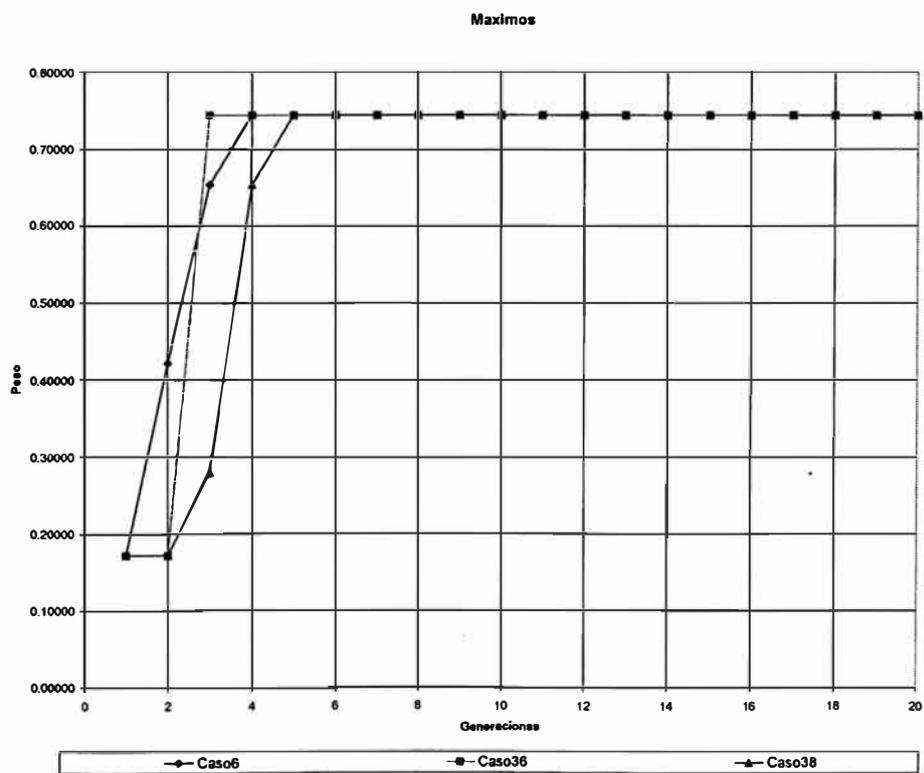
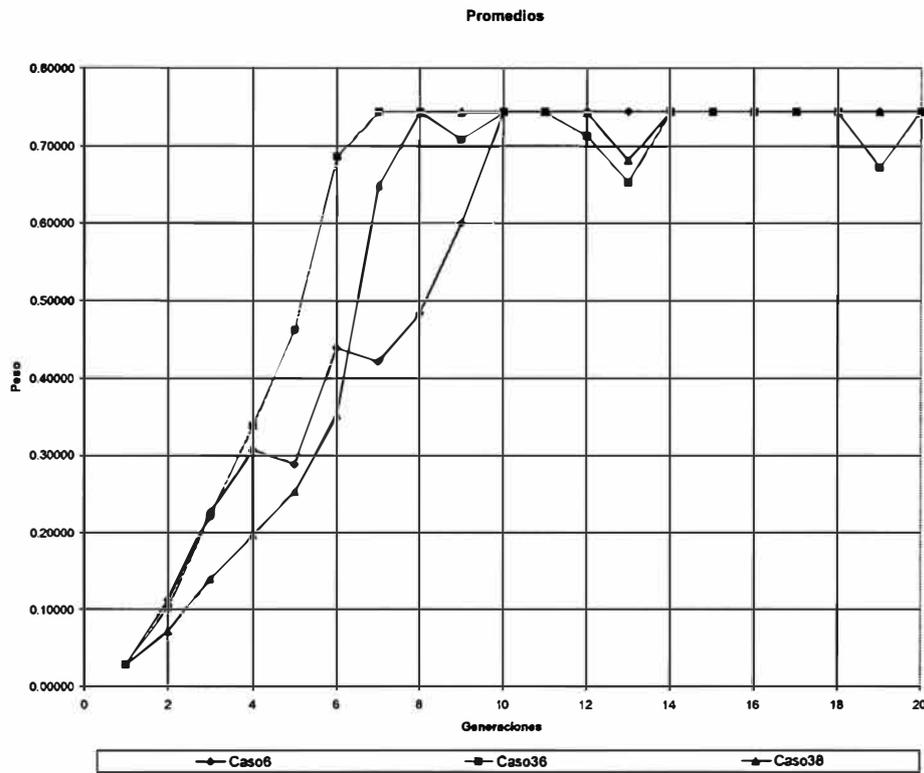
Promedios



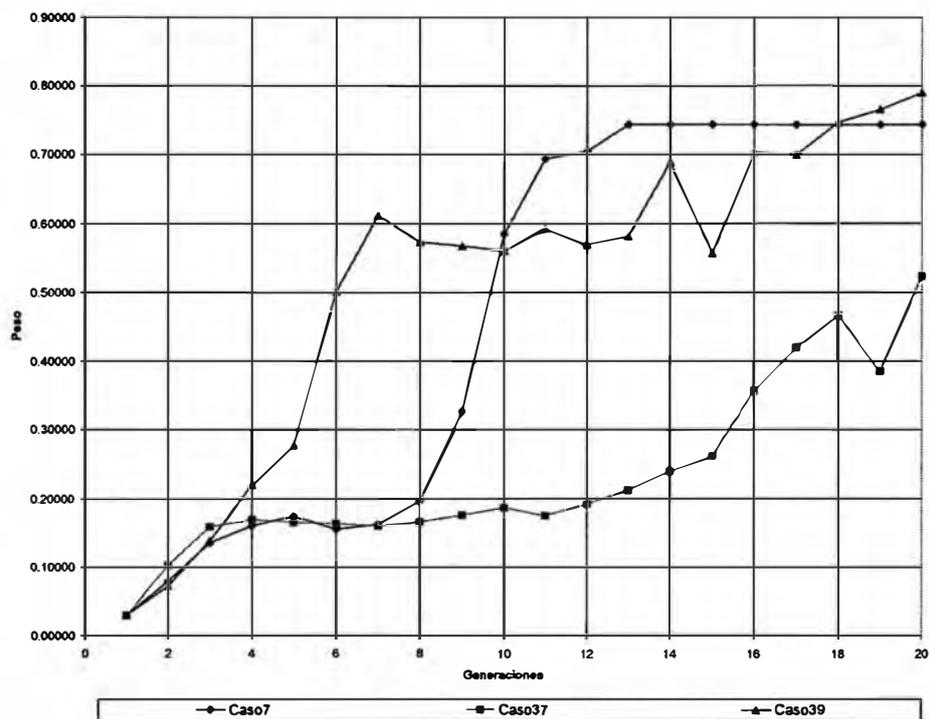


Al agregar la compartición, se logra que en el caso 6 se obtenga pesos cercanos al máximo, a la inversa del caso anterior donde no había compartición. Se ve también que se obtienen promedios altos, es decir, que el valor máximo obtenido probablemente sea el máximo que se pueda obtener mediante el algoritmo genético binario.

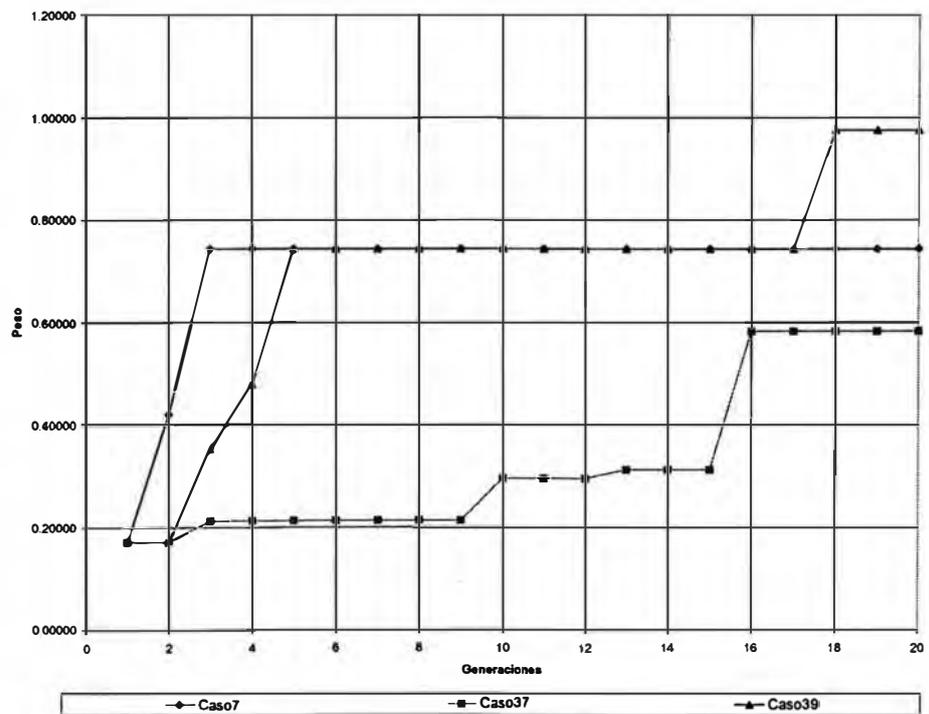
Análisis del apiñamiento: Caso 6, caso 7, caso 36, caso 37, caso 38 y caso 39



