

### **FE DE ERRATAS:**

Se digitalizó el trabajo de grado de acuerdo al contenido presentado por el autor, en este trabajo de grado se han detectado algunas inconsistencias en la paginación.

**PROPUESTA DE ALGORITMOS GENÉTICOS DE  
CROMOSOMA VARIABLE Y BÚSQUEDA  
DE GAUSS.**

**FABIÁN ALEXANDER CONTRERAS BASTOS**

UNIVERSIDAD INDUSTRIAL DE SANTANDER		No Clasificación
<b>BIBLIOTECA</b>		<b>5</b> <b>11146</b>
No Adquisición	Fecha Recibo	
	<b>19 JUL. 2002</b>	
No Inventario	Precio	Dpto Solicitante
<b>103353</b>		

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECAICAS  
INGENIERÍA DE SISTEMAS  
2002**

**PROPUESTA DE ALGORITMOS GENÉTICOS DE  
CROMOSOMA VARIABLE Y BÚSQUEDA  
DE GAUSS.**

**FABIÁN ALEXANDER CONTRERAS BASTOS**

**Trabajo de Grado para optar al título de  
Ingeniero de Sistemas**

**DIRECTOR  
FERNANDO RUIZ DIAZ  
Ingeniero de Sistemas**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECHANICAS  
INGENIERÍA DE SISTEMAS  
2002**

Nota de aceptación

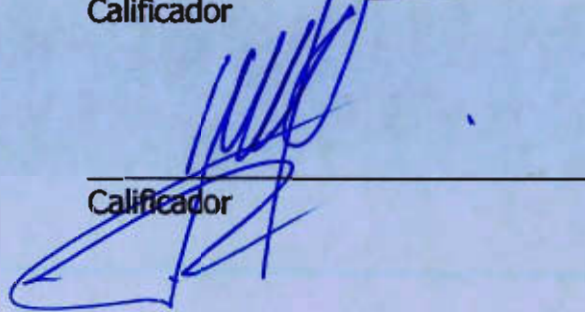
---

---

---



Calificador



Calificador

BIBLIOTECA UIS

Bucaramanga, 22 de Febrero de 2002

Este proyecto de grado se los dedico a  
mis Padres Rodolfo e Ilba,  
a mis Hermanos Juan Carlos y Sergio  
a toda mi familia y  
a mi Novia Claudia Marcela.

## **AGRADECIMIENTOS**

Expreso mis agradecimientos a:

Mis Padres Rodolfo e Ilba, por su apoyo, colaboración, paciencia y comprensión, porque sin ellos no hubiera podido lograrlo.

Mis Hermanos Juan Carlos y Sergio, por estar ahí siempre y ayudarme de una u otra manera.

Mi prima Patricia Díaz, por sus consejos y su preocupación.

Mi Primo Carlos Felipe Reyes, por su ayuda y colaboración a lo largo de mi carrera.

A mi familia, que de una u otra forma estuvo pendiente de mi trabajo.

Mi novia Claudia Marcela, por estar ahí siempre cuando la necesité y darme apoyo en esos momentos difíciles.

Fernando Ruiz Díaz, Ingeniero de Sistemas y Director del Proyecto, por sus ideas y orientaciones a lo largo de la investigación.

Fernando Calvete, Ingeniero de Petróleos, por su colaboración y orientación, para realizar el módulo de la ecuación de estado.

Mis amigos Juan Pablo, Pablo Antonio, Carlos Francisco y Andrés que estuvieron colaborándome y alentándome durante todo el proyecto.

Sandy Moncada, Cesar García y Juan Carlos Moreno, Ingenieros de Sistemas por su colaboración y ayuda para realizar el módulo del Solver.

Los amigos que de una u otra forma me colaboraron durante el proyecto. ✓

Todos aquellos que directa o indirectamente ayudaron para la finalización de la investigación.

## CONTENIDO

	pág.
1. INTRODUCCIÓN	1
2. PRESENTACIÓN	2
2.1 ANTECEDENTES	2
2.2 OBJETIVOS DEL PROYECTO	3
2.2.1 Objetivo General	3
2.2.2 Objetivos Específicos	3
2.3 DESCRIPCIÓN DE LOS ALCANCES DEL PROYECTO	4
2.4 CONTENIDO DEL INFORME	5
3. COMPUTACIÓN EVOLUTIVA (CE)	6
3.1 INTRODUCCIÓN	6
3.2 BASES BIOLÓGICAS	6
3.3 ALGORITMO EVOLUTIVO	8
3.3.1 Esquema de un Algoritmo Evolutivo	9
3.3.2 División de los Algoritmos Evolutivos	10
3.3.2.1 Programación Evolutiva	10
3.3.2.2 Estrategias de Evolución	11
3.3.2.3 Algoritmos Genéticos	11
3.3.3 Subdivisión de Los Algoritmos Evolutivos	12

3.3.3.1	Programas de Evolución	12
3.3.3.2	Programación Genética	12
3.4	COMPONENTES COMUNES EN LOS ALGORITMOS DE LA CE	13
3.5	RELACIÓN CON OTRAS TÉCNICAS DE BÚSQUEDA	13
4.	ALGORITMOS GENÉTICOS TRADICIONALES	15
4.1	INTRODUCCIÓN	15
4.2	DEFINICIÓN	17
4.3	DEFINICIONES BÁSICAS	19
4.3.1	Bit	19
4.3.2	Gen	19
4.3.3	Cromosoma	19
4.3.4	Población	19
4.3.5	Función de Aptitud	19
4.3.6	Función Objetivo	20
4.3.7	Espacio de Búsqueda	20
4.3.8	Codificación	20
4.3.8.1	Codificación Binaria	22
4.3.8.2	Codificación Entera	22
4.3.9	Convergencia Prematura	22
4.3.10	Finalización Lenta	23
4.4	OPERADORES	23
4.4.1	Generación	23

4.4.2	Selección	24
4.4.2.1	Técnicas de Selección	24
4.4.2.1.1	Selección Directa	25
4.4.2.1.2	Selección Aleatoria	25
4.4.2.1.2.1	Selección Equiprobable	25
4.4.2.1.2.2	Selección Estocástica	25
4.4.2.1.2.2.1	Selección por Sorteo	25
4.4.2.1.2.2.2	Selección por Torneo	25
4.4.2.1.2.2.3	Selección por Ranking	26
4.4.2.1.2.2.4	Selección por Ruleta	26
4.4.3	Cruce	27
4.4.3.1	Técnicas de Cruce	28
4.4.3.1.1	Cruce Básico	28
4.4.3.1.2	Cruce Multipunto	28
4.4.3.1.3	Cruce Segmentado	29
4.4.3.1.4	Cruce Uniforme	29
4.4.3.1.5	Cruces para Permutaciones	29
4.4.3.1.5.1	Cruce de Mapeamiento Parcial	29
4.4.3.1.5.2	Cruce de Orden	29
4.4.3.1.5.3	Cruce de Ciclo	29
4.4.4	Mutación	30
4.4.4.1	Técnicas de Mutación	30
4.4.4.1.1	Mutación de un Bit	30
4.4.4.1.2	Mutación Multibit	30

4.4.4.1.3 Mutación Multigen	31
4.4.4.1.4 Mutación de Intercambio	31
4.5 FUNCIONAMIENTO DE UN ALGORITMO GENÉTICO	31
4.6 TIPOS ALGORITMOS GENETICOS	32
4.6.1 Algoritmos Genéticos Generacionales	32
4.6.2 Algoritmos Genéticos de Estado Fijo	33
4.6.3 Algoritmos Genéticos Paralelos	33
4.6.3.1 Modelo de Islas	34
4.6.3.2 Modelo Celular	34
4.7 VENTAJAS Y DESVENTAJAS DE LOS ALGORITMOS GENETICOS	35
4.8 FUNDAMENTOS MATEMÁTICOS DE OPERACIÓN DE UN ALGORITMO GENÉTICO	36
4.8.1 Esquema	36
4.8.2 Propiedades de los Esquemas	37
4.8.2.1 Orden	37
4.8.2.2 Longitud de Definición	38
4.8.3 El Teorema de los Esquemas	38
4.8.4 Visualización de un Esquemas con Hiperplanos	43
4.8.5 Critica al Teorema de los Esquemas	43
4.9 AREAS DE APLICACIÓN DE LOS ALGORITMOS GENÉTICOS	44
4.9.1 Tipos de Aplicaciones de los Algoritmos Genéticos	45
4.9.2 Algoritmos Genéticos y el Aprendizaje Supervisado	47
4.9.2.1 Método Michigan	49
4.9.2.2 Método Pittsburg	52

5.	MEJORAS PROPUESTAS A LOS ALGORITMOS GENÉTICOS TRADICIONALES	54
5.1	CROMOSOMA VARIABLE	60
5.1.1	Intervalo Variable	61
5.2	GENERACIÓN GAUSSIANA O NORMAL	62
5.3	MUTACIÓN DIRIGIDA	64
5.3.1	Mutación de los genes menos significativos con factor de Mutación	65
5.3.2	Mutación por Intervalo	66
5.4	SELECCIÓN POR PROMEDIO	67
5.5	CRUCE POR VALOR PONDERADO DE LA APTITUD	68
6.	PRUEBA DE LAS MEJORAS PROPUESTAS A LOS ALGORITMOS GENÉTICOS TRADICIONALES	69
6.1	DEFINICIÓN DEL PROBLEMA ESCOGIDO PARA REALIZAR LAS PRUEBAS	69
6.2	APLICACIÓN DESARROLLADA PARA REALIZAR LA COMPARACIÓN ENTRE LOS ALGORITMOS GENÉTICOS CLÁSICOS Y LOS MEJORADOS	69
6.2.1	Funcionamiento de la Aplicación	70
6.2.1.1	Ingreso de Datos	70
6.2.1.2	Tipo de Algoritmo	71
6.2.1.2.1	Algoritmo Genético Clásico	72
6.2.1.2.2	Algoritmo Genético Sub_Mejorado	72
6.2.1.2.3	Algoritmo Genético Mejorado	73
6.2.1.2.4	Mínimos Cuadrados	74
6.2.1.3	Modo de Operación	74
6.2.1.3.1	Manual	74
6.2.1.3.2	Semi-Automático	75
6.2.1.3.3	Automático	76

6.2.1.4	Módulo de Comparación de Algoritmos	76
6.2.1.5	Barra de Opciones	77
6.3	RESULTADOS OBTENIDOS AL REALIZAR LA COMPARACIÓN ENTRE ALGORITMOS	78
6.4	ANÁLISIS DE RESULTADOS	79
7.	USO DE LOS ALGORITMOS GENÉTICOS MEJORADOS	84
7.1	CREACIÓN DE UN SOLVER	84
7.1.1	Definición del Problema	84
7.1.2	Implementación de la Aplicación	85
7.1.2.1	Ingreso de la Ecuación	85
7.1.2.2	Ingreso de Valores a las variables	87
7.1.2.3	Respuesta	88
7.1.3	Resultados	95
7.2	AJUSTE DE LOS PARÁMETROS DE LA ECUACIÓN DE ESTADO DE PENG-ROBINSON A UN PROBLEMA ESPECÍFICO	91
7.2.1	Marco Teórico	91
7.2.2	Definición del Problema	93
7.2.3	Implementación de la Aplicación	93
7.2.4	Resultados	94
8.	CONCLUSIONES	97
9.	RECOMENDACIONES	100
	BIBLIOGRAFÍA	102
	ANEXOS	109