Promedios

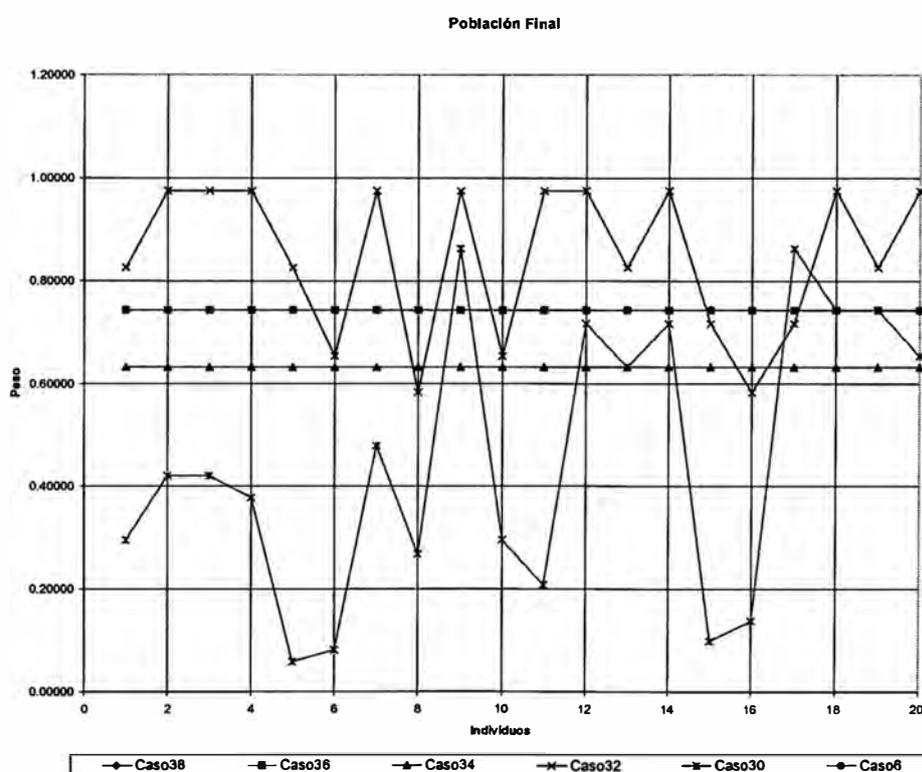


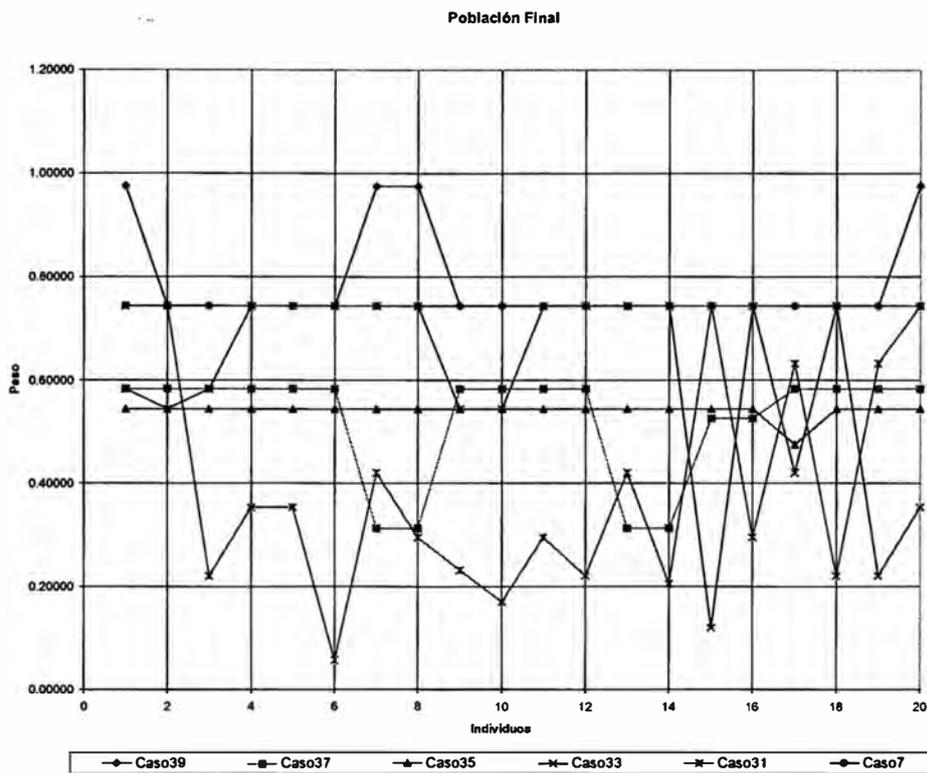
Máximos



Se ve que al aplicar el apiñamiento en el caso 6, lo que varía mayormente son los promedios, se tiene promedios similares para los tres casos. En cambio, en el caso 7, se ve un cambio de los máximos, se observa una diferencia más marcada con los promedios; también aquí se logra llegar al máximo al que se llegó anteriormente.

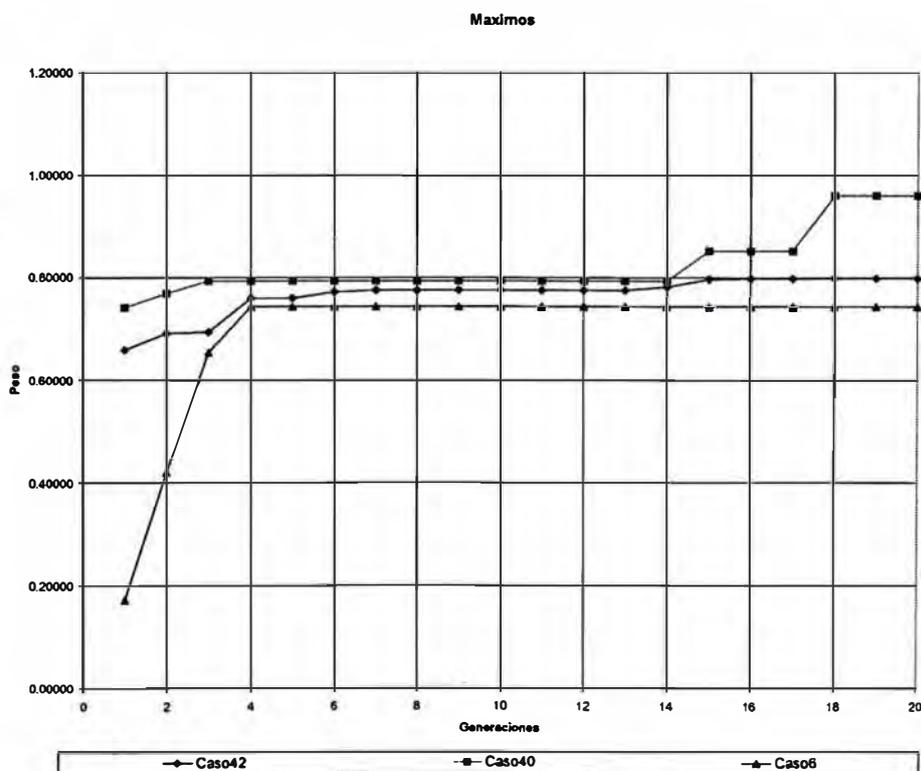
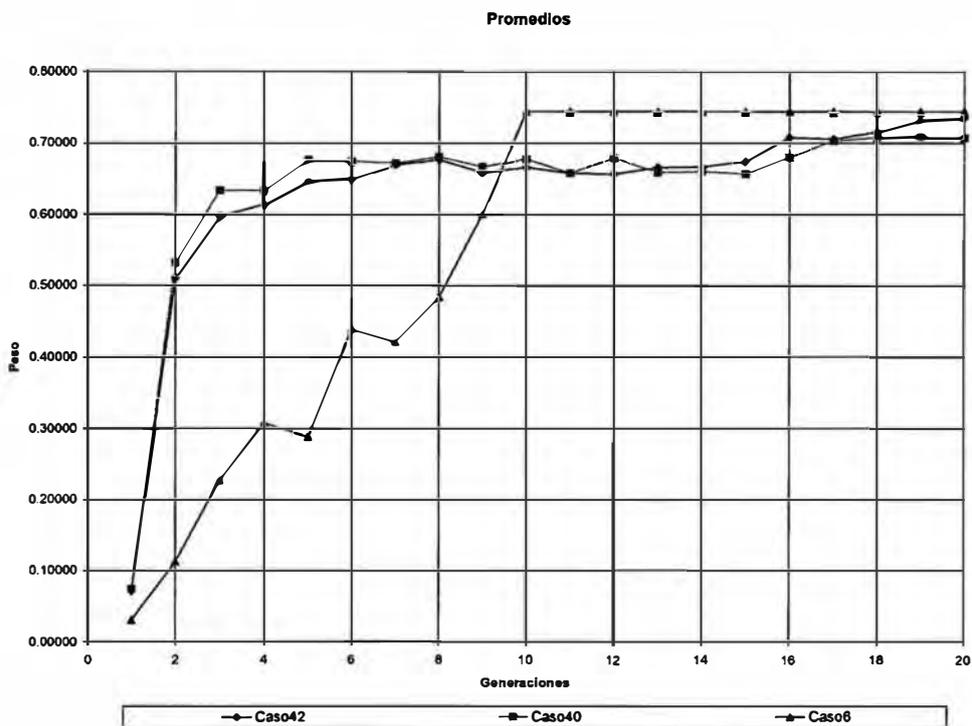
Análisis del niching(nicheo): Caso 10, caso 4, caso 30, caso 31, caso 32, caso 33, caso 34, caso 35, caso 36, caso 37, caso 38 y caso 39



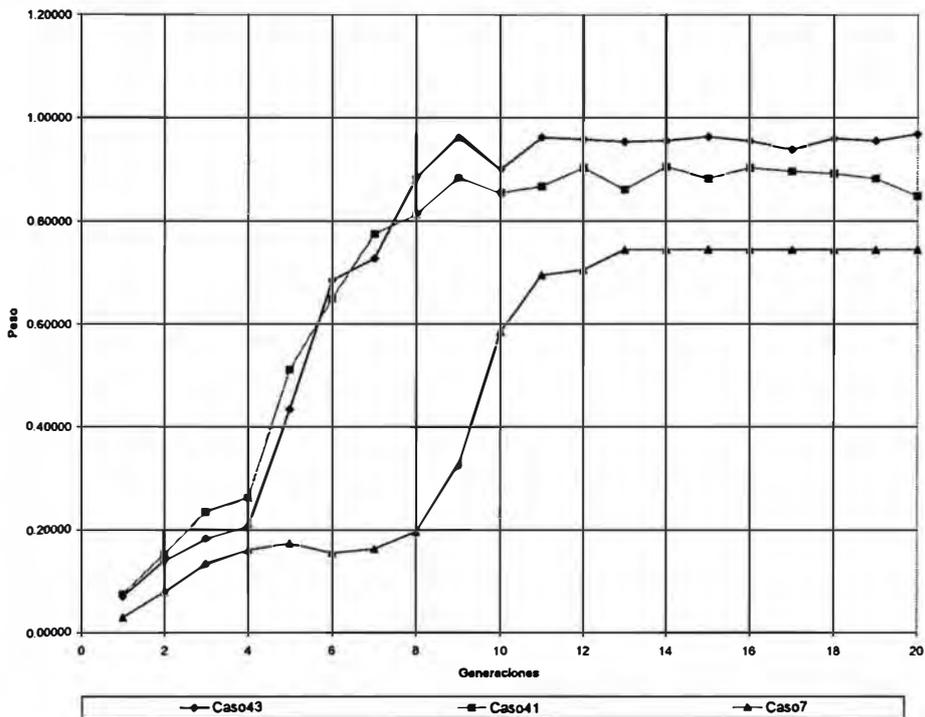


Se ve que los mejores resultados de diversificación se obtienen con la compartición, pero no se ve una subpoblación estable, esto quiere decir que sólo existe un máximo.

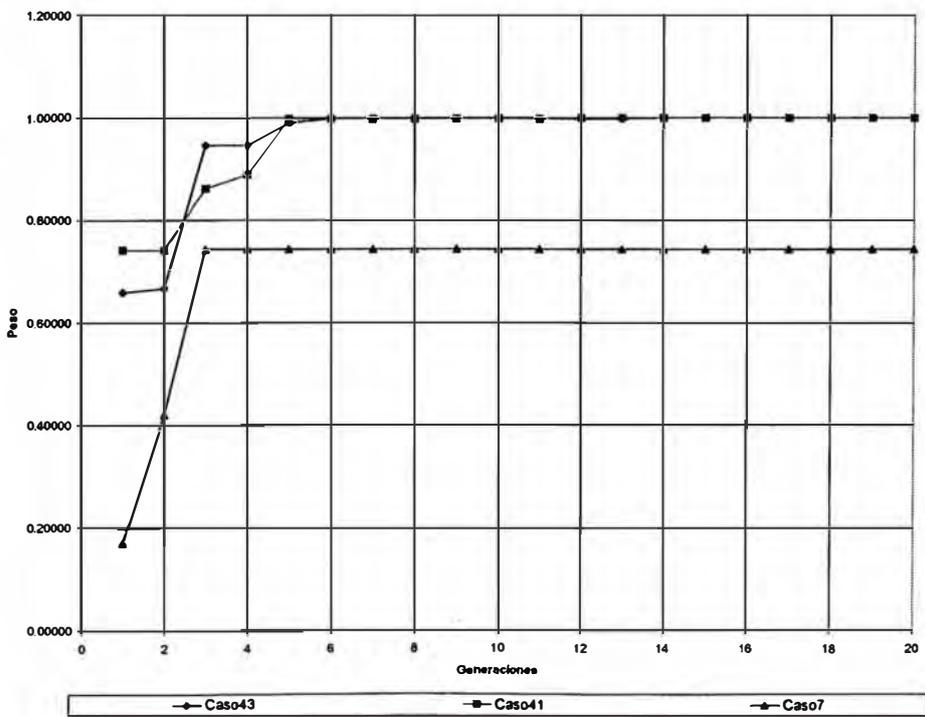
Análisis del ruido A: Caso 6, caso 7, caso 40, caso 41, caso 42 y caso 43