## LISTADO DE TABLAS

		pág.
Tabla 1.	Ejemplo de Codificación de Reglas.	55
Tabla 2.	Ejemplo de Codificación de Parámetros.	58
Tabla 3.	Recopilación de datos de la comparación entre los Algoritmos.	81
Tabla 4.	% de Mejoramiento en cada conjunto de puntos.	83
Tabla 5.	% de Mejoramiento del AG Mejorado y los otros dos AG.	83
Tabla 6.	Comparación entre Herramientas Solver.	90
Tabla 7.	Composición del Gas.	94
Tabla 8.	Comparación Estadística entre Metodologías.	96
Tabla 9.	Datos Obtenidos por los dos métodos de la Aplicación.	96

## LISTADO DE FIGURAS

	pág.
Figura 1. Esquema de un Algoritmo Evolutivo.	9
Figura 2. Posible Clasificación de las Técnicas de Búsqueda.	14
Figura 3. Selección por Ruleta.	27
Figura 4. Mutación de un Bit.	30
Figura 5. Esquema de Funcionamiento de un AG.	32
Figura 6. Esquema de longitud 3.	43
Figura 7. Ejemplo de posibles Valores para un cromosoma de 3 bits.	58
Figura 8. Intervalo de $(-1, 2)$ con cromosoma de Longitud 4.	63
Figura 9. Intervalo Variable.	62
Figura 10. Generación Uniforme.	63
Figura 11. Generación Gaussiana.	63
Figura 12. Ejemplo Mutación Dirigida.	66
Figura 13. Ingreso de Datos.	71
Figura 14. Cargar puntos desde archivo.	71
Figura 15. Menú de Algoritmos.	74
Figura 16. Modo de Operación Manual para un Algoritmo Genético Clásico.	75
Figura 17. Modo de Operación Semi-Automático para un Algoritmo Genético Mejorado.	75
Figura 18. Modo de Operación Automático para un Algoritmo Genético Sub_Mejorado.	76
Figura 19. Selección de Comparar Algoritmos.	77

Figura 20.	Resultados Obtenidos de Comparar los tres Algoritmos Genéticos.	78
Figura 21.	Barra de Opciones.	78
Figura 22.	Equation Solver.	86
Figura 23.	Ingreso de la Ecuación manualmente.	86
Figura 24.	Cargar Formula desde Archivo.	87
Figura 25.	Ingreso de Valores a la Variables.	87
Figura 26.	Búsqueda del Intervalo de respuesta.	89
Figura 27.	Mensaje de que no se encontró la respuesta.	89
Figura 28.	Presentación de Resultados.	96
Figura 29.	Comparación Gráfica y Estadística de los Resultados.	96

## LISTADO DE ECUACIONES

	pág.
Ecuación 1. Cálculo de los valores codificados en código binario.	22
Ecuación 2. Formula para el cálculo de las probabilidades del Ranking Lineal.	26
Ecuación 3. Esquema H en la población $A(t+1)$ .	39
Ecuación 4. Desempeño promedio de toda la población.	39
Ecuación 5. Otra Forma Esquema H en la población $t+1$ .	39
Ecuación 6. Esquema H en la Población $t+1$ con constante c.	40
Ecuación 7. Esquema H en la Población t.	40
Ecuación 8. Probabilidad de supervivencia de un esquema.	41
Ecuación 9. Probabilidad de supervivencia con probabilidad de cruce.	41
Ecuación 10. Efecto combinado de la Selección y el cruce.	42
Ecuación 11. Probabilidad de supervivencia con probabilidad de mutación.	42
Ecuación 12. Probabilidad de supervivencia si $P_m \ll 1$ .	42
Ecuación 13. Teorema del Esquema.	42
Ecuación 14. Paso con codificación binaria, en el intervalo (a, b).	60
Ecuación 15. Nuevo valor al Aplicar Mutación dirigida.	65
Ecuación 16. Error entre Aptitudes.	65
Ecuación 17. Factor de Mutación.	66
Ecuación 18. Promedio de las Aptitudes.	67
Ecuación 19. Desviación Estándar.	67
Ecuación 20. Factor de Selección.	67

Ecuación 21. Cruce Ponderado.	68
Ecuación 22. Variable T que no se puede despejar de la ecuación.	85
Ecuación 23. Formato de las formulas Equation Solver	87
Ecuación 24. EOS para Gases Ideales.	91
Ecuación 25. Ecuación de Peng-Robinson.	91
Ecuación 26. Expresión para el cálculo del parámetro a.	92
Ecuación 27. Expresión para el cálculo del parámetro b.	92

## **LISTADO DE ANEXOS**

	pág.
Anexo A EJEMPLOS.	115
Anexo B APLICACIONES DE LOS ALGORITMOS GENÉTICOS.	126
Anexo C PUNTOS UTILIZADOS PARA REALIZAR LAS PRUEBAS DE LAS MEJORAS PROPUESTAS.	129
Anexo D ARTICULO ECUACIÓN DE ESTADO.	130

**TITULO**

PROPUESTA DE ALGORITMOS GENÉTICOS DE CROMOSOMA VARIABLE Y BÚSQUEDA DE GAUSS.

**AUTOR**

Fabián Alexander Contreras Bastos.\*\*

**PALABRAS CLAVES**

Computación Evolutiva, Algoritmos Genéticos, Cromosoma Variable, Búsqueda de Gauss, Mutación Dirigida.

**DESCRIPCIÓN**

Los Algoritmos Genéticos (AG) son herramientas poderosas en la búsqueda de soluciones y optimización de diversos problemas.

Los AG tienen su origen en la teoría de la evolución de Darwin, que plantea que sólo sobrevivirán aquellos organismos que sean los más aptos, o los que estén mejor adaptados. Se les llama genéticos porque realizan una analogía genética, debido a que se usan términos como cromosomas, genes y operadores como selección, cruzamiento, mutación entre otras.

Con la realización de este proyecto se pretende introducir mejoras a los AG existentes para minimizar los tiempos de proceso, y conseguir una mayor precisión en la respuesta; por medio de cromosomas de longitud variable, generaciones secundarias de la población utilizando la distribución de Gauss y mutación dirigida para afinación de la respuesta. Para realizar las pruebas a dichas mejoras se toma como base el problema para encontrar los parámetros de la ecuación de una recta que mejor se ajuste a un conjunto de puntos determinados.

Los resultados obtenidos mediante la aplicación de estas mejoras, indican una disminución del 20 % en los tiempos de proceso, una disminución del 70% en el número de poblaciones utilizadas para alcanzar la respuesta y en todas las pruebas que se realizan el error que se obtiene es menor.

Se desarrollan dos nuevas metodologías con el enfoque de los AG para resolver dos problemas de ingeniería, uno el de despejar numéricamente una variable de una ecuación y el otro en el ajuste de los parámetros de la ecuación de Peng-Robinson.

La aplicación de estas mejoras produce resultados alentadores para continuar con la investigación en el campo de mejoras a los Algoritmos Genéticos. Siendo de gran utilidad en áreas como Medicina, Biología Molecular, Robótica, Criminología entre otras.

---

\* Proyecto de Grado.

\*\* Facultad de Ciencias Físico-Mecánicas - Escuela de Ingeniería de Sistemas  
Director: Fernando Ruiz.

**TITLE**

Proposal of Genetic Algorithms of Variable Chromosomes and Gauss' Searching

**AUTHOR**

Fabián Alexander Contreras Bastos\*\*

**KEY WORDS**

Evolutionary Computation, Genetic Algorithms, Variable Chromosomes, Gauss' Searching, Directed Mutation and Crossing.

**DESCRIPTION**

The Genetic Algorithms (GAs) are powerful tools which are used in searching solutions and optimization of many problems.

The GAs have their origin in the Darwin's evolution theory which establishes that only those organisms most capable or better adapted survive. These algorithms are called genetic because of they work as a genetic analogy using terms such as chromosomes, genes and operators such as selections, crossing and mutation among others.

The main objective of executing the present project is to introduce improvements in the existing GAs. The improvements lead to maximization of the process time and achievement of a better precision in the answers. This is reached by working of chromosomes of variable length, secondary generation of populations by using Gauss distribution and also by working with mutations directed to get a refining answer.

In order to carry out the tests of the mentioned improvements, it is taken the problem of finding out the equation's parameters of a straight line which is best fitted to a group of points .

The results obtained by means of these improvements indicate a diminution of 20% in the process time and a decreasing of 70% in the number of populations used to get the answer.

In addition, new methodologies by employing GAs to solve two engineering problems were developed. In the first one, the value of a variable was found out in an equation, in the second one , the parameters of the Peng-Robinson equation were adjusted.

The application of the improved GAs gave hopeful results to continue searching in this field, specially in areas such as Medicine, Molecular Biology, Robotic, Criminology and others.

---

\* Final Degree Project

\*\* School of Computers Science. Faculty of Physical and Mechanical Engineering.  
Supervisor: Fernando Ruiz Díaz

## 1 INTRODUCCIÓN

Este proyecto de grado presenta una introducción a la Computación Evolutiva que son técnicas heurísticas basadas en principios de la evolución natural que permiten encontrar soluciones a problemas donde su complejidad no permite el tratamiento por métodos tradicionales. Aquí se describe una rama de la Computación Evolutiva relativamente nueva: Los Algoritmos Genéticos.

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden ser utilizados para implementar búsquedas y problemas de optimización. Ellos están basados en los procesos genéticos de organismos biológicos como la selección, cruzamiento y mutación, codificando una posible solución a un problema en un "*cromosoma*" compuesto por una cadena de bits o caracteres.

Este trabajo propone ciertas mejoras a los Algoritmos Genéticos existentes como utilización de un cromosoma variable, mutación para la afinación de la solución óptima entre otras. Estos nuevos algoritmos se aplicarán para la creación de un Solver, que es una nueva área de aplicación de los Algoritmos Genéticos.

Este trabajo será de utilidad a todos los interesados en conocer acerca de los Algoritmos Genéticos y sus aplicaciones, además de sus posibles campos de acción.

## **2 PRESENTACIÓN**

### **2.1 ANTECEDENTES**

La búsqueda de soluciones y la optimización de problemas en el contexto de ingeniería significa encontrar soluciones que satisfagan de una manera óptima los requisitos de nuestro problema.

Existen diversas técnicas de búsqueda tradicionales (Basadas en Cálculo, Métodos Enumerativos y Aleatorios) que sirven para encontrar soluciones óptimas a problemas específicos que se presentan en casos especiales; como por ejemplo métodos iterativos para encontrar el valor de una variable dentro de una ecuación, mínimos cuadrados para encontrar los parámetros que mejor se ajusten a una serie de puntos.

Pero en problemas donde no es posible utilizar dichas técnicas, surgen nuevas alternativas de solución como las técnicas heurísticas y entre estas los Algoritmos Genéticos. Dado que estas nuevas técnicas presentan una mayor robustez que las técnicas tradicionales también se aplican a problemas donde éstas encuentran solución a un problema.

Los Algoritmos Genéticos son utilizados en Diseño de redes neuronales, trazado de redes de comunicación, sistemas dinámicos no lineales y modelos de seguridad internacional, estrategias de adquisición, conformación de análisis de DNA, Automatización de la

afinación de parámetros para procesamiento de información de un sonar, planes de inversión, etc.