Promedios



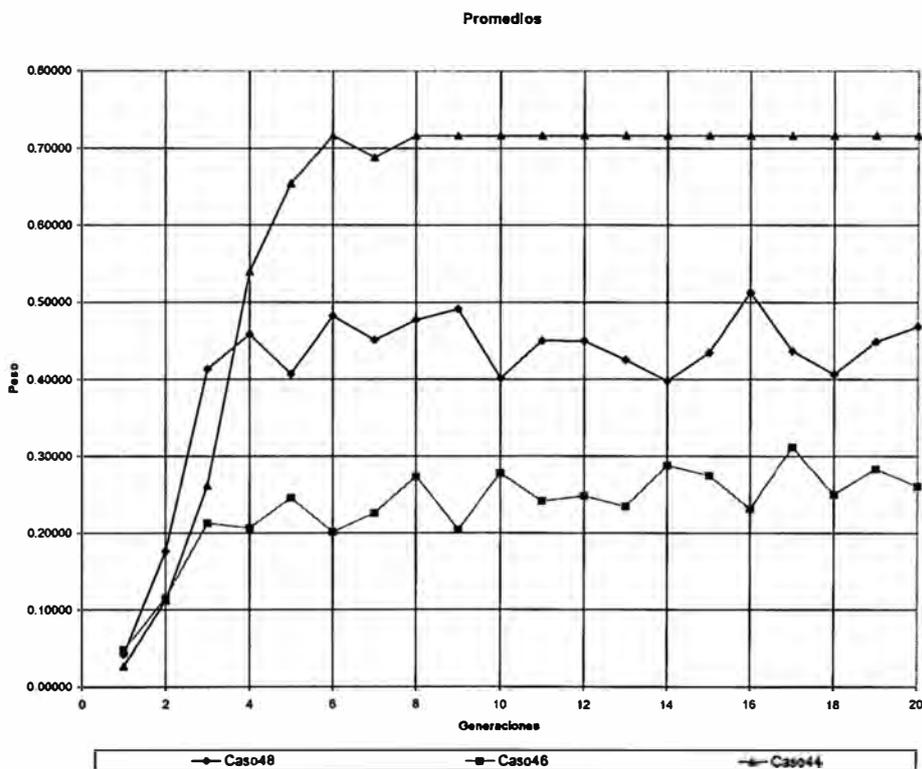
Maximos



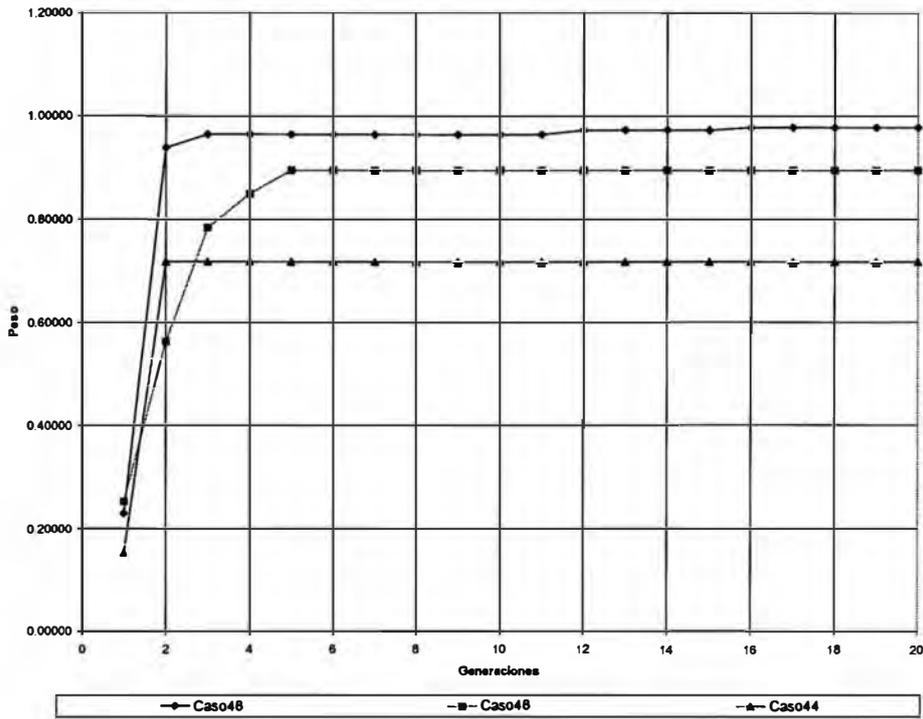
En este caso se ve que el ruido tienen un efecto muy positivo en la búsqueda de un mejor resultado, al aplicar al caso 7, se obtiene el máximo posible, es decir, se consigue un peso igual a uno. Para el caso 6, se obtiene resultados cercanos al máximo, en el caso 40 (Ruido tipo Gauss), se obtiene un valor muy cercano a uno.

Análisis del ruido B: Caso 6, caso 7, caso 44, caso 45, caso 46, caso 47, caso 48 y caso

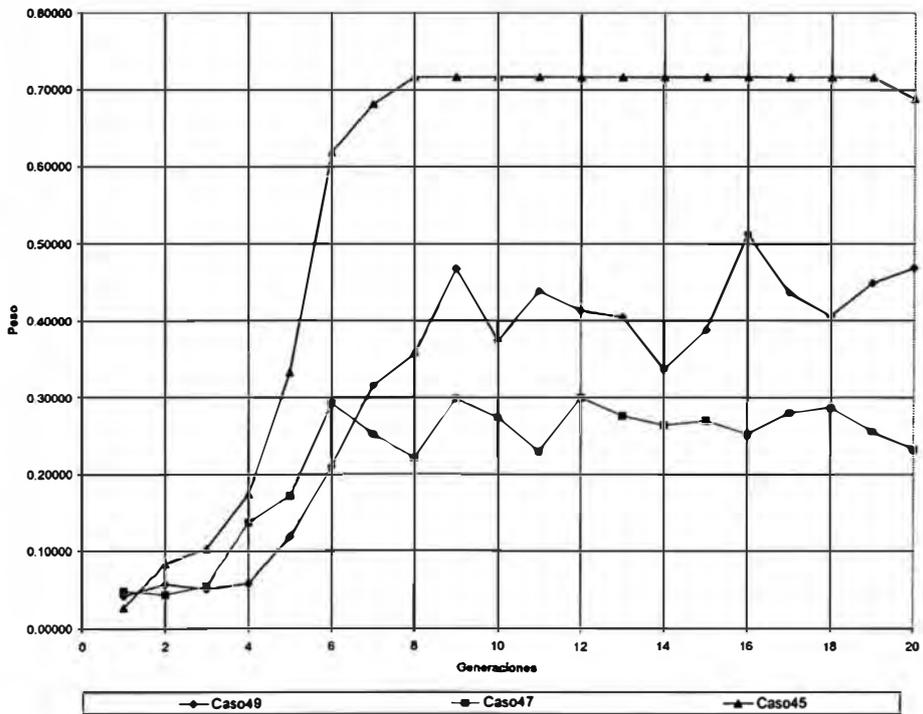
49

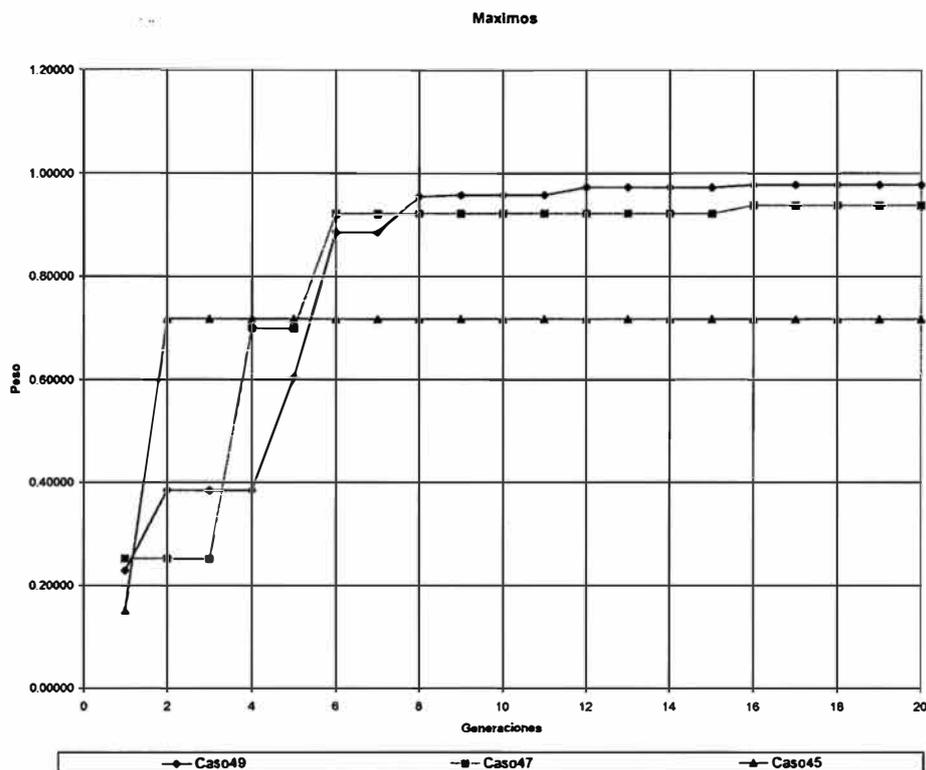


Maximos



Promedios





Se aprecia que aunque la longitud del cromosoma disminuye, se obtiene muy buenos resultados al aplicar el ruido, a diferencia del algoritmo usado para las hidráulicas. Así que el ruido será considerado, en la obtención final de resultados.

7.3 Análisis de resultados

En el acápite anterior se han analizado los operadores y sus tipos, y se ha establecido que para la optimización hidráulica se usará la secuencia del caso 10:

CASO	Longitud de cromosoma	The best	Preselección	Selección	Probabilidad de cruce	Número de Puntos	Tipo de Cruce
10	30	Copiar el mejor	Ninguna	Ruleta	0.9	3	De n puntos

Probabilidad de mutación	Tipo de Mutación	Tipo de Reemplazo	Tipo de niching	Prevención de Incesto	Tipo de Ruido
0.01	Simple inversión	Agregar	Ninguno	-----	Ninguno

Para la optimización térmica se usará la secuencia del caso 41:

CASO	Longitud de cromosoma	The best	Preselección	Selección	Probabilidad de cruce	Número de Puntos	Tipo de Cruce
41	10	Copiar el mejor	Escalamiento lineal	Ruleta	0.75	2	De n puntos

Probabilidad de mutación	Tipo de Mutación	Tipo de Reemplazo	Tipo de niching	Prevención de Incesto	Tipo de Ruido
0.01	Simple inversión	Agregar	Ninguno	-----	Gauss

Adicionalmente se han realizado una serie de corridas para averiguar cual es el mejor número de individuos. Se ha probado en la optimización, y se ha establecido cual responde mejor, se ha obtenido que en las hidráulicas se va a colocar 30 individuos y en las térmicas se va a colocar 40 individuos.

Con la secuencia de operadores y tipos se procede a realizar los casos del acápite 7.1.4:

▪ Comprobación de flujo de potencia

En el sistema de 9 barras la matriz de admitancia hallada es la siguiente:

	1	2	3	4	5	6	7	8	9
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	3.307	-1.365	-1.942	0.000	0.000	0.000	0.000	0.000
3	0.000	-1.365	2.553	0.000	-1.188	0.000	0.000	0.000	0.000
4	0.000	-1.942	0.000	3.224	0.000	0.000	-1.282	0.000	0.000
5	0.000	0.000	-1.188	0.000	69.686	-68.499	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	-68.499	69.654	-1.155	0.000	0.000
7	0.000	0.000	0.000	-1.282	0.000	-1.155	2.437	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Matriz de Conductancia G

	1	2	3	4	5	6	7	8	9
1	-17.361	17.361	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	17.361	-39.309	11.604	10.511	0.000	0.000	0.000	0.000	0.000
3	0.000	11.604	-17.338	0.000	5.975	0.000	0.000	0.000	0.000
4	0.000	10.511	0.000	-15.841	0.000	0.000	5.588	0.000	0.000
5	0.000	0.000	5.975	0.000	-79.770	58.022	0.000	16.000	0.000
6	0.000	0.000	0.000	0.000	58.022	-67.628	9.784	0.000	0.000
7	0.000	0.000	0.000	5.588	0.000	9.784	-32.154	0.000	17.065
8	0.000	0.000	0.000	0.000	16.000	0.000	0.000	-16.000	0.000
9	0.000	0.000	0.000	0.000	0.000	0.000	17.065	0.000	-17.065

Matriz de Susceptancia B

Hay que considerar que la matriz esta ordenada de forma tal que la barra slack ocupa la fila 1, las siguientes son barras PQ y las restantes son barras PV. A continuación se muestran las tensiones de barra calculadas mediante el programa, además también se aprecia el verdadero orden de las barras:

Bus	E	f
1	1.04	0
4	1.02531	-0.03969
5	0.99421	-0.07578
6	1.01101	-0.05808
7	1.02588	0.0466
8	1.01882	0.04076
9	1.03203	0.05595
2	1.01464	0.14539
3	1.01976	0.10355

La potencia se calcula con la ecuación(4.5):

$$P_p - jQ_p = E_p^* I_p$$

Combinando con la ecuación (4.1), en forma matricial se obtiene:

$$P_{BUS} - jQ_{BUS} = (E_{BUS}^*)^{\wedge} (Y_{BUS} E_{BUS})$$

donde el circunflejo “ \wedge ” indica un producto de elemento a elemento. Al separar por partes la ecuación se puede obtener:

$$I_{BUS} = c_{BUS} + jd_{BUS} = (G_{BUS} e_{BUS} - B_{BUS} f_{BUS}) + j(B_{BUS} e_{BUS} + G_{BUS} f_{BUS})$$

finalmente, las potencias en barra:

$$P_{BUS} = e_{BUS} \wedge c_{BUS} + f_{BUS} \wedge d_{BUS}$$

$$Q_{BUS} = -(e_{BUS} \wedge d_{BUS} - f_{BUS} \wedge c_{BUS})$$

A continuación se presenta una tabla donde se muestra los resultados de las operaciones:

Bus	$G_{BUS}e_{BUS}$	$B_{BUS}f_{BUS}$	c_{BUS}	$B_{BUS}e_{BUS}$	$G_{BUS}f_{BUS}$	d_{BUS}	P_{BUS}	Q_{BUS}
1	0.00000	-0.68906	0.68906	-0.25503	0.00000	-0.25503	0.71662	0.26524
4	0.07023	0.07035	-0.00012	-0.08493	0.08499	0.00006	-0.00013	-0.00006
5	-0.08007	1.13176	-1.21183	0.78983	-0.19461	0.59523	-1.24992	-0.49995
6	-0.05472	0.81553	-0.87026	0.52861	-0.18190	0.34671	-0.89997	-0.29998
7	0.52121	0.52116	0.00005	-0.54521	0.54537	0.00016	0.00006	-0.00016
8	-0.49886	0.49477	-0.99363	0.72125	-0.41758	0.30368	-0.99995	-0.34989
9	0.04221	0.04230	-0.00009	-0.16352	0.16373	0.00021	-0.00008	-0.00022
2	0.00000	-1.58064	1.58064	0.17984	0.00000	0.17984	1.62993	0.04734
3	0.00000	-0.81229	0.81229	0.20939	0.00000	0.20939	0.85002	-0.12941

Los datos de potencia ordenados quedan:

Bus	P_{BUS}	$P_{Schedule}$	errorPU	Q_{BUS}	$Q_{Schedule}$	errorPU
1	0.71662			0.26524		
2	1.62993	1.630	0.0001	0.04734		
3	0.85002	0.850	0.0000	-0.12941		
4	-0.00013	0.000	0.0001	-0.00006	0.000	0.0001
5	-1.24992	-1.250	-0.0001	-0.49995	-0.500	0.0000
6	-0.89997	-0.900	0.0000	-0.29998	-0.300	0.0000
7	0.00006	0.000	-0.0001	-0.00016	0.000	0.0002
8	-0.99995	-1.000	0.0000	-0.34989	-0.350	-0.0001
9	-0.00008	0.000	0.0001	-0.00022	0.000	0.0002

Se aprecia que el error es menor a 0.001pu que es el valor asignado en el programa, esto no se ve en la barra slack(potencia activa y reactiva) y de generación(potencia reactiva).

En el caso de las 30 barras la matriz de admitancia es:

CUADRO 03

Las tensiones de barra calculadas mediante el programa son:

Bus	e	f
1	1.05	0
3	1.0249	-0.06918
4	1.01858	-0.0823
6	1.01088	-0.09668
7	0.99854	-0.11229
9	1.01984	-0.13303
10	1.01276	-0.15222
12	1.01713	-0.13665
14	1.00137	-0.15075
15	0.99827	-0.152
16	1.00722	-0.14743
17	1.00512	-0.15385
18	0.98909	-0.16288
19	0.98735	-0.16631
20	0.99275	-0.16379
21	1.00491	-0.16396
22	1.00722	-0.16617
23	0.9912	-0.15581
24	1.01577	-0.19858
25	0.99877	-0.18332
26	0.98003	-0.18742
27	0.99726	-0.17114
28	1.00685	-0.10508
29	0.97367	-0.18927
30	0.95931	-0.20228
2	1.04446	-0.0338
5	1.00351	-0.11452
8	1.00506	-0.09993
11	1.05561	-0.09683
13	1.04469	-0.11277

A continuación se presenta la tabla donde se muestran los resultados de las operaciones:

Bus	$G_{BUS}e_{BUS}$	$B_{BUS}f_{BUS}$	c_{BUS}	$B_{BUS}e_{BUS}$	$G_{BUS}f_{BUS}$	d_{BUS}	P_{BUS}	Q_{BUS}
1	0.06015	-0.88145	0.94160	-0.19003	0.26259	0.07256	0.98868	-0.07618
3	0.02059	0.04288	-0.02229	-0.00824	0.02151	0.01327	-0.02377	-0.01205
4	-0.04658	0.02653	-0.07311	0.11955	-0.09807	0.02148	-0.07624	-0.01586
6	-0.00751	-0.00765	0.00014	0.08519	-0.08511	0.00007	0.00013	-0.00009
7	-0.05898	0.15446	-0.21344	0.18243	-0.04945	0.13298	-0.22806	-0.10882
9	0.00000	-0.00011	0.00011	0.00017	0.00000	0.00017	0.00009	-0.00019
10	0.12050	0.17361	-0.05311	-0.09608	0.12353	0.02745	-0.05797	-0.01972
12	0.10178	0.20029	-0.09851	-0.00316	0.09008	0.08692	-0.11207	-0.07495
14	-0.01634	0.04193	-0.05827	0.04304	-0.01841	0.02462	-0.06206	-0.01587
15	-0.03559	0.04110	-0.07669	0.05998	-0.02346	0.03651	-0.08211	-0.02479
16	-0.01542	0.01613	-0.03155	0.03148	-0.00905	0.02243	-0.03509	-0.01794

17	-0.03415	0.04493	-0.07908	0.08802	-0.01844	0.06958	-0.09019	-0.05777
18	-0.01124	0.01883	-0.03008	0.02307	-0.00912	0.01395	-0.03202	-0.00890
19	-0.03712	0.05098	-0.08809	0.07435	-0.02537	0.04898	-0.09512	-0.03371
20	-0.00395	0.01646	-0.02041	0.01622	-0.00583	0.01039	-0.02197	-0.00697
21	-0.08611	0.06626	-0.15237	0.15169	-0.01582	0.13587	-0.17540	-0.11155
22	0.00983	0.00983	0.00000	-0.00176	0.00173	-0.00003	0.00000	0.00003
23	-0.01392	0.01515	-0.02906	0.02811	-0.00750	0.02061	-0.03202	-0.01590
24	0.04399	0.11342	-0.06943	0.18124	-0.10233	0.07892	-0.08620	-0.06637
25	0.00352	0.00351	0.00000	-0.00085	0.00088	0.00003	0.00000	-0.00003
26	-0.02280	0.00745	-0.03025	0.03405	-0.00499	0.02907	-0.03509	-0.02282
27	0.04659	0.04655	0.00004	-0.06357	0.06355	-0.00002	0.00004	0.00002
28	-0.01499	-0.01497	-0.00002	0.04403	-0.04408	-0.00004	-0.00002	0.00005
29	-0.01039	0.01168	-0.02207	0.01963	-0.00618	0.01344	-0.02403	-0.00891
30	-0.03919	0.06272	-0.10190	0.07385	-0.03327	0.04058	-0.10596	-0.01832
2	0.11834	-0.43155	0.54989	-0.37098	0.10389	-0.26709	0.58337	0.26038
5	-0.03183	0.40006	-0.43189	0.17401	-0.09827	0.07574	-0.44208	-0.02655
8	-0.03919	0.04686	-0.08605	0.14926	-0.01300	0.13625	-0.10010	-0.12834
11	0.00000	-0.17404	0.17404	-0.17197	0.00000	-0.17197	0.20037	0.16468
13	0.00000	-0.17057	0.17057	-0.19686	0.00000	-0.19686	0.20039	0.18642

Los datos de potencia ordenados quedan:

Bus	P_{BUS}	$P_{Schedule}$	errorPU	Q_{BUS}	$Q_{Schedule}$	errorPU
1	0.98868			-0.07618		
2	0.58337	0.583	-0.0004	0.26038	-0.127	
3	-0.02377	-0.024	-0.0002	-0.01205	-0.012	0.0000
4	-0.07624	-0.076	0.0002	-0.01586	-0.016	-0.0001
5	-0.44208	-0.442	0.0001	-0.02655	-0.190	
6	0.00013	0.000	-0.0001	-0.00009	0.000	0.0001
7	-0.22806	-0.228	0.0001	-0.10882	-0.109	-0.0002
8	-0.10010	-0.100	0.0001	-0.12834	-0.300	
9	0.00009	0.000	-0.0001	-0.00019	0.000	0.0002
10	-0.05797	-0.058	0.0000	-0.01972	-0.020	-0.0003
11	0.20037	0.200	-0.0004	0.16468	0.000	
12	-0.11207	-0.112	0.0001	-0.07495	-0.075	0.0000
13	0.20039	0.200	-0.0004	0.18642	0.000	
14	-0.06206	-0.062	0.0001	-0.01587	-0.016	-0.0001
15	-0.08211	-0.082	0.0001	-0.02479	-0.025	-0.0002
16	-0.03509	-0.035	0.0001	-0.01794	-0.018	-0.0001
17	-0.09019	-0.090	0.0002	-0.05777	-0.058	-0.0002
18	-0.03202	-0.032	0.0000	-0.00890	-0.009	-0.0001
19	-0.09512	-0.095	0.0001	-0.03371	-0.034	-0.0003
20	-0.02197	-0.022	0.0000	-0.00697	-0.007	0.0000
21	-0.17540	-0.175	0.0004	-0.11155	-0.112	-0.0004
22	0.00000	0.000	0.0000	0.00003	0.000	0.0000
23	-0.03202	-0.032	0.0000	-0.01590	-0.016	-0.0001
24	-0.08620	-0.087	-0.0008	-0.06637	-0.067	-0.0006
25	0.00000	0.000	0.0000	-0.00003	0.000	0.0000
26	-0.03509	-0.035	0.0001	-0.02282	-0.023	-0.0002

27	0.00004	0.000	0.0000	0.00002	0.000	0.0000
28	-0.00002	0.000	0.0000	0.00005	0.000	-0.0001
29	-0.02403	-0.024	0.0000	-0.00891	-0.009	-0.0001
30	-0.10596	-0.106	0.0000	-0.01832	-0.019	-0.0007

El error es menor a 0.001pu que es el valor que maneja el programa. Así que al igual que en el caso de 9 barras, el programa corre el flujo de potencia correctamente.

▪ **IEEE-9 con el Gen2 Hidráulico y el resto fijo**

Se ha elegido un error de 0.001 para todos los casos. Los resultados son los siguientes:

```
Potencia en generadores despues de optimizar hidraulicas:
0 100.00000 1 114.08929 2 85.00000
.....
Potencia en generadores despues de optimizar termicas:
0 100.00000 1 114.08929 2 85.00000
Costo final: 0.00
1 1.14089 2 0.85000
Barra slack: 1.18886 Pérdidas: 0.02975
Factores de penalidad en generadores:
0 1.00000 1 1.05501 2 1.05054
Error: 0
Iteracion: 1
```

Hay que notar que los resultados de la central hidráulica se analizan en el anexo D.