Debido a su gran campo de acción se hace necesario estudiar más a fondo la estructura, funcionamiento y comportamiento de los Algoritmos Genéticos, para proponer mejoras que los hagan más competitivos, robustos y óptimos, para expandir sus aplicaciones.

## **2.2 OBJETIVOS DEL PROYECTO**

**2.2.1 Objetivo General.** Proponer mejoras a los algoritmos genéticos clásicos para la obtención de soluciones en problemas de optimización de modelos matemáticos no lineales, proporcionando derivaciones en el algoritmo clásico, utilizando cromosomas de longitud variables, mutación para afinación de la solución óptima, y búsquedas secundarias por generación de la población siguiendo la distribución de Gauss.

### **2.2.1 Objetivos Específicos.**

- Desarrollar un software, que obtenga soluciones a algunos problemas de optimización de sistemas descritos por modelos matemáticos no lineales aplicando la teoría clásica y una derivación que mejore los tiempos de proceso.
- Aplicar el algoritmo derivado propuesto a un problema práctico.
- Incluir en el algoritmo propuesto las siguientes características:
  - Generación inicial de la población uniforme, y posteriores generaciones usando una distribución probabilística de Gauss.
  - Utilizar la mutación del cromosoma para afinar la mejor solución.

- Aumentar el tamaño del cromosoma para reducir el error o diferencia de la mejor solución encontrada.

### 2.3 DESCRIPCIÓN DE LOS ALCANCES DEL PROYECTO

- Se elaboró una Monografía acerca de la Computación Evolutiva, de los fundamentos y aplicaciones de los Algoritmos Genéticos Tradicionales.
- Se elaboró una Monografía acerca de las mejoras propuestas a los Algoritmos Genéticos Tradicionales.
- Se desarrolló un software para encontrar los parámetros de la ecuación de una recta que mejor se ajuste a una serie de puntos, dicho software mostrará los resultados obtenidos al utilizar un algoritmo genético tradicional y los resultados aplicando un algoritmo con las derivaciones propuestas.
- Se hizo una comparación de los resultados experimentales obtenidos del problema del ajuste de los parámetros de la recta entre los algoritmos genéticos tradicionales y los algoritmos con las derivaciones propuestas.
- Se desarrolló un programa como ejemplo de aplicación de los algoritmos genéticos con las derivaciones propuestas:
  - - Desarrollo de un Solver
    - Implementación de una Aplicación para el Ajuste de los Parámetros de la ecuación de estado de Peng\_Robinson de un problema específico.

## 2.4 CONTENIDO DEL INFORME

El presente documento está organizado de la siguiente manera:

Los capítulos 3 y 4 comprenden la monografía elaborada acerca de los Algoritmos Genéticos Tradicionales, esta se halla organizada como sigue: La Primera parte (Capítulo 3) Computación Evolutiva, la segunda parte (Capítulo 4) Algoritmos Genéticos Tradicionales.

En los tres siguientes capítulos (Capítulos 5, 6 y 7) se encuentra el grueso del trabajo de grado en cuanto a las derivaciones propuestas a los Algoritmos Genéticos tradicionales. En el capítulo 5 se exponen las mejoras como la generación con distribución gaussiana, cromosoma variable y mutación dirigida y el porqué de ellas. En el capítulo 6 se encuentra un análisis comparativo entre los Algoritmos Tradicionales y los Mejorados en cuanto a tiempos y precisión. En el capítulo 7 se expone el desarrollo de las dos aplicaciones donde se utilizaron los algoritmos genéticos mejorados.

Los últimos capítulos contienen las conclusiones del proyecto, recomendaciones y directrices futuras.

## **3 COMPUTACIÓN EVOLUTIVA**

### **3.1 INTRODUCCION**

El término Computación Evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas basadas en los principios de la evolución natural. Estas técnicas pueden ser clasificadas en tres grandes categorías:

- Algoritmos Genéticos
- Estrategias Evolutivas
- Programación Evolutiva

La Computación Evolutiva, es un enfoque alternativo para abordar problemas complejos de búsqueda y aprendizaje a través de modelos computacionales de procesos evolutivos. Las implementaciones concretas de tales modelos se conocen como algoritmos evolutivos. El propósito genérico de los algoritmos evolutivos consiste en guiar una búsqueda estocástica haciendo evolucionar a un conjunto de estructuras y seleccionando de modo iterativo las más adecuadas.

### **3.2 BASES BIOLÓGICAS**

La computación evolutiva parte de un hecho observado en la naturaleza: los organismos vivos poseen una destreza consumada en la resolución de los problemas que se les presentan, y obtienen sus habilidades, casi sin proponérselo, a través del mecanismo de la

evolución natural. La evolución se produce, en casi todos los organismos, como consecuencia de dos procesos primarios:

- La selección natural determina qué miembros de la población sobrevivirán hasta reproducirse (es un proceso sencillo: cuando un organismo falla una prueba de idoneidad, muere).
- La reproducción sexual garantiza la mezcla y recombinación de genes en la descendencia de un organismo.

Además se basa en los principios propuestos en la *Teoría de la Selección Natural* de Darwin (1859) y los trabajos sobre *Herencia Genética* de Mendel (1865), y los cuales pueden resumirse en los siguientes puntos:

- La evolución es un proceso que no opera directamente sobre organismos, sino sobre cromosomas. Éstos son unos instrumentos orgánicos mediante los cuales se codifica la estructura de un ser vivo, por ejemplo, un ser vivo se crea mediante la decodificación de un conjunto de cromosomas. Estos cromosomas (más concretamente, la información que contienen) pasan de una generación a otra mediante los procesos reproductivos.
- El proceso evolutivo propiamente dicho tiene lugar durante la etapa de reproducción. En la naturaleza existe una gran cantidad de mecanismos reproductivos. Los más básicos son los de *mutación* (que modelan la variabilidad del material genético de los individuos) y los de *recombinación* (que combinan los materiales genéticos de los padres para producir el de los descendientes).

- La selección natural es el mecanismo que permite relacionar los cromosomas con la eficiencia de las entidades que representan, provocando que aquellos organismos eficaces y adaptados al medio se reproduzcan con mayor probabilidad que aquellos que no lo están, ocasionando de esta manera la extinción de los organismos inadaptados o poco eficientes.

Las condiciones que determinan un proceso evolutivo en la naturaleza son:

- Existe una población de entidades o individuos con capacidad para reproducirse.
- Existe alguna diferencia o variedad entre estos individuos.
- Esta variedad está relacionada con algunas diferencias en la habilidad para sobrevivir.<sup>1</sup>

### 3.3 ALGORITMO EVOLUTIVO

El término *algoritmo evolutivo* se usa para designar a un conjunto de técnicas que basan su funcionamiento en metáforas de procesos biológicos. Esta definición es bastante amplia ya que no existe una caracterización totalmente consensuada de los mismos.

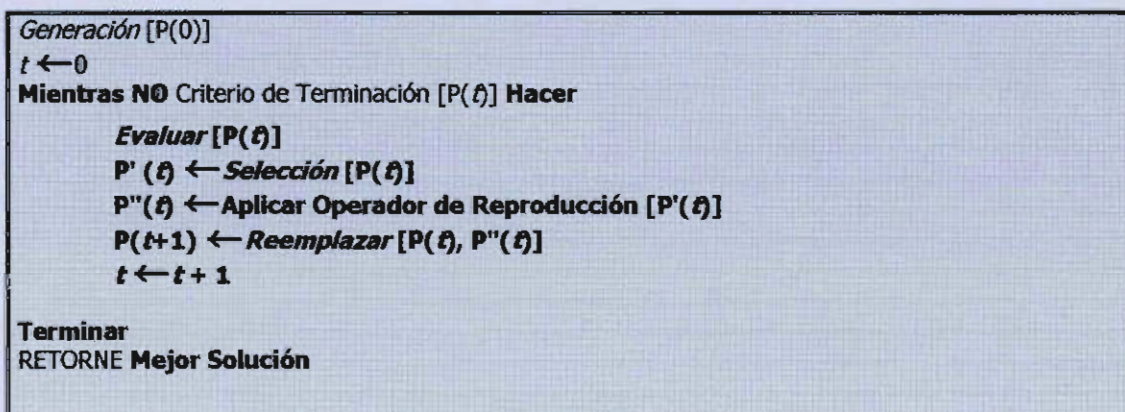
Estos pueden resumirse en la siguiente definición dada por Jones (1995):

*«El algoritmo mantiene un conjunto de soluciones potenciales a un problema. Dichas soluciones son usadas para producir nuevas soluciones potenciales mediante la aplicación de una serie de operadores. Dichos operadores actúan sobre algunas soluciones que han sido seleccionadas por su bondad con respecto al problema atacado. Este proceso se repite hasta que se alcanza un cierto criterio de terminación.»*

**3.3.1 Esquema de un Algoritmo Evolutivo.** Un algoritmo evolutivo es un proceso estocástico e iterativo que opera sobre un conjunto  $P$  de *individuos* (población), cada uno de los cuales contiene uno o más  *cromosomas*. Dichos cromosomas permiten que cada individuo represente una posible solución al problema que se está considerando. Un proceso de codificación/decodificación ( $\rho$ ) permite obtener la solución que los cromosomas de cada individuo contienen. Inicialmente, esta población es generada aleatoriamente o con la ayuda de algún heurístico de construcción.

Cada uno de los individuos de la población recibe, a través de una función de adecuación, una medida de su bondad con respecto al problema que se desea resolver. Este valor es empleado por el algoritmo para guiar la búsqueda. El proceso se esquematiza en la figura 1. Como puede apreciarse, el algoritmo está estructurado en tres fases principales que se ejecutan de manera circular, selección, reproducción y reemplazo, las cuales se llevan a cabo de manera repetitiva. Cada una de las iteraciones del algoritmo se denomina *ciclo reproductivo básico* o *generación*.

Este proceso se realiza hasta que se alcanza un determinado criterio de terminación, verbigracia, un número determinado de iteraciones.



**Figura 1. Esquema de un Algoritmo Evolutivo**

<sup>1</sup> Información recopilada de las ideas del Ingeniero Informático Manuel de la Herrán Gascón Universidad de Deusto.

Durante la fase de selección se crea una población temporal  $P'$  en la que aquellos individuos más aptos (los correspondientes a las mejores soluciones contenidas en la población) estarán representados un mayor número de veces que los poco aptos (*principio de selección natural*). A los individuos contenidos en esta población temporal les son aplicados diferentes operadores de cambio (también denominados operadores reproductivos) en la fase de reproducción. El objetivo de esta fase es producir individuos con nuevas características, idealmente mejores (*principio de adaptación*). Finalmente, durante la fase de reemplazo, se substituyen individuos de la población original por los nuevos individuos creados. Este reemplazo afecta a los peores individuos y tiende a conservar los mejores (*supervivencia de los más adaptados*). Todo este proceso descrito se repite hasta que se cumple un determinado criterio de terminación (normalmente al completar un cierto número de iteraciones).

Obsérvese cómo este algoritmo descrito establece un compromiso entre la explotación de las buenas soluciones (fase de selección), y la exploración de nuevas zonas del espacio de búsqueda (fase de reproducción), apoyado en el hecho de que el mecanismo de reemplazo puede permitir la aceptación de nuevas soluciones que no proporcionen una mejora inmediata sobre las ya existentes.