▪ **IEEE-9 con los tres generadores de la referencia [16]**

Los resultados son los siguientes:

```
Potencia en generadores despues de optimizar hidraulicas:
0 100.00000 1 163.00000 2 85.00000
.....
Potencia en generadores despues de optimizar termicas:
0 154.03377 1 70.515472 93.07432
Costo final: 3681.10
1 0.70516 2 0.93074
Barra slack: 1.54079 Pérdidas: 0.02669
Factores de penalidad en generadores:
0 1.00000 1 1.06015 2 1.05657
Error: 0.00057
Iteracion: 5
```

- **IEEE-9 con una Gen3 Hidráulica, Gen1 grupo térmico No1 de [16] y Gen2 grupo térmico No1 de [15]**

Los resultados son los siguientes:

Potencia en generadores despues de optimizar hidraulicas:
 0 100.00000 1 163.00000 2 114.08929

Potencia en generadores despues de optimizar termicas:
 0 120.00000 1 84.101192 114.08929

Costo final: 1638.02

1 0.84101 2 1.14089

Barra slack: 1.19885 Pérdidas: 0.03076

Factores de penalidad en generadores:

0 1.00000 1 1.05971 2 1.06242

Error: 0.00007

Iteracion: 3

- **IEEE-30 con los cinco generadores de la referencia[15]**

Los resultados son los siguientes:

Potencia en generadores despues de optimizar hidraulicas:
 0 98.95000 1 80.00000 2 50.00000 3 20.00000 4 20.00000 5
 20.00000

Potencia en generadores despues de optimizar termicas:
 0 169.66607 1 49.73583 2 21.45174 3 21.52873 4 16.20240 5
 14.51917

Costo final: 805.65

1 0.280362 -0.72748 3 -0.08471 4 0.16202 5 0.14519

Barra slack: 1.69410 Pérdidas: 0.09448

Factores de penalidad en generadores:

0 1.00000 1 1.06671 2 1.12500 3 1.10290 4 1.16432 5 1.13860

Error: 0.00043

Iteracion: 6

- **IEEE-30 donde se reemplaza el Gen2 por los datos Hidráulicos**

Los resultados son los siguientes:

Potencia en generadores despues de optimizar hidraulicas:
 0 98.95000 1 114.08929 2 50.00000 3 20.00000 4 20.00000 5
 20.00000

Potencia en generadores despues de optimizar termicas:
 0 127.94052 1 114.08929 2 18.66957 3 10.00000 4 10.00000 5
 12.00000

Costo final: 463.15

1 0.923892 -0.75530 3 -0.20000 4 0.10000 5 0.12000

Barra slack: 1.27866 Pérdidas: 0.09225

Factores de penalidad en generadores:

0 1.000001 1.05753 2 1.12641 3 1.11280 4 1.17361 5 1.14602

Error: 0.00007

Iteracion: 3

CAPÍTULO VIII CONCLUSIONES

El manejo de las matrices esparcidas y el asignamiento dinámico de memoria, hace que los cálculos de flujo de potencia sean más eficientes, con lo que se obtiene un gran incremento en la velocidad. En el aspecto del almacenamiento se puede ver la cantidad de ceros que tiene la matriz de admitancia del sistema de 30 barras, si se tuviera que trabajar la matriz completa tendríamos que almacenar 1800 elementos de doble precisión, en cambio con el formato esparcido sólo tenemos que almacenar 110 elementos de precisión doble y 141 enteros para direccionar los elementos. Esto nos lleva a un ahorro en el jacobiano, sobretodo, en las barras PV, cabe mencionar que se ha buscado que el jacobiano no tenga ceros en la diagonal principal para facilitar la solución del sistema de ecuaciones lineales.

En la tesis al optimizar la potencia de una manera básica, esto es, sin considerar como variables de control las tensiones de generación, ni los taps controlados bajo carga, se obtienen buenos resultados. Se tiene que para el caso de IEEE-30 con los datos de los generadores de la referencia [15], que es el mismo caso que se analiza en la referencia, se obtiene un costo de 805.65(\$/h) y las pérdidas son de 9.45MW, mientras que en la referencia [15] se obtiene un costo de 802.4(\$/h), con unas pérdidas de 9.48MW, con lo cual se demuestra que el algoritmo genético funciona para la optimización térmica en función del costo. En el caso de la optimización

hidráulica; al compararla con otro método como se muestra en el Anexo D, se observa que en el caso de linealizar las funciones se obtienen buenos resultados(115.1442MW), pero el problema es que le quita algo de realidad, ya que con las funciones cuadráticas se obtiene un caudal de 18.313m³/s. Además hay que mencionar que si bien es cierto la suma total de potencias (115.14 y 114.09) son similares los puntos donde se encontraron son diferentes, esto es:

P1(MW)		P2(MW)		P3(MW)	
LINDO	FlexView	LINDO	FlexView	LINDO	FlexView
50.9000	47.7156	49.1900	47.8917	15.0542	18.4819

Debido a la diferencia en la concepción de ambos métodos, no se puede realizar una comparación directa entre sus resultados, pero si se puede concluir que el algoritmo genético cumple con la tarea de optimizar.

Es importante remarcar que en el algoritmo genético se deben hacer análisis previos, en ese sentido se puede ver como cada algoritmo reacciona de manera diferente a los operadores que se le aplican. En el caso de 30 barras se tuvo que modificar el número de individuos y ese parámetro tuvo una influencia sobre la velocidad de convergencia (se redujo a 2 o a 3 generaciones). Un exceso de individuos sería innecesario, en vez de ayudar empezaría a perjudicar, aumentando los elementos a almacenar y el número de operaciones. Por todo eso hay que decir que para aplicar los algoritmos genéticos además de seleccionar la función peso adecuada se debe buscar el juego de parámetros adecuados. En la actualidad se están usando métodos más estrictos para analizar los algoritmos genéticos como son las cadenas finitas de Markov ([26]), adicionalmente hay que notar que los operadores que se utilizaron son los más importantes, pero no son todos y que muchas veces los algoritmos genéticos aunados con métodos determinísticos (Newton) y con métodos

inteligentes (redes neuronales), en muchos casos se obtienen mejores resultados en muchos campos de la ingeniería de Sistemas de Potencia como son el Pronóstico de Carga, Valoración de la seguridad en estado estacionario y en estado dinámico, Estabilidad de Tensión, Planeamiento, Estimación de ubicación de fallas y Protección entre otras aplicaciones.

CAPÍTULO IX REFERENCIAS BIBLIOGRAFICAS

1. Glen W. Stagg , Ahmed H. El-abiad, *Computer Methods in Power System Analysis*, Mac Graw-Hill, 1968
2. IEEE Comittee Report, "*Present practices in the economic operation of power systems*", IEEE Trans. on Power Apparatus and Systems, PAS-90, pp. 1768-1775, July/Aug 1971
3. Wood, A. J. y B. F. Wollenberg, *Power Generation, Operation and Control*, John Wiley & Sons, New York, 1984
4. Walters, D.C. and G.B. Sheble, "*Genetic Algorithm solution of economic dispatch with valve point loading*", IEEE Trans. on Powr Systems, Vol. 8, No 3, pp. 1325-1332, August 1993
5. N.Sato y W.F. Tinney, "*Técnicas para explotar la esparcidad de la matriz de admitancia de redes*", IEEE Trans. on Power Apparatus and Systems, PAS-86, pp. 944-950, December 1963
6. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989
7. W.F. Tinney y C.E. Hart, "*Power flow solution by Newton's method*" presented at IEEE Winter Power Meeting, New York, N. Y., 1967

8. Lance Chamber, *Practical handbook of genetic algorithms*, Volume I Applications, CRC, 1995
9. William F. Tinney and John W. Walker, "*Direct Solutions of Sparse Network Equations by optimally ordered Triangular Factorization*" Proceedings of the IEEE, Vol 55 No 11, pp 1801-1809, November 1967
10. Marciano Morowski Filho, "*Matrizes Esparsas em Redes de Potência*" , Livros Técnicos e Científicos.
11. J. Carpentier, "*Ordered eliminations*", Proceedings of Power Systems Computation Conference, London (1963)
12. R. Bauman, "*Some new aspects of load flow calculations*", Proceedings of PICA, Florida (May 1965)
13. R.C. Burchett, H.H. Happ, D.R. Vierath, K.A. Wirgau, "*Developments in optimal Power Flow*", IEEE Trans. on Power Apparatus and Systems, PAS-101, pp. 406-420, February 1982
14. Pawlosky, Marc Andrew. "*Modified uniform crossover and desegregation in Genetic Algorithm*". Master thesis. Concordia University, Montreal, Canada (1992)
15. O. Alsac and B. Scott. "*Optimal Load Flow with Steady State-Security*" IEEE Trans. On Power Apparatus and Systems, May 1973
16. Po-Hung Chen and Hong-Chan Chang, "*Large-Scale Dispatch by Genetic Algorithm*", IEEE Trans. on Power Systems, Vol 10 No 4, November 1995
17. Huber A. Salas Santos, "*Despacho económico utilizando algoritmos genéticos*", INTERCON' 99, Memorias del Concurso de Proyectos, pp. 42-47, Agosto 1999

18. Ebrahim Vaahedi and H. Magdy Zein El-Din, "*Considerations in Applying Optimal Power Flow to Power System Operation*", IEEE Trans. on Power Systems, Vol 4 No 2, May 1989
19. F. Schlaepfer, T.C. Kelly and A. G. Dewey, "*An interactive Load Flow Program*", IEEE Trans. on Power Apparatus and Systems, pp. 78-84, June 1971
20. Roldan Pozo and Karin A. Remington, "*Reference Guide of SparseLib++ v. 1.5*", April 1996
21. Albert M. Sasson and Fernando J. Jaimes, "*Digital Methods Applied to Power Flow Studies*", IEEE Trans. on Power Apparatus and Systems, PAS-86, No7, July 1967
22. Yehuda Wallach, "*Gradient Methods for Load-Flow Problems*", IEEE Trans. on Power Apparatus and Systems, PAS-87, No5, May 1968
23. A.P. Bonaert, A. H. El-Abiad, A. J. Koivo, "*Optimal Scheduling of Hydro-Thermal Power Systems*", IEEE Trans. on Power Apparatus and Systems, May 1971
24. David I. Sun, Bruce Ashley, Brian Brewer, Art Hughes, William F. Tinney, "*Optimal Power Flow by Newton Approach*", IEEE Trans. on Power Apparatus and Systems, PAS-103, No10, October 1984
25. H. H. Happ, "*Optimal Power Dispatch*", IEEE Trans. on Power Apparatus and Systems, May 1973
26. Samir W. Mahfoud, "*Finite Markov Chain Models of an Alternative Selection Strategy for the Genetic Algorithm*", Department of Computer Science, University of Illinois, April 1993

27. Kalyanmoy Deb and David E. Goldberg, "*A Messy Genetic Algorithm in C*",
Department of General Engineering, University of Illinois, September 1991
28. Roldan Pozo and Sharon L. Smith, "*Limited Resource Scheduling in Sparse Matrix Algorithms*", CERFACS, September 1993
29. Roldan Pozo and Victor Eijkhout, "*Basic Concepts for Distributed Sparse Linear Algebra Operations*", University of Tennessee, August 1995
30. Roldan , Jack Dongarra, Andrew Lumsdaine, Xinhu Niu and Karin Remington, "*A Sparse Matrix Library in C++ for High Performance Architectures*", University of Tennessee, August 1995

ANEXO A

TÉCNICAS PARA LA GENERACIÓN DE LOS NÚMEROS ALEATORIOS

A.1. Introducción

La siguiente terminología será muy útil al tratar el concepto referente a la generación de los números aleatorios. El término *variable aleatoria* se emplea para nombrar una función de valor real, definida sobre un espacio muestral asociado con los resultados de un experimento conceptual de naturaleza azarosa. El resultado particular de un experimento, o sea, un valor numérico o de la muestra de una variable aleatoria, se llama *valor de la variable aleatoria*. Las letras mayúsculas se usan para denotar las variables aleatorias, y las minúsculas para los valores de las variables aleatorias. Por ejemplo, $F(x)$; la función de la distribución acumulativa para una variable aleatoria X , indica la probabilidad de que X sea menor o igual al particular valor x de la variable aleatoria. En forma semejante, $f(x)$ representa el valor de la función de densidad de probabilidad de la variable aleatoria X cuando $X=x$.

Las variables aleatorias se clasifican, de una manera muy conveniente, de acuerdo con sus funciones de densidad de probabilidad. Se tratarán exclusivamente las variables aleatorias con *funciones de densidad de probabilidad uniforme*. La función de distribución acumulativa para la distribución uniforme estandarizada está definida por

$$F(x) = \begin{cases} 0, & x \leq 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x \geq 1 \end{cases} \quad (\text{a.1})$$

Los valores de x , en el intervalo unitario, se llamarán *valores uniformes de las variables aleatorias*. Los valores de las variables aleatorias que se consideran aquí, se representarán por números que, al menos parezcan haberse obtenido al azar, como los valores de muestra de una población uniformemente distribuida. El valor de la variable aleatoria se emplea como un término colectivo, que significa números aleatorios dados en forma de dígitos, enteros o números racionales, con un intervalo y un sistema de numeración bien definidos. Puesto que en los párrafos que siguen, se enfatizarán las propiedades de los números correspondientes a los valores uniformes de las variables aleatorias, se utilizará el término *número aleatorio uniforme* o simplemente *número aleatorio*, en lugar del término colectivo.