### **3.3.2 División de los Algoritmos Evolutivos.**

**3.3.2.1 Programación Evolutiva.** Esta familia de algoritmos tiene su origen en el trabajo de Fogel (1966), y ponen un especial énfasis en la adaptación de los individuos más que en la evolución del material genético de éstos. Ello implica una visión mucho más abstracta del proceso, en la cual se modifica directamente el comportamiento de los individuos en lugar de trabajar sobre sus genes. Dicho comportamiento se modela

mediante estructuras de datos relativamente complejas como pueden ser autómatas finitos.

Tradicionalmente, estas técnicas emplean mecanismos de reproducción asexual y técnicas de selección mediante competición directa entre individuos.

**3.3.2.2 Estrategias de Evolución.** Su objetivo inicial era servir de herramienta para optimización de parámetros en problemas de ingeniería (Rechenberg, 1973). Debido a este objetivo primordial, estas técnicas se caracterizan por manejar vectores de números codificados en punto flotante (sí bien existen versiones de las mismas que se aplican a problemas discretos). Al igual que la programación evolutiva con la que se halla estrechamente emparentada, una estrategia de evolución basa su funcionamiento en el empleo de un operador de reproducción asexual o de mutación. Dicho operador está especialmente diseñado para trabajar con números flotantes, e incluye un complejo mecanismo de autoadaptación, en virtud del cual se consigue optimizar la dirección preferente de búsqueda.

**3.3.2.3 Algoritmos Genéticos.** Estas técnicas son probablemente el representante más conocido de los algoritmos evolutivos, y aquellas cuyo uso está más extendido. Fueron concebidas originalmente por John Holland y descritas en el ya clásico «*Adaptation in Natural and Artificial Systems*» (Holland, 1975). Este texto ha tenido una gran trascendencia en el posterior desarrollo de estas técnicas ya que los mecanismos en él descritos han sido tomados durante largo tiempo como auténticos dogmas.

La principal característica de los algoritmos genéticos es el uso de un operador de recombinación o cruce como mecanismo principal de búsqueda. Este operador debe

recombinar los cromosomas de los padres para construir descendientes que posean características de ambos. La utilidad de este operador se fundamenta en la suposición de que diferentes partes de la solución óptima pueden ser descubiertas independientemente y luego ser combinadas para formar mejores soluciones. Adicionalmente, emplean un operador de mutación cuyo uso se considera importante como responsable del mantenimiento de la diversidad en la población, aunque secundario en relación con el operador de cruce.<sup>2</sup>

**3.3.3 Subdivisión de los Algoritmos Evolutivos.** Estas tres familias (Programación evolutiva, estrategias de Evolución y Algoritmos Genéticos) no han permanecido aisladas y han interactuado frecuentemente, motivo por el cual las fronteras entre ellas son difusas en la actualidad. Fruto de este contacto, han aparecido nuevas variedades de algoritmos evolutivos. En las que sobresale por ejemplo la programación genética.

**3.3.3.1 Programas de Evolución.** Estas técnicas están avaladas por el trabajo de, entre otros, Z. Michalewicz (1992), englobándose bajo esa denominación a los heurísticos que, siguiendo los mismos principios de funcionamiento que los algoritmos genéticos, hacen evolucionar estructuras complejas (Alba y Cotta, 1997).

**3.3.3.2 Programación Genética.** Inspirada en el trabajo de Cramer (1985) y popularizada por Koza (1992), la programación genética es otra variante de los algoritmos genéticos en la que se hace evolucionar estructuras (típicamente árboles) que representan programas de ordenador. El objetivo final es el diseño automático de un programa que resuelva una tarea determinada, expresada a partir de una serie de casos de ejemplo.

---

<sup>2</sup> División propuesta por el profesor Carlos A. Coello, investigador de Lania Ltda.

### 3.4 COMPONENTES COMUNES EN LOS ALGORITMOS DE LA CE

Algunos Algoritmos Evolutivos están compuestos por un conjunto de elementos comunes.

Los elementos comunes más importantes son:

- Una población de N cadenas, por lo general las cadenas son binarias, flotantes o alguna estructura compleja de genes.
- Una función de Aptitud, que evalúa cada cadena de la población. Esta puede ser un simple polinomio o un sistema complejo de evaluación como una red neuronal artificial o algún tipo de sistema simulador.
- Algún mecanismo de selección que simule el proceso de escogencia de las cadenas mejor adaptadas.
- Una Política de Reemplazo para mantener el tamaño de la población constante. Esto es por lo general llevado a cabo reemplazando los peores individuos de la población por otros mejor adaptados.
- Operadores inspirados en la naturaleza para cambiar los individuos a nuevos y mejores individuos. Un tipo de operador común es el de la mutación, otro tipo de estos operadores es el de cruce, pero éste es encontrado en los Algoritmos Genéticos.

### 3.5 RELACIÓN CON OTRAS TÉCNICAS DE BÚSQUEDA

La Localización de este tipo de técnicas (CE), con respecto a procedimientos determinísticos y no determinísticos, se muestra en la figura 2.<sup>3</sup>

---

<sup>3</sup> P. Larragaña Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco.

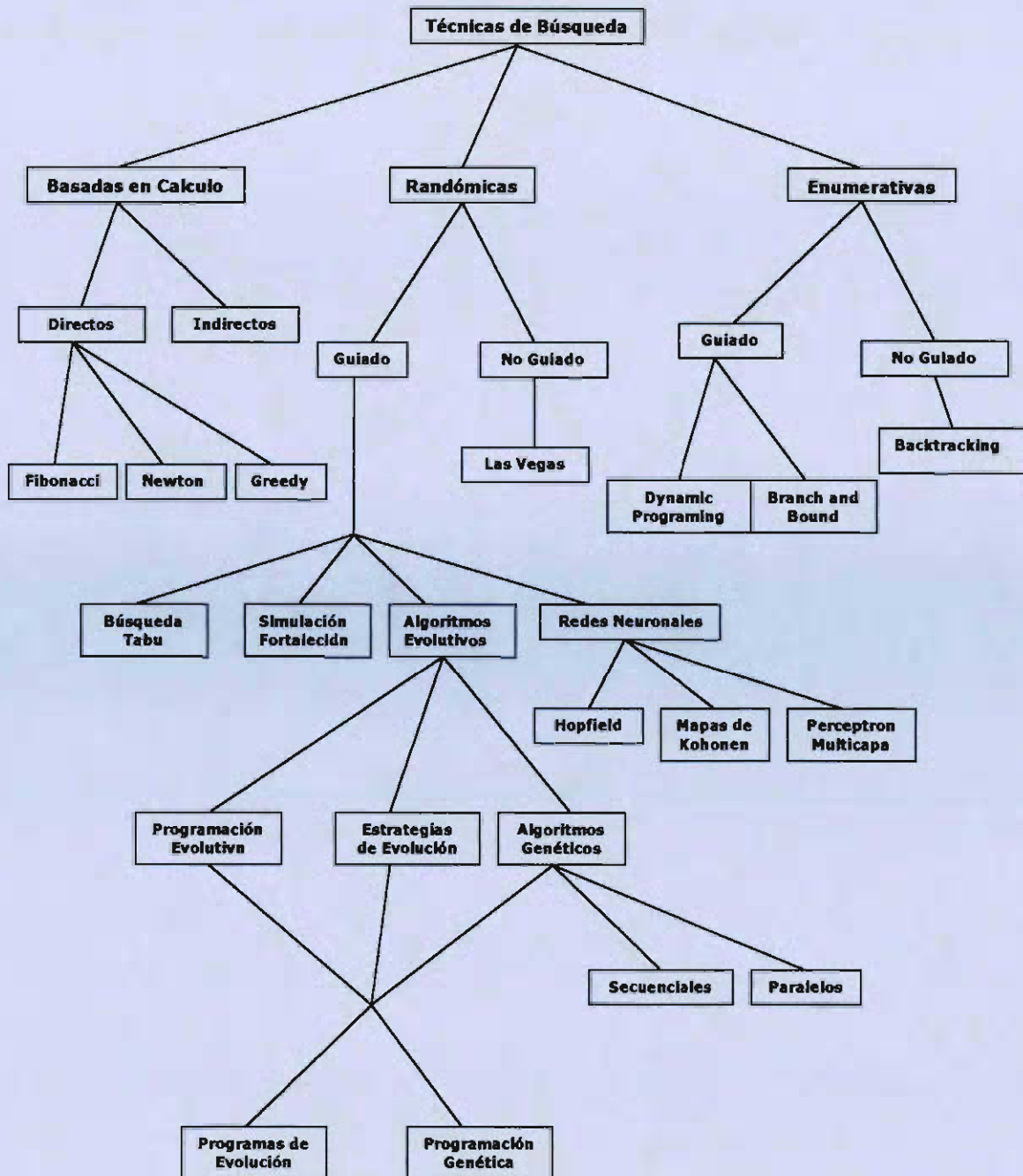


Figura 2. Posible Clasificación de las Técnicas de Búsqueda.

## 4 ALGORITMOS GENÉTICOS TRADICIONALES

### 4.1 INTRODUCCIÓN

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden ser utilizados para implementar búsquedas y problemas de optimización. Ellos están basados en los procesos genéticos de organismos biológicos, codificando una posible solución a un problema en un "*cromosoma*" compuesto por una cadena de bits o caracteres.

Estos cromosomas representan individuos que son llevados a lo largo de varias generaciones, en forma similar a las poblaciones naturales, evolucionando de acuerdo a los principios de selección natural y "*supervivencia*" del más apto, descritos por primera vez por Charles Darwin en su libro "*Origen de las Especies*". Emulando estos procesos, los Algoritmos Genéticos son capaces de "*evolucionar*" soluciones a problemas del mundo real.

En la naturaleza, los individuos compiten entre sí por recursos tales como comida, agua y refugio. Adicionalmente, los animales de la misma especie normalmente antagonizan para obtener una pareja. Aquellos individuos que tengan más éxito tendrán probablemente un número mayor de descendientes, por lo tanto, mayores probabilidades de que sus genes sean propagados a lo largo de sucesivas generaciones. La combinación de características de los padres bien adaptados, en un descendiente, puede producir muchas veces un

nuevo individuo mucho mejor adaptado que cualquiera de sus padres a las características de su medio ambiente.

Los Algoritmos Genéticos utilizan una analogía directa del fenómeno de evolución en la naturaleza. Trabajan con una población de individuos, cada uno representando una posible solución a un problema dado. A cada individuo se le asigna una puntuación de adaptación, dependiendo de que tan buena fue la respuesta al problema. A los más adaptados se les da la oportunidad de reproducirse mediante cruzamientos con otros individuos de la población, produciendo descendientes con características de ambos padres. Los miembros menos adaptados poseen pocas probabilidades de que sean seleccionados para la reproducción, y desaparecen.

Una nueva población de posibles soluciones es generada mediante la selección de los mejores individuos de la generación actual, emparejándolos entre ellos para producir un nuevo conjunto de individuos. Esta nueva generación contiene una proporción más alta de las características poseídas por los mejores miembros de la generación anterior. De esta forma, a lo largo de varias generaciones, las características buenas son difundidas a lo largo de la población mezclándose con otras. Favoreciendo el emparejamiento de los individuos mejor adaptados, es posible recorrer las áreas más prometedoras del espacio de búsqueda. Si el Algoritmo Genético ha sido diseñado correctamente, la población convergerá a una solución óptima o casi óptima al problema.