En la práctica, generalmente se requieren *sucesiones de números aleatorios*. En consecuencia, todos los métodos que aquí se consideran involucran algún proceso físico cuasialeatorio, que genera sucesiones de números aleatorios de cualquier longitud que se desee. Uno de los requisitos principales se imponen sobre estas sucesiones, al igual que cualesquiera otros procesos aleatorios de muestreo, es el de la independencia estadística.

Los técnicos han venido usando cuatro métodos alternativos para generar las sucesiones de números aleatorios:

1. Métodos manuales
2. Tablas de biblioteca
3. Métodos de computación analógica

4. Métodos de computación digital

En nuestro caso interesan los métodos de computación digital. Toucher ha sugerido tres métodos para producir números aleatorios cuando se usan *computadoras digitales*: la provisión externa, la generación interna a partir de un proceso físico al azar y la generación interna de sucesiones de dígitos por medio de una relación de recurrencia. De estos tres métodos se ha utilizado la generación de sucesiones de dígitos por medio de una relación de recurrencia. Esta alternativa comprende la generación de números pseudoaleatorios por medio de “una transformación indefinidamente continuada, aplicada a un grupo de números elegidos en forma arbitraria”. El término *número pseudoaleatorio* ha sido definido por Lehmer como “una noción vaga que encierra la idea de una sucesión en la cual cada término es impredecible para la persona ajena al problema, cuyos dígitos se someten a cierto número de pruebas, tradicionales a los estadísticos, y depende en cierta forma del uso que se dará a la sucesión”. En este método no se presenta el problema de los dispositivos de entrada o de la capacidad de almacenamiento de la computadora; todo el proceso se puede reproducir, dependiendo sólo de los pasos aritméticos, Toucher ha declarado que la principal objeción a esta solución radica en los aspectos un tanto filosóficos respecto a que una sucesión de dígitos, generada mediante una regla puramente determinística, resulta ser la antítesis directa de una sucesión aleatoria. Sin embargo, esta objeción puede superarse, al menos parcialmente, al tomar el punto pragmático de que una sucesión aleatoria. Sin embargo, esta objeción puede superarse, al menos parcialmente, al tomar el punto de vista un tanto pragmático de que una sucesión puede considerarse aleatoria si satisface un cierto conjunto de pruebas estadísticas de aleatoriedad, previamente

determinadas. Desde punto de vista, el método para generar una sucesión es totalmente independiente.

Después de examinar varios métodos para generar números aleatorios parece ser que el aceptable de todos deberá producir sucesiones de números que sean: (1) uniformemente distribuidos, (2) estadísticamente independientes. (3) reproducibles y (4) sin repetición dentro de una longitud determinada de la sucesión. Aún más, tal método deberá ser capaz de: (5) generar números aleatorios a grandes velocidades, así como también (6) requerir un mínimo de la capacidad de almacenamiento de la computadora. Los métodos de congruencias se diseñaron específicamente para satisfacer, tanto como sea posible, los requisitos antes mencionados.

A.2. Métodos de Congruencias para Generar Números Pseudoaleatorios

Se dispone de fórmulas que permiten calcular de antemano el valor exacto del i -ésimo número en una sucesión $\{n_0, n_1, n_2, \dots, n_i, \dots\}$ sin que necesariamente se haya generado toda la sucesión. Aunque estos no son del todo aleatorios, hay fundamentos pragmáticos que nos permiten considerarlos como si en efecto lo fueran, si las sucesiones que resultan se someten consistentemente con éxito a cierto conjunto de pruebas estadísticas diseñadas para probar varias propiedades de los números aleatorios.

Los métodos de congruencias se basan en una relación fundamental de congruencia que puede expresarse por medio de la siguiente fórmula recursiva

$$n_{i+1} = an_i + c(\text{mod } m) \quad (\text{a.2})$$

donde n_i , a , c y m son enteros no negativos. Al desarrollar la ecuación (a.2) para $i=0, 1, 2, \dots$, se obtienen:

$$\begin{aligned}
n_1 &= an_0 + c \pmod{m} \\
n_2 &= an_1 + c \pmod{m} = a^2 n_0 + (a+1)c \pmod{m} \\
n_3 &= a^3 n_0 + (a^2 + a + 1)c = a^3 n_0 + \frac{c(a^3 - 1)}{(a-1)} \pmod{m} \\
&\dots\dots\dots \\
n_i &= a^i n_0 + \frac{c(a^i - 1)}{(a-1)} \pmod{m}
\end{aligned} \tag{a.3}$$

Una vez dados un *valor inicial* n_0 , un *factor constante* a y una *constante aditiva* c , las ecuaciones (a.3) conducen a una relación de congruencia (módulo m) para todo valor de i en la sucesión $\{n_1, n_2, \dots, n_i, \dots\}$. Los términos subsecuentes de $\{n_i\}$ son todos enteros tal como lo determina la ecuación (a.3) y forman una sucesión de residuo módulo m . Esto, a su vez implica que $n_i < m$ para todo n_i ; a partir de los números enteros de la sucesión $\{n_i\}$, se pueden obtener números racionales en el intervalo $(0,1)$ con sólo formar la sucesión $\{r_i\} = \{n_i/m\}$.

Respecto a la pregunta de si existe un mínimo valor positivo para i , $i=h$, tal que $n_h = n_0$ en donde h es el *período* de la sucesión $\{n_i\}$; si tal h existe, cabe preguntarse ¿qué condiciones se pueden imponer sobre n_0 , a , c , y m , a fin de que el período de $\{n_i\}$ sea lo más largo posible?. Nuestro interés en este problema se origina en el hecho de que, si $n_i = n_0$ para alguna $i=h$, entonces $n_{h+1} = n_1$, $n_{h+2} = n_2$, etc. Esto es, la sucesión de números pseudoaleatorios se volverá a repetir después de un período h .

Se tienen teoremas para probar que h siempre existe y que además su valor máximo depende de m . Esto equivale a decir que es imposible obtener sucesiones que no se repiten, utilizando los métodos de congruencias. Sin embargo, en la práctica, el período de una sucesión puede fijarse con un valor muy alto con sólo elegir un módulo lo suficientemente grande, o bien, por otras técnicas.

Se han desarrollado tres métodos básicos de congruencias para generar pseudoaleatorios mediante el ejemplo de distintas versiones de la relación dada en la ecuación (a.2). El objetivo de cada uno de los métodos es la generación, en un mínimo de tiempo, de sucesiones con un período máximo. Estos métodos de congruencias son: el aditivo, el multiplicativo y el mixto.

El método aditivo de congruencias presupone k valores iniciales dados, con k un entero positivo y computa una sucesión de números mediante la siguiente relación de congruencia:

$$n_{i+1} = n_i + n_{i-k} \pmod{m} \quad (\text{a.4})$$

Si $k=1$, la ecuación (a.4) genera la bien conocida sucesión de Fibonacci, la cual se comporta como las sucesiones que se pueden obtener por el método multiplicativo de congruencias, en las que se tiene el muy desfavorable multiplicador $a=(1+\sqrt{5})/2$; las propiedades estadísticas de la sucesión tienden a mejorarse a medida que k se incrementa. Este es el único método que produce períodos mayores que m .

El método multiplicativo de congruencias calcula una sucesión $\{n_i\}$ de enteros no negativos, cada uno de los cuales siempre es menor que m , por medio de la relación de congruencia

$$n_{i+1} = an_i \pmod{m} \quad (\text{a.5})$$

Este método es un caso especial de la citada relación de congruencia, en donde $c=0$. Se ha encontrado que el método multiplicativo se comporta de manera muy satisfactoria en lo que toca a su estadística, esto es, tanto las pruebas de frecuencia y las de serie, como otras pruebas relativas a la aleatoriedad, cuando se aplican las sucesiones que se obtienen mediante este método indican que los números aleatorios

así encontrados están uniformemente distribuidos y no correlacionados. Más aún, para fines de asegurar un período máximo para las sucesiones generadas por este método, es posible imponer condiciones convenientes tanto para el multiplicador a como para el valor inicial N_0 con que se empieza el cálculo. Existen otras ventajas que también ofrece el método multiplicativo en términos de las velocidades de computación.

Los números que se obtienen mediante la relación de congruencia en su forma original (con a y c mayores que cero), se dice que son generados por el *método mixto de congruencias*. Tal método ofrece algunas pequeñas ventajas, al compararlo con el método multiplicativo, en términos de incremento de velocidad y pérdida de periodicidad en los últimos dígitos. La ventaja principal del método mixto radica en su período completo. Aunque su comportamiento estadístico es generalmente bueno, en pocos casos resulta inaceptable por completo.

Aunque los estudios teóricos previos relativos a las propiedades tal como se analizan en la teoría de los números, especialmente lo relativo al período y la simplicidad del cálculo de sucesiones de los números pseudoaleatorios, la fórmula de Greenberger presenta condiciones teóricas sobre los valores óptimos de a , c y m en la ecuación (a.2), a partir del punto de vista de una determinación *a priori* de la correlación seriada entre los números. La magnitud del coeficiente de correlación seriada $\rho(n_i, n_{i+1})$ se encuentra entre los valores

$$\frac{1}{a} - \left(\frac{6c}{am} \right) \left(1 - \frac{c}{m} \right) \pm \frac{a}{m} \quad (\text{a.6})$$

A partir de la ecuación (a.6) se puede obtener otra conclusión interesante y radica en que si se muestrea solamente cada k -ésimo número de la sucesión, esto

tendrá el efecto equivalente a elegir a a^k como el multiplicador y $c_k=(a^k-1)c/(a-1)$ como la constante aditiva, en la ecuación (a.2).

ANEXO B MODELOS DE LOS ELEMENTOS DEL SISTEMA

B.1. El transformador de dos devanados

El transformador de dos devanados es un elemento de dos puertos conectado a dos cubículos en un sistema de potencia. El diagrama de la representación de la secuencia positiva equivalente se muestra en la figura b.1.

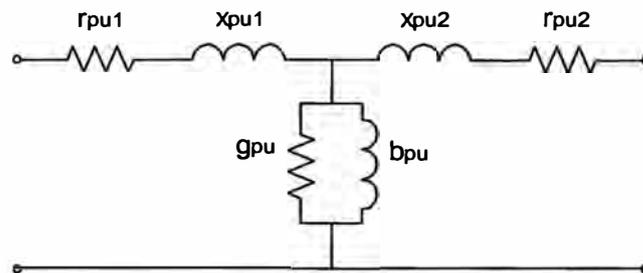


Figura b.1 Modelo del transformador de dos devanados

La relación entre los parámetros matemáticos en el modelo y los parámetros de ingreso en la base de datos son los siguientes:

$$z_{pu} = \frac{V_{cc} (\%)}{P_{nom}}$$
$$r_{pu} = \frac{P_{cu} (\%)}{P_{nom}}$$
$$x_{pu} = \sqrt{z_{pu}^2 - r_{pu}^2}$$

$$y_o = \left(\frac{C_{mag}(\%)}{100} \right) \times \left(\frac{P_{nom}}{100} \right)$$

$$g_{pu} = \left(\frac{P_{fe}(\%)}{100} \right) \times \left(\frac{P_{nom}}{100} \right)$$

$$b_{pu} = -\sqrt{y_o^2 - g_{pu}^2}$$

Para cada devanado del transformador, se divide la impedancia de cortocircuito en dos.

B.2. La línea de transmisión

La línea de transmisión es un elemento de dos puertos conectado a dos cubículos en un sistema de potencia. El diagrama de representación de la secuencia positiva se muestra en la figura b.2.

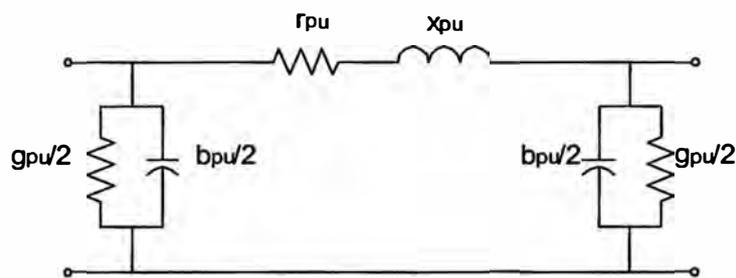


Figura b.2 Modelo de la línea de transmisión

La relación entre los parámetros matemáticos en el modelo y los parámetros de ingreso en la base de datos son los siguientes:

$$r_{pu} = \frac{R * L}{Z_{base}}$$

$$x_{pu} = \frac{X * L}{Z_{base}}$$

$$b_{pu} = C * 2\pi f * 1e - 9 * L * Z_{base}$$

$$g_{pu} = G * 1e - 6 * L * Z_{base}$$

B.3. El elemento shunt

El elemento shunt se conecta a un cubículo en un sistema de potencia. El diagrama de representación se muestra en la figura b.3.