Los dos procesos que más contribuyen a la evolución son el cruce y la adaptación basada en la selección / reproducción. La mutación también juega un papel significativo pero es de menor relevancia en el funcionamiento de un Algoritmo Genético, aunque la mutación utilizada de una manera correcta puede ser de gran ayuda para la convergencia del

Algoritmo, sin embargo ella no debe ser utilizada demasiado, ya que el Algoritmo Genético se puede convertir en una búsqueda al azar, pero su utilización asegura que ningún punto en el espacio de búsqueda tiene probabilidad 0 de ser examinado.

## 4.2 DEFINICIÓN

Los Algoritmos Genéticos son técnicas de búsqueda que se basan en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales, los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Combina los sobrevivientes mejor adaptados de entre cadenas de estructuras mediante un mecanismo randomizado de intercambio de información. En cada generación, un nuevo conjunto de criaturas artificiales es creado usando la unidad de información correspondiente (bits) y las partes mejor adaptadas de los individuos de la generación anterior.

Aunque en la literatura podemos encontrar definiciones mucho más completas y especializadas.

"Son métodos estocásticos de búsqueda ciega de soluciones cuasi-óptimas. En ellos se mantiene una población que representa un conjunto de posibles soluciones, la cual es sometida a ciertas transformaciones con las que se trata de obtener nuevos candidatos, y un proceso de selección sesgado en favor de los mejores candidatos".

"Los Algoritmos Genéticos son algoritmos de búsqueda basados en los mecanismos de la Selección Natural y la Genética".<sup>4</sup>

---

<sup>4</sup> Definición dada por David Goldberg en su Libro "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Publishing Company, 1989, 421 p.

Una definición bastante completa de lo que es un Algoritmo Genético es la propuesta por Jhon Koza :

“Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (Letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con cierta función matemática que refleja su aptitud”.<sup>5</sup>

Los Algoritmos Genéticos no son la única técnica basada en una analogía de la naturaleza. Por ejemplo, las Redes Neuronales están basadas en el comportamiento de las neuronas en el cerebro. Pueden ser utilizadas en una gran variedad de tareas de clasificación, como reconocimiento de patrones o proceso de imágenes.

El poder de los Algoritmos Genéticos proviene del hecho de que la técnica es robusta, y puede manejar exitosamente un amplio rango de problemas, incluso algunos que son difíciles de resolver por otros métodos. Los Algoritmos Genéticos no garantizan que encontrarán la solución óptima al problema, pero son generalmente buenos encontrando soluciones aceptables a problemas en corto tiempo.

---

<sup>5</sup> Koza, J.R., “Genetic Programming. On the Programming of Computers by Means of Natural Selection”, The Mit Press, 1992, 819 p.

### 4.3 DEFINICIONES BÁSICAS

**4.3.1 Bit.** Es un campo que puede tomar el valor de 0 ó 1 en la mayoría de casos, aunque un bit también puede ser un caracter.

**4.3.2 Gen.** La unión de varios bits conforma un gen.

**4.3.3 Cromosoma.** Es un conjunto de uno ó varios genes, cuya longitud ( $L$ ) depende de la precisión con que se quiera trabajar. Poseen ciertos parámetros que son característicos de cada uno como su valor, su aptitud, porcentaje de aptitud también llamado probabilidad, entre otros. Llamados también individuos (Teoría de Darwin), y es sobre los cuales se aplican los operadores, que serán descritos más adelante.

**4.3.4 Población.** Es un conjunto de  $N$  cromosomas cada uno de tamaño  $L$ . También conocida como Generación. Al igual que los cromosomas poseen parámetros propios como Aptitud de la generación.

**4.3.5 Función de Aptitud.** Es la función que nos determina que tan bueno o tan malo, es un cromosoma (Aptitud del Cromosoma). Parámetro que utilizamos para la selección o no de dicho cromosoma.

En muchos casos, el desarrollo de una función de aptitud involucra hacer una simulación, en otros, la función puede estar basada en el rendimiento y representar sólo una evaluación parcial del problema. Adicionalmente debe ser rápida, ya que hay que aplicarla para cada individuo de cada población en las sucesivas generaciones, por lo cual, gran

parte del tiempo de corrida de un algoritmo genético se emplea en la función de evaluación.

**4.3.6 Función Objetivo.** Es la ecuación o modelo matemático que representa el problema a resolver en forma abstracta y aproximada.

**4.3.7 Espacio de Búsqueda.** Es el espacio donde se encuentran todas las posibles soluciones. Cada punto dentro de este espacio representa una solución factible. Cada solución factible puede ser representada por su valor o aptitud para el problema.

**4.3.8 Codificación.** Las partes que relacionan un Algoritmo Genético con un problema dado son la codificación y la función de aptitud.

Si un problema puede ser representado por un conjunto de parámetros (conocidos como genes), estos pueden ser unidos para formar una cadena de valores (cromosoma), a este proceso se le llama *codificación*. En genética este conjunto representado por un cromosoma en particular es referido como genotipo, este contiene la información necesaria para construir un organismo, conocido como fenotipo. Estos mismos términos se aplican en Algoritmos Genéticos, por ejemplo, si se desea diseñar un puente, el conjunto de parámetros especificando el diseño es el genotipo, y la construcción final es el fenotipo. La adaptación de cada individuo depende de su fenotipo, el cual se puede inferir de su genotipo, es decir, puede calcularse desde el cromosoma utilizando la función de evaluación.

Por ejemplo, si se tiene un problema de maximizar una función de tres variables,  $F(X, Y, Z)$ , se podría representar cada variable por un número binario de 10 bits, obteniéndose un cromosoma de 30 bits de longitud y 3 genes.

Existen varios aspectos relacionados con la codificación de un problema a ser tomados en cuenta en el momento de su realización:

- Se debe utilizar el alfabeto más pequeño posible para representar los parámetros, normalmente se utilizan dígitos binarios.
- Las variables que representan los parámetros del problema deben ser discretizadas para poder representarse con cadenas de bits, hay que utilizar suficiente resolución para asegurar que la salida tiene un nivel de precisión adecuado, se asume que la discretización es representativa de la función objetivo.
- La mayor parte de los problemas tratados con Algoritmos Genéticos son no lineales y muchas veces existen relaciones "ocultas" entre las variables que conforman la solución. Esta interacción es referida como *epítasis*, y es necesario tomarla en cuenta para una representación adecuada del problema.
- El tratamiento de los genotipos inválidos debe ser tomado en cuenta para el diseño de la codificación. Supóngase que se necesitan 1200 valores para representar una variable, esto requiere al menos 11 bits, pero estos codifican un total de 2048 posibilidades, "sobrando" 848 patrones de bits no necesarios. A estos patrones se les puede dar un valor cero de adaptación, ser substituidos por un valor real, o eliminar el cromosoma.

Existen varios tipos de Codificación pero los más usados son la Codificación Binaria y la Codificación Entera.

**4.3.8.1 Codificación Binaria.** También conocida como codificación Gray. Se realiza mapeando linealmente un rango de valores ( $X_i$ ) a números enteros binarios, ó código Gray, de longitud  $L$ .

Ejemplo para  $L = 5$  y un rango de valores entre (2,5) tenemos:

Codificación	$X_i$
0 0 0 0 0	2
0 0 0 0 1	2.096774194
0 0 0 1 0	2.193548387
.	.
.	.
.	.
.	.
1 1 1 1 1	5

Los vectores de  $X_i$  se calculan a partir del código (binario ó Gray) como:

$$X = \frac{X_{\max} - X_{\min}}{2^L - 1} \cdot X_{cod} + X_{\min}$$

**Ecuación 1. Cálculo de los valores codificados en código binario.**

donde:  $X_{cod}$  es el equivalente decimal de un valor de  $X$  codificada en binario ó Gray.

**4.3.8.2 Codificación Entera.** Consiste en asignar números enteros o reales a los parámetros del problema. Esta codificación ha venido tomando fuerza debido a que mejora la eficiencia del Algoritmo ya que no es necesario decodificar cada parámetro para evaluar la función objetivo. Se utiliza menos memoria ya que pueden utilizarse arreglos de variables de punto flotante para representar un cromosoma.

**4.3.9 Convergencia Prematura.** Un problema de los Algoritmos Genéticos dado por una mala formulación del modelo es aquel en el cual los genes de una pocos individuos relativamente bien adaptados, pero no óptimos, pueden rápidamente dominar la

población, causando que converja a un máximo local. Una vez que esto ocurre, la habilidad del modelo para buscar mejores soluciones es eliminada completamente, quedando sólo la mutación como vía de buscar nuevas alternativas, y el algoritmo se convierte en una búsqueda lenta al azar.

Para evitar este problema, es necesario controlar el número de oportunidades reproductivas de cada individuo, tal que, no obtenga ni muy alta o muy baja probabilidad. El efecto es comprimir el rango de adaptación y prevenir que un individuo "superadaptado" tome control rápidamente.

**4.3.10 Finalización Lenta.** Este es un problema contrario al anterior, luego de muchas generaciones, la población habrá convergido, pero no habrá localizado el máximo global. La adaptación promedio será alta y habrá poca diferencia entre el mejor y el individuo promedio, por consiguiente será muy baja la tendencia de la función de adaptación a llevar el algoritmo hacia el máximo. Las mismas técnicas aplicadas en la convergencia prematura son utilizadas en este caso.<sup>6</sup>

## 4.4 OPERADORES

Son funciones que se aplican a los cromosomas o a las Generaciones, como generación de individuos, selección, cruce y mutación.

**4.4.1 Generación.** Es el operador que se encarga de crear aleatoriamente la primera población de individuos.

---

<sup>6</sup> Definiciones y conceptos sacados de la dirección de internet:  
[www.iamnet.com/users/jcontre/genetic/definiciones.htm](http://www.iamnet.com/users/jcontre/genetic/definiciones.htm)

Esto se realiza utilizando un mecanismo de generación de números aleatorios uniformemente distribuidos para crear los  $N$  individuos (cromosomas) iniciales de la primera población.

Existen otros tipos de Generación como la generación extendida donde cada individuo de la población se genera aleatoriamente varias veces, se evalúa el desempeño de cada opción y se toma aquella con mejor desempeño. Otra opción consiste en crear la población inicial con individuos de los que se sabe de ante mano que corresponden a valores cercanos al óptimo.

**4.4.2 Selección.** Simula el proceso de selección natural, en el que el más fuerte tiene mayor capacidad de supervivencia. En los Algoritmos Genéticos la capacidad de supervivencia de un individuo está relacionado con el valor numérico de la función de Aptitud (fitness of good). Como el Algoritmo Genético es un algoritmo de carácter iterativo al igual que los métodos numéricos de optimización, se aplica en cada iteración, sobre una población de  $N$  individuos, el operador de Selección para escoger los mejores  $N$  individuos para la siguiente iteración. Por consiguiente, en los  $N$  individuos escogidos para la siguiente iteración, pueden encontrarse dos o más individuos idénticos debido a que los individuos con baja Aptitud (fitness of bad) tienen menos posibilidades de ser escogidos, en cambio, los que poseen aptitudes altas (fitness of good) tienen más posibilidades de ser seleccionados.

**4.4.2.1 Técnicas de Selección.** Para realizar el proceso de Selección sobre los individuos de una población existen diversos métodos o procedimientos.

**4.4.2.1.1 Selección Directa.** Toma elementos de acuerdo a un criterio objetivo, como por ejemplo "los  $x$  mejores", "los  $x$  peores" o los del tipo "el quinto individuo a partir del último escogido" son empleados con mucha frecuencia cuando se quieren seleccionar dos individuos distintos, y se selecciona el primero por un método aleatorio o estocástico.

**4.4.2.1.2 Selección Aleatoria.** Puede ser realizada por dos procedimientos: Selección Equiprobable o Selección Estocástica.

**4.4.2.1.2.1 Selección Equiprobable.** Todos los individuos (cromosomas) de la Población tienen la misma probabilidad de ser escogidos.

**4.4.2.1.2.2 Selección Estocástica.** La probabilidad de que un individuo de la Población sea escogido depende de una heurística (método Estocástico). Existen distintos procedimientos estocásticos.

**4.4.2.1.2.2.1 Selección por Sorteo.** Cada individuo de la población tiene asignado un rango proporcional - ó inversamente proporcional - a su adaptación. Se escoge un número aleatorio dentro del rango global, y el escogido es aquel que tenga dicho número dentro de su rango. La probabilidad de ser escogido es proporcional/inversamente proporcional al grado de adaptación del individuo.

**4.4.2.1.2.2.2 Selección por Torneo.** Este mecanismo de selección consiste en tomar un subgrupo de  $t$  individuos de la población actual, copiando el mejor de éste subgrupo a la población seleccionada. Frecuentemente se toma  $t=2$  (torneo binario).

Esta técnica tiene la ventaja de que permite un cierto grado de elitismo - el mejor nunca va a morir, y los mejores tienen más probabilidad de reproducirse y de emigrar que los peores - pero sin producir una convergencia genética prematura, si la población es, al menos, un orden de magnitud superior al del número de elementos involucrados en el torneo.

**4.4.2.1.2.2.3 Selección por Ranking.** Este procedimiento de selección fue ideado por Whitley (1987), consistiendo en construir la población seleccionada atendiendo a la ordenación de los valores de adecuación en lugar de a sus valores absolutos, esto es, el mejor individuo se selecciona con probabilidad  $p_1$ , el siguiente con  $p_2$ , etc. Dichas probabilidades pueden ser calculadas de muy diversas maneras. La forma más extendida de hacerlo se denomina *ranking lineal*:

$$P_i = \frac{1}{N} \left[ h^- + (h^+ - h^-) \cdot \frac{i-1}{N-1} \right]$$

**Ecuación 2. Fórmula para el cálculo de las probabilidades del Ranking Lineal.**

Donde  $h^+ + h^- = 2$ . Este mecanismo de selección presenta numerosas ventajas sobre la selección proporcional, como es la transparencia al sentido de la optimización (maximización o minimización), la invariancia a traslaciones (y por lo tanto, ausencia de problemas de escalado) y mantenimiento de una presión selectiva constante a lo largo de la ejecución del algoritmo.

**4.4.2.1.2.2.4 Selección por Ruleta.** Consiste en la creación de una Ruleta, donde la cantidad de divisiones de la misma constituye el número de individuos de la población, y el valor del porcentaje que le corresponde a cada uno constituye el valor de aptitud (fitness) del individuo. Al hacer girar la Ruleta para realizar la Selección es de esperar que aquel

individuo con mejor valor de aptitud (fitness of good) tenga mayor probabilidad de ser escogido del grupo de individuos.

El elemento escogido será aquel en cuyo rango esté el número resultante de sumar el número aleatorio con el resultado total que sirvió para escoger el elemento anterior. En la Figura 3. se muestra un ejemplo de una selección por Ruleta.

**4.4.3 Cruce.** El operador de cruzamiento es aplicado a los individuos que fueron seleccionados de la población anterior, en donde se toma aleatoriamente un punto de corte que será aplicado a la pareja de cromosomas de los individuos seleccionados.

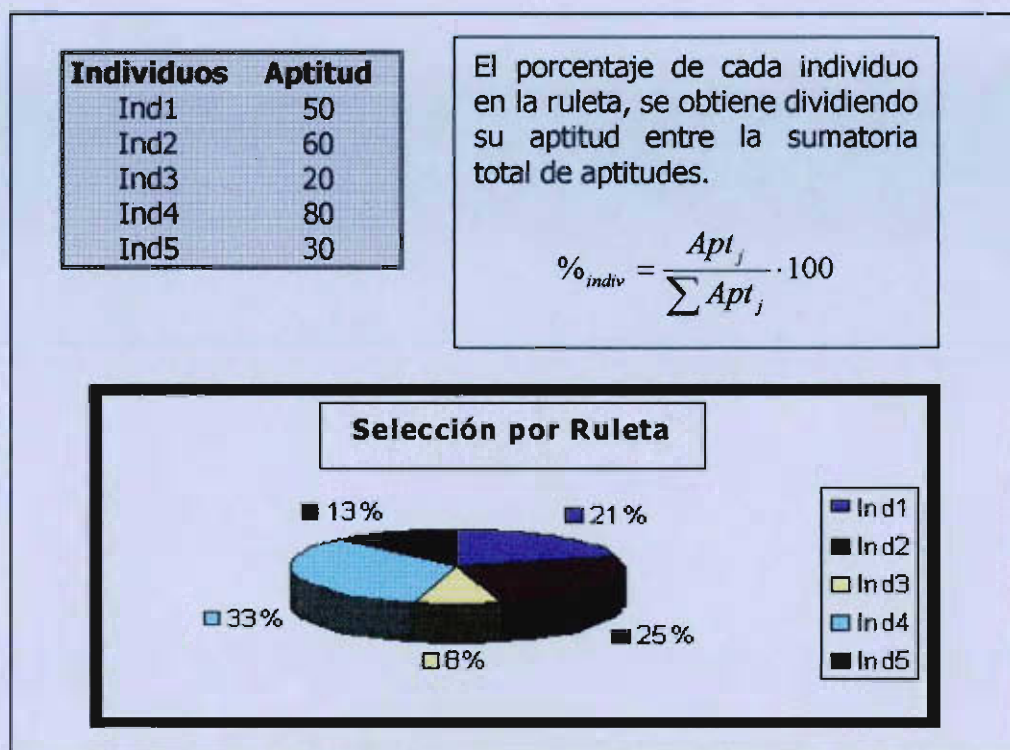


Figura 3. Selección por Ruleta.

Posteriormente los caracteres más significativos, a partir del punto de corte, se conservan en sus posiciones relativas en el nuevo par de cromosomas, y los caracteres restantes son intercambiados de los cromosomas progenitores a los nuevos dos cromosomas obtenidos.

Sean los siguientes cromosomas:

A:

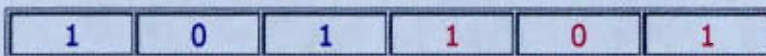


B:



La Cruza establece un "PUNTO DE CORTE" e intercambia la información genética de los cromosomas; obteniendo así el siguiente par de nuevos cromosomas:

A':



B':



**4.4.3.1 Técnicas de Cruce.** Existen diferentes técnicas de para realizar el cruzamiento entre los individuos seleccionados.

**4.4.3.1.1 Cruce Básico.** Se selecciona un punto al azar de la cadena (cromosoma). La parte anterior del punto es copiada del genoma del padre y la posterior del de la madre.

**4.4.3.1.2 Cruce Multipunto.** Es igual que el cruce básico, sólo que se establecen más de un punto de cruce en el cromosoma.

**4.4.3.1.3 Cruce Segmentado.** Existe una probabilidad de que un cromosoma sea punto de un cruce. Conforme se conforma la nueva cadena del descendiente, para cada gen, se verifica si ahí se va a producir un cruce.

**4.4.3.1.4 Cruce Uniforme.** Para cada gen de la cadena del descendiente existe una probabilidad de que el gen pertenezca al padre, y otra de que pertenezca a la madre.

**4.4.3.1.5 Cruces para Permutaciones.** Existe una familia de cruces específicos para los problemas de permutación.

**4.4.3.1.5.1 Cruce de Mapeamiento Parcial.** Toma una subsecuencia del genoma del padre y procura preservar el orden absoluto de los fenotipos - es decir, orden y posición en el genoma - del resto del genoma lo más parecido posible de la madre.

**4.4.3.1.5.2 Cruce de Orden.** Toma una subsecuencia del genoma del padre y procura preservar el orden relativo de los fenotipos del resto del genoma lo más parecido posible de la madre.

**4.4.3.1.5.3 Cruce de Ciclo.** Tomamos el primer gen del genoma del padre, poniéndolo en la primera posición del hijo, y el primer gen del genoma de la madre, poniéndolo dentro del genoma del hijo en la posición que ocupe en el genoma del padre.

El fenotipo que está en la posición que ocupa el gen del genoma del padre igual al primer gen del genoma de la madre se va a colocar en la posición que ocupe en el genoma del padre, y así hasta completar el genoma del hijo.

**4.4.4 Mutación.** El proceso de mutación es básicamente una búsqueda aleatoria. Se selecciona aleatoriamente una posición específica dentro de un cromosoma, para cambiar luego el valor contenido en dicha posición. Como en la naturaleza, la probabilidad de que ocurra una mutación en los individuos de una especie es muy pequeña, en condiciones normales de vida, en el AG se trata de representar lo mismo con un valor de probabilidad muy baja para la aplicación del operador de Mutación.

La mutación establece un cromosoma aleatoriamente y cambia la información contenida en él (Ver Figura 4).

1	0	1	0	1	1
1	0	1	1	1	1

Figura 4. Mutación de un Bit

**4.4.4.1 Técnicas de Mutación.** Al igual que en la selección y el cruce, existen también diversos procedimientos para realizar la mutación.

**4.4.4.1.1 Mutación de un Bit.** Existe una única probabilidad de que se produzca una mutación de algún bit. De producirse, el algoritmo toma aleatoriamente un bit, y lo invierte.

**4.4.4.1.2 Mutación Multibit.** Cada bit tiene una probabilidad de mutarse o no, que es calculada en cada pasada del operador de mutación multibit.

**4.4.4.1.3 Mutación Multigen.** Igual que la mutación de multibit, solamente que, en vez de cambiar un conjunto de bits, cambia un conjunto de genes. Puede sumar un valor aleatorio, un valor constante, o introducir un gen aleatorio nuevo.

**4.4.4.1.4 Mutación de Intercambio.** Existe una probabilidad de que se produzca una mutación. De producirse, toma dos bits/genes aleatoriamente y los intercambia.

#### **4.5 FUNCIONAMIENTO DE UN ALGORITMO GENÉTICO**

La forma de operar de los Algoritmos Genéticos es más o menos la misma para todos, se diferencian unos de otros en la manera de cómo y cuales operadores utilizan (Ver Anexo A numera A.1 y numeral A.3). A continuación se describen una serie de pasos generales para el modo de trabajar de un AG:

- Identificar el problema a resolver, encontrar una función de Aptitud que se acomode mejor a las necesidades del problema, es decir una función de Aptitud con la cual se pueda establecer un buen criterio de selección.
- Codificar los parámetros con los cuales se va a trabajar (Por lo general se utiliza la codificación binaria).
- Establecer el Espacio de Búsqueda, y generar una población inicial aleatoria que se encuentre uniformemente distribuida en este espacio.
- Realizar una selección de los mejores individuos de la población generada inicialmente, tomando como criterio de escogencia la función de Aptitud.
- Una vez escogida la nueva población, se realiza el cruce entre los individuos seleccionados.
- En cada nueva población existe la probabilidad de que se produzca una mutación o no, esto depende del criterio que se haya establecido.

- Por último se procede a evaluar la nueva población, es decir se realiza nuevamente el proceso de selección y se continua con los demás pasos, hasta que el algoritmo cumpla con algún criterio de finalización (EL algoritmo converja).