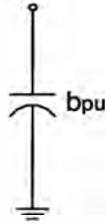


Figura b.3 Modelo del elemento shunt

La relación entre los parámetros matemáticos en el modelo y los parámetros de ingreso en la base de datos son los siguientes:

$$y_{pu} = \left(\frac{Q_{nom}}{V_{nom}^2} \right) * Z_{base}$$

ANEXO C MANUAL DEL PROGRAMA

C.1. Introducción

FlexView es el programa que ha sido diseñado para realizar el flujo de potencia óptimo utilizando algoritmos genéticos. El programa toma los datos del sistema a través de un archivo tipo *.mdb, donde cada tabla almacena datos de un tipo de elemento.

FlexView tiene las siguientes posibilidades:

- Abrir el archivo de datos y observar los datos que contiene.
- Realizar una optimización de las centrales hidráulicas
- Realizar optimización de las centrales térmicas, usando el resultado de flujo
- Realizar un flujo de potencia basándose en los datos de la base de datos

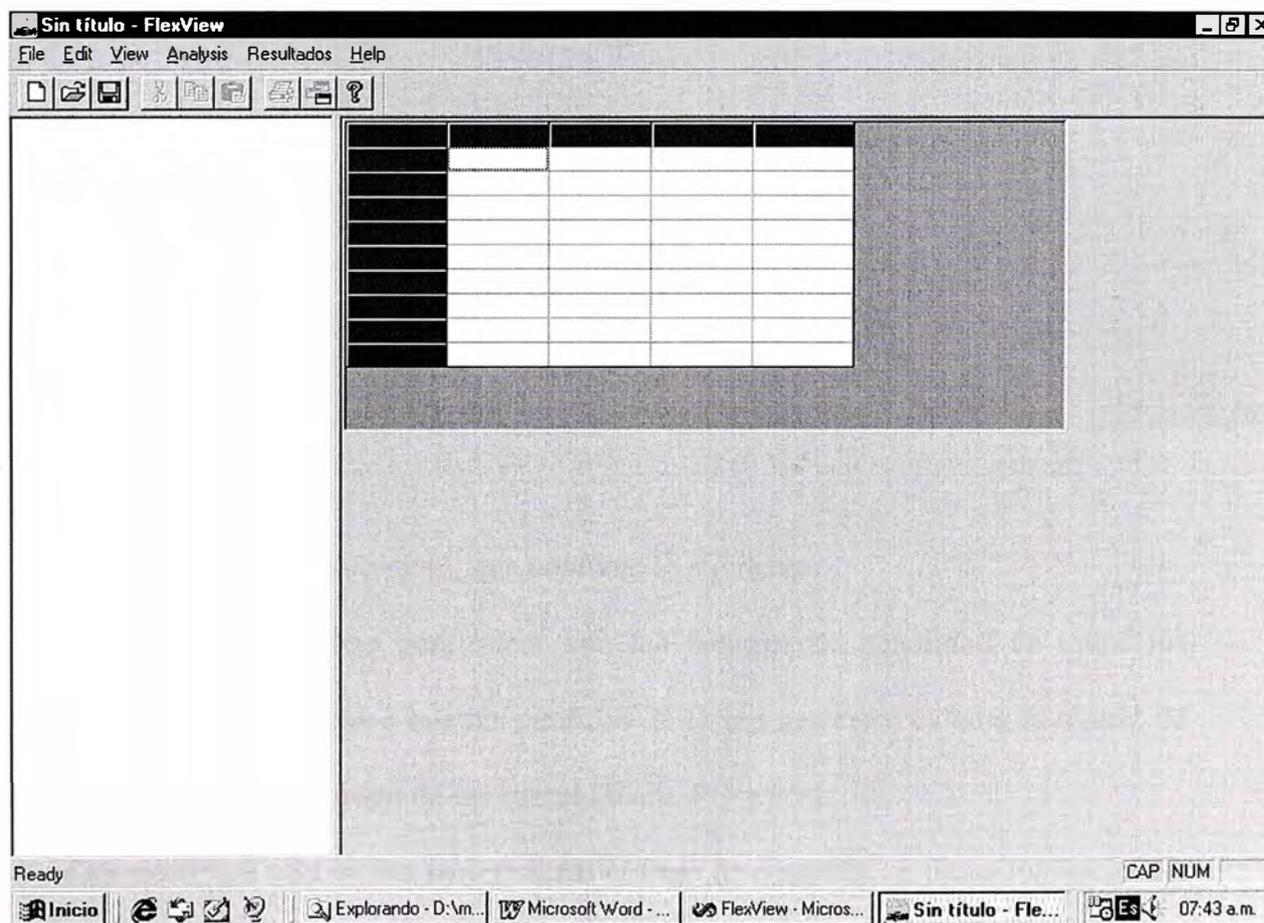
Realizar un flujo de potencia basándose en los datos optimizados

- Calcular los factores de penalidad
- Realizar el flujo de potencia óptimo completo
- Arrojar los resultados a archivo
- Se puede graficar los máximos.
- Se puede graficar los promedios.
- Se puede graficar la población final

C.2. Descripción de la Ventana Principal

Al ejecutar el programa aparece la ventana principal, que contiene el menú principal, este menú consta de las siguientes partes:

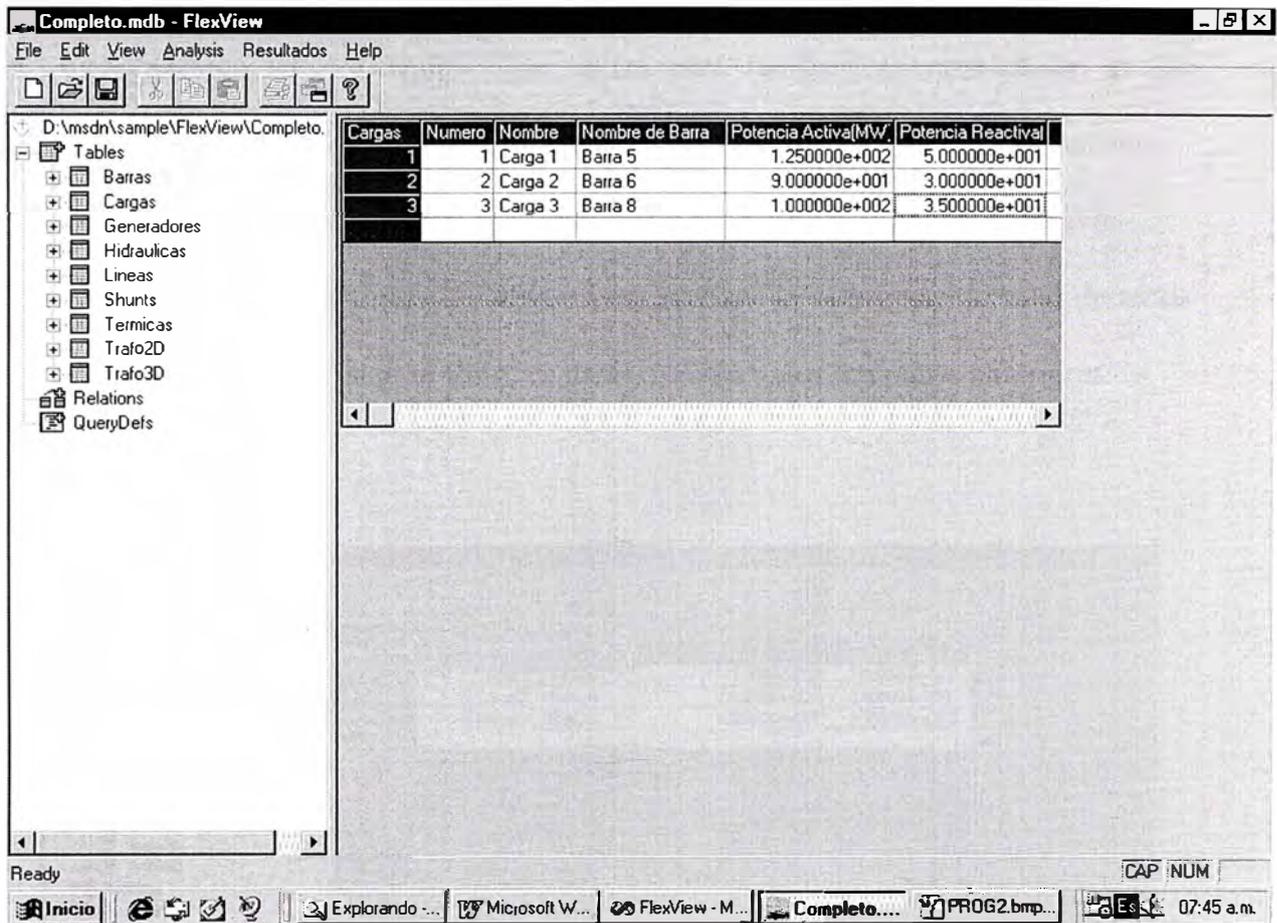
- File : Contiene lo necesario para el manejo de archivos
- Edit : No se ha implementado.
- View : Se utiliza, para ver u ocultar la barra de herramientas y la barra de estado.
- Analysis : Se utiliza para realizar todos los cálculos en el programa
- Resultados : Se utiliza para ver los gráficos de los resultados de la optimización



C.3. File

Lo primero que se hace es abrir un archivo de base de datos para cargar los datos del caso en que se va a trabajar, en ese momento se observan los tipos de

elementos al lado izquierdo y los datos de los elementos al lado derecho.