Los pasos anteriores se visualizan en la Figura 5.

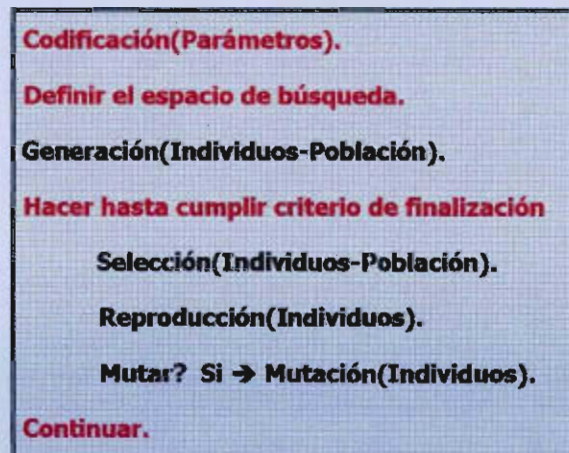


Figura 5. Esquema de Funcionamiento de un AG.<sup>7</sup>

## 4.6 TIPOS DE ALGORITMOS GENÉTICOS

Existen varios tipos de Algoritmos Genéticos, cada uno basado en una metáfora distinta de la naturaleza.

**4.6.1 Algoritmos Genéticos Generacionales.** Se asemeja a la forma de reproducción de los insectos, donde una generación pone huevos, se aleja geográficamente o muere y es substituida por una nueva. En este modelo se realizan cruces en una población de individuos, los descendientes son puestos en otra, al final de la fase reproductiva se

<sup>7</sup> Este esquema se encuentra en la mayoría de sitios con información de Algoritmos Genéticos, aunque en este se le dio un toque personal.

elimina la generación anterior y se pasa a utilizar la nueva. Este modelo también es conocido como Algoritmo Genético Canónico.

**4.6.2 Algoritmos Genéticos de Estado Fijo.** Utilizan el esquema generacional de los mamíferos y otros animales de vida larga, donde coexisten padres y sus descendientes, permitiendo que los hijos sean educados por sus progenitores, pero también que a la larga se genere competencia entre ellos.

En este modelo, no sólo se debe seleccionar los dos individuos a ser padres, si no también cuales de la población anterior serán eliminados, para hacer espacio a los descendientes.

La diferencia esencial entre el reemplazo generacional y el modelo de estado fijo, es que las estadísticas de la población son recalculadas luego de cada cruce y los nuevos descendientes están disponibles inmediatamente para la reproducción. Esto permite al modelo utilizar las características de un individuo prometedor tan pronto como es creado.

Algunos autores dicen que este modelo tiende a evolucionar mucho más rápido que el modelo generacional, sin embargo investigaciones de Goldberg y Deb (1993) encontraron que las ventajas parecen estar relacionadas con la alta tasa de crecimiento inicial, ellos dicen que los mismos efectos pueden ser obtenidos en rangos de adaptación exponencial o selección por competencia. No encontraron evidencia que este modelo sea mejor que el generacional.

**4.6.3 Algoritmos Genéticos Paralelos.** Parte de la metáfora biológica que motivó a utilizar la búsqueda genética consiste en que es inherentemente paralela, donde al evolucionar se recorren simultáneamente muchas soluciones, cada una representada por un individuo de la población. Sin embargo, es muy común en la naturaleza que no sólo sea

una población evolucionando, si no varias poblaciones, normalmente aisladas geográficamente, que originan respuestas diferentes a la presión evolutiva.

Esto origina dos modelos que toman en cuenta esta variación, y utilizan no una población como los anteriores si no múltiples concurrentemente.

**4.6.3.1 Modelo de Islas.** Si se tiene una población de individuos, esta se divide en subpoblaciones que evolucionan independientemente como un algoritmo genético normal. Ocasionalmente, se producen migraciones entre ellas, permitiéndoles intercambiar material genético.

Con la utilización de la migración, este modelo puede explotar las diferencias en las subpoblaciones; esta variación representa una fuente de diversidad genética. Sin embargo, si un gran número de individuos emigran en cada generación, ocurre una mezcla global y se eliminan las diferencias locales, y si la migración es infrecuente, es probable que se produzca convergencia prematura en las subpoblaciones.

**4.6.3.2 Modelo Celular.** Coloca cada individuo en una matriz, donde cada uno sólo podrá buscar reproducirse con los individuos que tenga a su alrededor (más cerca de casa) escogiendo al azar o al mejor adaptado. El descendiente pasará a ocupar una posición cercana.

No hay islas en este modelo, pero hay efectos potenciales similares. Asumiendo que el cruce está restringido a individuos adyacentes, dos individuos separados por 20 espacios están tan aislados como si estuvieran en dos islas, este tipo de separación es conocido como aislamiento por distancia.

Luego de la primera evaluación, los individuos están todavía distribuidos al azar sobre la matriz.

Posteriormente, empiezan a emerger zonas con cromosomas y adaptaciones semejantes. La reproducción y selección local crea tendencias evolutivas aisladas, luego de varias generaciones, la competencia local resultará en grupos más grandes de individuos semejantes.

#### **4.7 VENTAJAS Y DESVENTAJAS DE LOS ALGORITMOS GENÉTICOS**

Los Algoritmos Genéticos presentan ciertas ventajas y desventajas con los métodos de búsqueda tradicionales como por ejemplo:

- No necesitan conocimientos específicos sobre el problema que intentan resolver.
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Cuando se usan para problemas de optimización -maximizar una función objetivo- resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivamente paralelas.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

BIBLIOTECA UIS

- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen -tamaño de la población, número de generaciones, etc.-.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

#### **4.8 FUNDAMENTOS MATEMÁTICOS DE OPERACIÓN DE UN ALGORITMO GENÉTICO**

Al iniciar alguna mejora en la adaptabilidad de la población por reproducción estocástica, observando con algún detalle los diferentes cromosomas se encuentra que hay ciertos patrones que parecen estar asociados con buenos comportamientos. Avanzando más en este análisis podemos considerar dos aspectos:

- Buscar semejanzas entre los miembros de la Población.
- Examinar en dichas semejanzas posibles relaciones de su aptitud.

Al hacer esta consideración, se obtiene un caudal de nueva información que ayuda a guiar la búsqueda. Con el fin de estudiar con mayor aproximación esta nueva información, veamos ahora el concepto de Esquema, Estructuras de Similitud ó Plantilla de Semejanza.

**4.8.1 Esquema.** Un esquema (H) es un patrón o modelo, que describe el subconjunto de todas las posibles cadenas, que tienen bits iguales en ciertas posiciones. Por ejemplo, considere la cadena de 6 bits; el esquema **\*\*0000**, representa cadenas con ceros en las 4

últimas posiciones, sin importar si hay un 1 o un 0 en las 2 primeras posiciones; así el subconjunto que representa dicho esquema es 000000, 010000, 100000, 110000. De esta manera, se crean los esquemas sobre el alfabeto extendido  $\{0, 1, *\}$ , donde \* representa un 1 o un 0.

Cada cadena representada por un esquema, recibe el nombre de instancia de un esquema. Las posiciones fijas de un esquema, son las posiciones de la cadena con ceros (0`s) o unos (1`s); así en la cadena **\*\*0000** las posiciones fijas son la tercera, la cuarta, la quinta y la sexta, mientras que el esquema **\*\*\*\*\***, no tiene posiciones fijas.

#### 4.8.2 Propiedades de los Esquemas.

**4.8.2.1 Orden.** Es el número de posiciones fijas en un esquema, y se denota por  $O(H)$ .

Por ejemplo:

para

$$H_1 = 0\ 1\ 1\ * \ 1\ * \ *$$

$$H_2 = 0\ * \ * \ * \ * \ * \ *$$

$$O(H_1) = 4; O(H_2) = 1.$$

El orden de un esquema es una medida de especificidad. El concepto de orden del esquema se utiliza para calcular la probabilidad de supervivencia del esquema con relación al operador mutación.

**4.8.2.2 Longitud de Definición.** Es la distancia entre la primera y la última posición fija de un esquema, se denota por  $\delta(H)$ .

Por ejemplo:

Tomando los esquemas del ejemplo anterior.

$$\delta(H_1) = 5-1 = 4.$$

$$\delta(H_2) = 0$$

$\delta(H)$  es una medida de la probabilidad de sobrevivencia de un esquema al aplicar el operador de cruce.

**4.8.3 El Teorema de los Esquemas.** El Teorema de los Esquemas plantea lo siguiente:

*"Los esquemas pequeños de bajo orden arriba del promedio reciben un incremento exponencial de representantes en las siguientes generaciones de un algoritmo genético"<sup>8</sup>.*

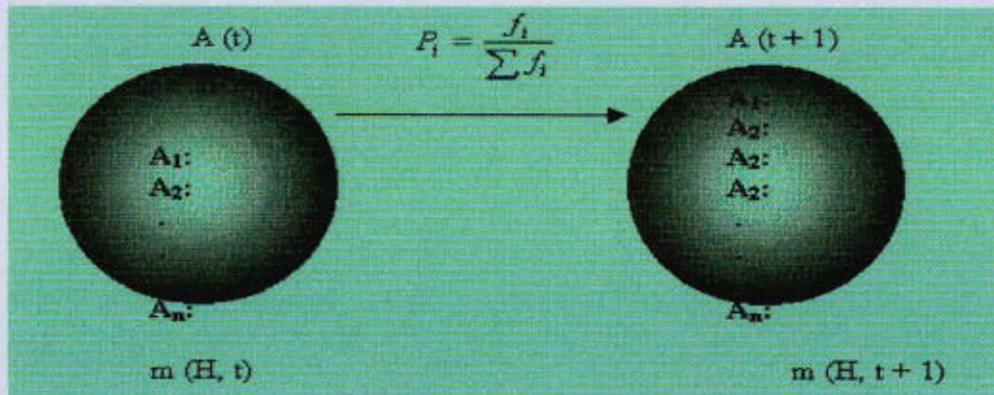
A continuación se presenta una descripción elaborada del Teorema:

$A(t) \rightarrow$  Población de  $n$  arreglos  $A_j$ ,  $j = 1, 2, 3, \dots, n$  en el tiempo  $t$ .

$m(H, t) \rightarrow$  Esquema  $H$  en la población  $A(t)$ .

---

<sup>8</sup> Holland, J.H; "Genetic Algorithms", Scientific American 26, July 1992, pp. 66-72.



$$m(H, t+1) = m(H, t) * n * \frac{f(H)}{\sum f_i}$$

**Ecuación 3. Esquema H en la población t+1.**

$f(H) \rightarrow$  Es el desempeño promedio de todos los arreglos  $A_j$  que representan al esquema H en el tiempo t.

El desempeño promedio de toda la población  $A(t)$  está dado por:

$$\bar{f} = \frac{\sum f_i}{n}$$

**Ecuación 4. Desempeño promedio de toda la población.**

Podemos describir la ecuación 3. como:

$$m(H, t+1) = m(H, t) \cdot \frac{f(H)}{\bar{f}}$$

**Ecuación 5. Otra Forma Esquema H en la población t+1.**

Entonces, los esquemas con un desempeño mayor que el promedio de la población reciben un número creciente de muestras en la siguiente generación. Mientras que los

que muestran un desempeño menor al promedio reciben un número decreciente de muestras.

Esto se da para cada esquema en  $A(t)$  en paralelo.