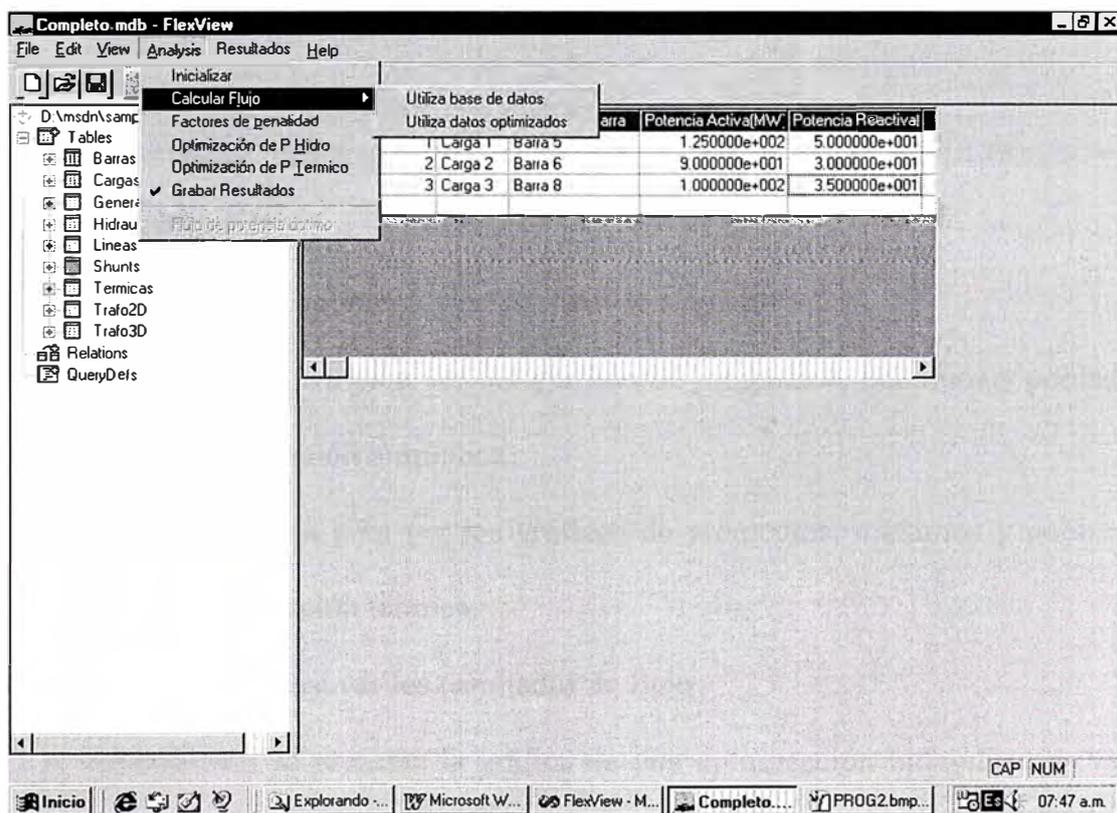


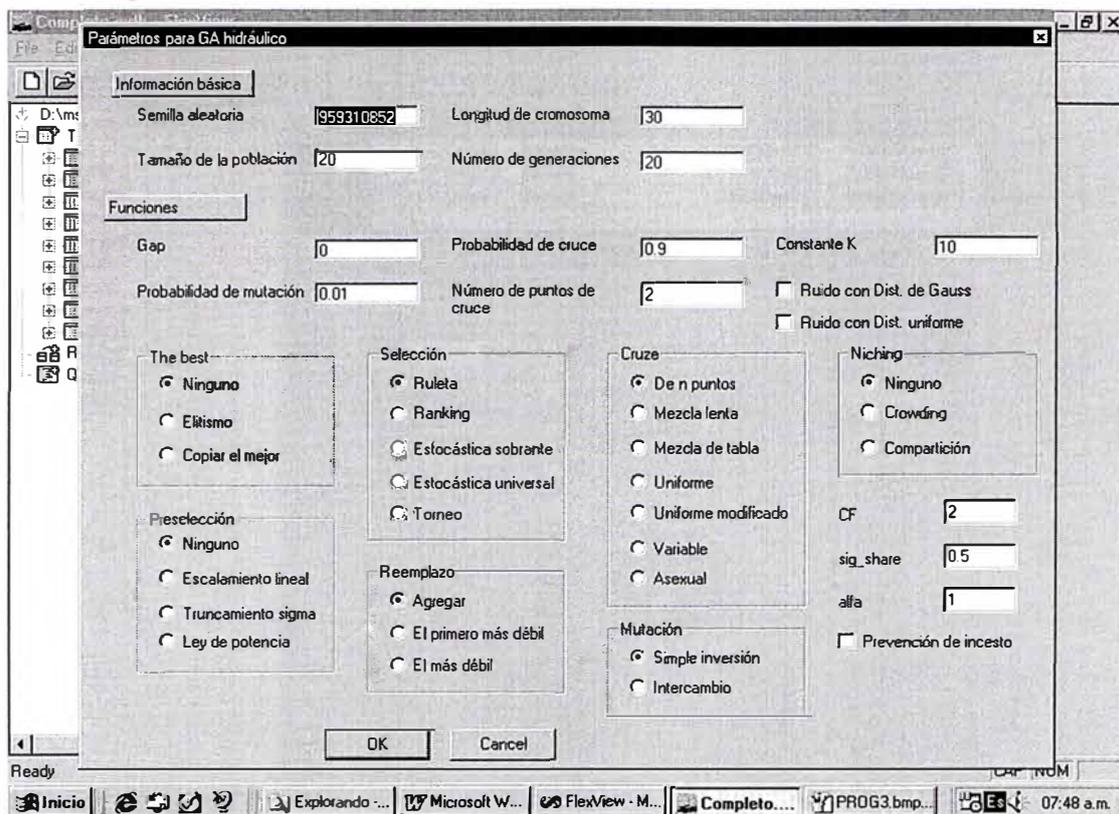
C.4. Analysis

Este ítem tiene un submenú, que contiene lo siguiente:

- Inicializar : Se utiliza para hacer que los factores de penalidad de todos los generadores sea 1, hace que las pérdidas de la red sea cero, calcula la matriz de admitancia pu y el orden de las barras (slack, PQ y PV)
- Calcular Flujo : Se utiliza para realizar el flujo de potencia, se tiene dos opciones, una es utilizando la base de datos y otra es utilizando los datos de la optimización.
- Factores de penalidad : Se utiliza para calcular los factores de penalidad, una vez que se haya realizado el flujo de potencia.

- Optimización de P Hidro : Se utiliza para realizar la optimización de las centrales hidráulicas.
- Optimización de P Termico : Se utiliza para realizar la optimización de las centrales térmicas, en base a los factores de penalidad y las pérdidas calculadas.
- Grabar resultados : Se utiliza para habilitar o deshabilitar la salida a archivo.
- Flujo de potencia óptimo : Se utiliza para realizar la optimización total de todos las centrales, incluyendo el flujo, es decir, realiza todos los pasos anteriores





C.5. Resultados

Este ítem tiene un submenú, que contiene lo siguiente:

- Hidraulica : Se utiliza para ver los gráficos de promedios, máximos y población final de la optimización hidráulica.
- Termica : Se utiliza para ver los gráficos de promedios, máximos y población final de la optimización térmica.
- Flujo : Se utiliza para ver los resultados de flujo

A continuación se presenta la gráfica de una optimización hidráulica, se ve la salida de los máximos versus las generaciones.

Completo.mdb - FlexView

File Edit View Analysis Resultados Help

D:\msdn\sample\FlexView\Completo.

Cargas	Numero	Nombre	Nombre de Barra	Potencia Activa(MW)	Potencia Reactiva
1	1	Carga 1	Barra 5	1.250000e+002	5.000000e+001
2	2	Carga 2	Barra 6	9.000000e+001	3.000000e+001
3	3	Carga 3	Barra 8	1.000000e+002	3.500000e+001

Tables

- Barras
- Cargas
- Generadores
- Hidraulicas
- Lineas
- Shunts
- Termicas
- Trafo2D
- Trafo3D
- Relations
- QueryDefs

Resultados de optimización Hidráulica

CE1

- Gráfico del promedio
- Gráfico del mejor
- Gráfico de ambos
- Gráfico de población final

OK Cancel

Ready CAP NUM

Inicio Exploran... Microsoft... FlexView... Completo... PROG4... Resultado... 07:50 a.m.

CompletoH.mdb - FlexView

File Edit View Analysis Resultados Help

D:\msdn\sample\FlexView\CompletoH

Barras	Numero	Nombre	Tension(kV)

Tables

- Barras
- Cargas
- Generadores
- Hidraulicas
- Lineas
- Shunts
- Termicas
- Trafo2D
- Trafo3D
- Relations
- QueryDefs

Salida gráfica

CE1 generacion

OK Cancel

Ready CAP NUM

Inicio Exploran... Microsoft... FlexView... Completo... PROG5... Resultado... 07:52 a.m.

ANEXO D RESULTADOS DE LA OPTIMIZACIÓN HIDRÁULICA

Para comprobar los resultados de la optimización hidráulica se utilizó el LINDO, que es un programa de optimización, utilizando el método de programación entera mixta.

D.1. Resultados del LINDO

El archivo de entrada que se utilizó para el problema es el siguiente:

```
terse
Max P1+P2+P3+P11+P21+P31
ST
Q1+Q2+Q3=18
Q1 - 0.086P1 - 1.26A1 - 0.153P11 - 0.3204B1 = 0
Q2 - 0.070P2 - 1.13A2 - 0.1543P21 - 0.4468B2 = 0
Q3 - 0.0802P3 - 0.0991A3 - 0.1768P31 + 0.8065B3 = 0
P1 14.145A1 < 0
P11- 50.900B1 < 0
P11- 14.145B1 >=0
A1+B1<=1
P2-7.318A2<0
P21- 49.190B2< 0
P21-7.318B2>=0
A2+B2<=1
P3-8.91A3<0
P31- 51.1B3 < 0
P31-8.91B3>=0
A3+B3<=1
END
INT A1
INT B1
INT A2
INT B2
INT A3
INT B3
DIVE HUBER.SAL
look all
GO
SOLU
RVRT
```

QUIT

Al ejecutar el LINDO se obtuvo los siguientes resultados:

LP OPTIMUM FOUND AT STEP 22
OBJECTIVE VALUE = 115.144203

NEW INTEGER SOLUTION OF 115.144203 AT BRANCH 0 PIVOT
28

BOUND ON OPTIMUM: 115.1442

ENUMERATION COMPLETE. BRANCHES= 0 PIVOTS= 28

LAST INTEGER SOLUTION IS THE BEST FOUND
RE-INSTALLING BEST SOLUTION...

OBJECTIVE FUNCTION VALUE

1) 115.1442

VARIABLE	VALUE	REDUCED COST
A1	0.000000	-0.137817
B1	1.000000	-5.039706
A2	0.000000	1.970801
B2	1.000000	-3.732888
A3	0.000000	-4.307726
B3	1.000000	-4.561652
P1	0.000000	0.000000
P2	0.000000	0.000000
P3	0.000000	0.000000
P11	50.900002	0.000000
P21	49.189999	0.000000
P31	15.054202	0.000000
Q1	8.108100	0.000000
Q2	8.036817	0.000000
Q3	1.855083	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	5.656109
3)	0.000000	-5.656109
4)	0.000000	-5.656109
5)	0.000000	-5.656109
6)	0.000000	0.513575
7)	0.000000	0.134615
8)	36.755001	0.000000
9)	0.000000	0.000000
10)	0.000000	0.604072
11)	0.000000	0.127262
12)	41.871998	0.000000
13)	0.000000	0.000000
14)	0.000000	0.546380
15)	36.045795	0.000000
16)	6.144203	0.000000
17)	0.000000	0.000000

NO. ITERATIONS= 33
BRANCHES= 0 DETERM.= 1.000E 0

Hay que notar que en este caso se tuvo que linealizar las funciones, debido a que la programación entera mixta se utiliza para funciones que tienen tramos rectos.

Resumiendo tenemos el siguiente resultado:

P1	q1	P2	q2	P3	q3	Suma de P	Suma de q
50.9000	8.1081	49.1900	8.0368	15.0542	1.8550	115.1442	18.0000

Si utilizamos las verdaderas funciones cuadráticas, no linealizadas se obtiene:

P1	q1	P2	q2	P3	q3	Suma de P	Suma de q
50.9000	8.2653	49.1900	8.2122	15.0542	1.8360	115.1442	18.3135

Se puede ver que la suma de caudales ha aumentado con las verdaderas funciones.

D.2. Resultados del FlexView

En la generación número trece se obtuvo el máximo que cumple con la condición de que la diferencia porcentual entre el máximo y el promedio de la generación se de 8%. La población resultante es la siguiente:

Numero	Cromosoma	Peso	Suma de P	Caudal	Error
1	0110111001011111001001110111111	4.68309	139.66426	22.788	0.2660
2	0010111001011111001001110111111	5.52277	114.08929	18.139	0.0077
3	0010101001111111001001110111111	5.46555	112.5408	17.874	0.0070
4	0110111001011111001001010100111	5.03223	125.73269	20.525	0.1403
5	0110101001111111001001110101111	4.76111	137.31969	22.360	0.2422
6	0010101000111111001001110110111	5.41215	112.04285	17.791	0.0116
7	0110001001011111001001100101011	5.04689	127.50512	20.666	0.1481
8	0111001001001111001001110111100	5.24772	116.49439	19.064	0.0591
9	0110101001011111001001010000111	5.12779	122.54207	19.968	0.1093
10	0110111001011111011001010100111	5.02100	126.11736	20.592	0.1440
11	0110111001001111001001100101111	5.16981	107.88041	17.603	0.0221
12	0111001001011111001001110111111	4.62364	141.2627	23.096	0.2831
13	0111001001001101111001100101111	5.16293	107.55548	17.618	0.0212
14	0111001001001111001011100101111	5.36243	109.52693	17.918	0.0045
15	0110001001011111001001010100111	5.20566	120.93738	19.614	0.0896
16	0110001001011101111011010100111	5.25974	119.0621	19.291	0.0717
17	0110001001011111110111001011111	5.00666	128.90625	20.906	0.1615
18	0010111001011111001001110111111	5.52277	114.08929	18.139	0.0077
19	0110001001011111001001010100111	5.20566	120.93738	19.614	0.0896

20	0110001001011111001001010100111	5.20566	120.93738	19.614	0.0896
----	---------------------------------	---------	-----------	--------	--------

Se observa que en el cromosoma número 2 se obtiene el mayor peso, se obtienen las potencias de cada grupo y finalmente se tiene:

P1	q1	P2	q2	P3	q3	Suma de P	Suma de q
47.7156	7.7317	47.8917	7.9874	18.4819	2.4194	114.0892	18.1385

En este caso se utilizan las verdaderas funciones sin linealizar.