Supongamos ahora que un cierto esquema  $H$  muestra consistentemente un desempeño arriba / abajo del promedio por un valor  $c\bar{f}$  donde  $c$  es una constante positiva / negativa.

Podemos escribir la ecuación 5 como:

$$m(H, t+1) = m(H, t) \cdot \frac{\bar{f} + c\bar{f}}{\bar{f}} = m(H, t) \cdot (1+c)$$

**Ecuación 6. Esquema H en la Población t+1 con constante c.**

donde para  $t = 0$  y para un valor constante de  $c$  tenemos:

$$m(H, t) = m(H, \Theta) \cdot (1+c)^t$$

**Ecuación 7. Esquema H en la Población t.**

La reproducción entonces asigna un número exponencialmente creciente (decreciente) de copias a los esquemas que muestran un desempeño superior (inferior) al promedio de la población.

La aplicación del cruce crea nuevas estructuras afectando al mínimo el proceso de reproducción. Consideremos la siguiente estructura de longitud  $L$  y 2 esquemas asociados:

$$A = 0111000$$

$$H_1 = *1****0$$

$$H_2 = * * * 1 0 * *$$

El esquema  $H_1$  tiene menos probabilidades de sobrevivir ya que en promedio es muy probable que en el punto de cruce este entre las posiciones 2 y 7.

$\delta(H_1) = 7 - 2 = 5$ ; existen  $l - 1$  posibles posiciones para realizar el cruce. Si seleccionamos una aleatoriamente y con igual posibilidad. Entonces la probabilidad de desaparecer de  $H_1$  esta dada por:

$$P_d = \frac{\delta(H_1)}{l-1} = \frac{5}{6} \text{ y } P_s = 1 - P_d = \frac{1}{6}$$

Para  $H_2$ :

$$P_d = \frac{1}{6}; P_s = \frac{5}{6}$$

En general la probabilidad de sobrevivencia de un esquema  $H$  esta dada por:

$$P_s = 1 - \frac{\delta(H)}{l-1}$$

**Ecuación 8. Probabilidad de sobrevivencia de un esquema.**

además se aplica el cruce con probabilidad  $P_c$ :

$$P_s \geq 1 - P_c \cdot \frac{\delta(H)}{l-1}$$

**Ecuación 9. Probabilidad de sobrevivencia con probabilidad de cruce.**

El efecto combinado de la selección y el cruce esta dado entonces por: (suponiendo independencia de ambas operaciones).

$$m(H, t+1) \geq m(H, t) * \frac{f(H)}{f} \left[ (1 - Pc) * \frac{\delta(H)}{L-1} \right]$$

**Ecuación 10. Efecto combinado de la Selección y el cruce.**

Si consideramos ahora una probabilidad de mutación  $P_m$ . La probabilidad de sobrevivir de un alelo particular es de  $1 - P_m$  y la probabilidad de sobrevivencia en un esquema está dada por:

$$P_{sm} = (1 - P_m)^{O(H)}$$

**Ecuación 11. Probabilidad de sobrevivencia con probabilidad de mutación.**

Si  $P_m \ll 1$  podemos aproximar  $P_{sm}$  como:

$$P_{sm} = 1 - O(H) * P_m$$

**Ecuación 12. Probabilidad de sobrevivencia si  $P_m \ll 1$ .**

El número esperado de copias de un esquema H considerando la aplicación de selección, cruce y mutación esta dado por el TEOREMA DEL ESQUEMA.

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{f} \left[ (1 - Pc) * \frac{\delta(H)}{L-1} - O(H) * P_m \right]$$

**Ecuación 13. Teorema del Esquema.**

**4.8.4 Visualización de un Esquemas con Hiperplanos.** Como ya se mencionaron, los esquemas son entidades definidas sobre el espacio de genotipos y que representan subconjuntos de éste. Dichos subconjuntos se obtienen fijando los valores de ciertas posiciones de un cromosoma y dando libertad al valor de las restantes. Puede comprobarse que, si se considera el espacio de genotipos como un hiperespacio  $\lambda$ -dimensional, al fijar el valor de  $k$  posiciones de una cadena sé esta definiendo un hiperplano  $(\lambda-k)$ -dimensional o esquema de orden  $k$  (Ver Figura 6).

**4.8.5 Crítica al Teorema de los Esquemas.** El Teorema de los esquemas anteriormente mostrado ha sido considerado durante largo tiempo como una explicación convincente del funcionamiento de los algoritmos genéticos.

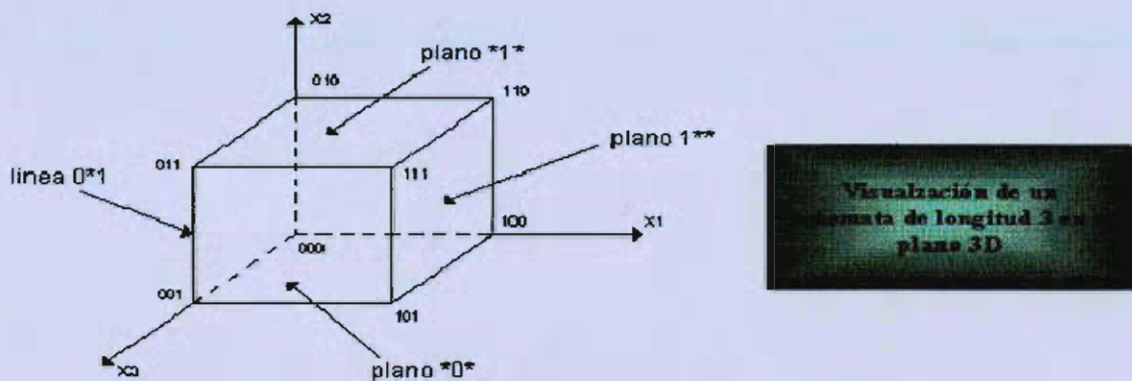


Figura 6. Esquema de longitud 3.

Sin embargo, actualmente ha surgido una fuerte corriente crítica cuestionando la validez de dicho teorema y de distintos resultados que emanaban de él. Entre los argumentos que justifican esta actitud se encuentran los siguientes:

- Desconexión entre el procesamiento de esquemas y la construcción de soluciones válidas (Altenberg, 1995): El Teorema de los esquemas afirma que los esquemas cuya adecuación media observada sea superior a la media aumentarán su número de ocurrencias en la población. Sin embargo, esto no guarda relación directa con que dichos esquemas permitan construir mejores soluciones. Por supuesto, la hipótesis de los bloques de construcción sí garantiza esto, pero dicha hipótesis es lógicamente independiente del teorema, el cual se debe seguir verificando incluso cuando la hipótesis no es válida.
- Inadecuación de los esquemas (Radcliffe, 1992): los esquemas permiten agrupar conjuntos de soluciones con similitud léxica en sus codificaciones. Sin embargo el número de posibles agrupaciones con relevancia para el problema en cuestión es astronómicamente mayor que el número de esquemas posibles, lo que implica que ciertos subconjuntos no podrán ser así agrupados.
- Generalidad de El Teorema de los esquemas: El Teorema de los esquemas continúa siendo válido si  $H$  es un subconjunto arbitrario de soluciones en lugar de ser un hiperplano definido por un esquema (Vose, 1991). Por lo tanto, la caracterización de un algoritmo genético como un manipulador de esquemas ha de ser replanteada.<sup>9</sup>

#### 4.9 AREAS DE APLICACION DE LOS ALGORITMOS GENÉTICOS

Inicialmente los Algoritmos Genéticos, fueron utilizados para la optimización (encontrar máximos y mínimos) y resolver problemas de optimización combinatoria, pero debido a su

---

<sup>9</sup> Información sobre el Teorema de los Esquemas se analizó de la siguiente dirección de internet: [tigre.aragon.unam.mx/geneticos/indice.htm](http://tigre.aragon.unam.mx/geneticos/indice.htm)

sencilla implementación, que presentan mayor robustez que las técnicas tradicionales y que compiten con estas en cuanto a tiempo y precisión. Los Algoritmos han encontrado un amplio campo de acción en áreas como:

- Las Finanzas
- El Sector Energético
- Las Telecomunicaciones
- Medicina
- Medio Ambiente
- Comercio
- Industria

Sus principales aplicaciones son La optimización de funciones, identificación de sistemas lineales y no lineales y el diseño de sistemas de control (Ver Anexo A numeral A.2).

**4.9.1 Tipos de Aplicaciones de los Algoritmos Genéticos.** Algunos ejemplos de las aplicaciones de los Algoritmos Genéticos son:

- Decisión y Estrategia
  - Toma de decisiones financieras (presupuestos)
  - Búsqueda de reglas en juegos
  - Experimentación de alternativas de marketing
  - Gestión de franquicias
- Diseño y Parametrización
  - Diseño de pistas de circuitos integrados VLSI

- Parametrización de Sistemas (configuraciones de equipos hardware)
- Diseño de redes (telecomunicaciones, carreteras). Ubicación de nodos
  
- Planificación y asignación de recursos
  - Asignación del orden de N procesos en M CPU
  - Ordenación (ordenar cajas en un almacén)
  - Asignación de horarios de clase o turnos de trabajo en un hotel
  - Resolución de sistemas de ecuaciones no lineales
  - Enrutamiento
  - El problema del viajante de comercio (TSP, Travel Salesman Problem)
  
- Predicción
  - Selección de combinación de métodos de realización de series temporales

En la actualidad podemos encontrar aplicaciones más específicas, interesantes y novedosas como por ejemplo:

- Reconocimiento de Criminales, creado por Caldwell and Jonson. (Ver anexo B numeral B.1).
- Generación de la Trayectoria de un brazo robótico, creado por Yuval Davidor. (Ver anexo B numeral B.2).
- Sistemas Dinámicos no lineales, y Seguridad Internacional, propuesto por Stephanie Forrest y Gottfried Mayer-Kress. (Ver anexo B numeral B.3).

- Síntesis Genética de la Arquitectura de una Red Neuronal propuesto por Steven A. Harp y Tariq Samad. (Ver anexo B numeral B.4).
- Método para Pronosticar Múltiples Fallas creador por Gunar E. Liepens y W. D. Potter. (Ver anexo B numeral B.5).
- Análisis Conformacional de DNA, por C. B. Lucasius, M. J. J. Blommers, L. M. C. Buydens, y G. Kateman. (Ver anexo B numeral B.6).<sup>10</sup>

**4.9.2 Algoritmos Genéticos y el Aprendizaje Supervisado.** Como hemos visto Los Algoritmos Genéticos se pueden usar a una gran cantidad de aplicaciones; incluso se han aplicado a diversos problemas dentro del aprendizaje automático solucionando tareas que van de la planificación al desarrollo de topologías neuronales. De entre todas estas aplicaciones la más prometedora es la del aprendizaje supervisado, es decir, a partir de una serie de ejemplos (y posiblemente contraejemplos) de uno o varios conceptos, generar un conjunto de reglas que permitan, dado un individuo, decidir la clase a la cual pertenece. Las soluciones propuestas para resolver este tipo de problemas pueden caracterizarse básicamente según dos criterios:

- ¿Cómo se representan los objetos?
- ¿Qué representa un individuo en la población?

Respecto de la primera de las preguntas existen básicamente dos tipos de representaciones, las cuales a su vez influyen en la definición de los operadores genéticos a utilizar.

---

<sup>10</sup> Para ampliar información acerca de las aplicaciones de los Algoritmos Genéticos consultar la página de los Profesores Enrique Alba, Carlos Cotta: [neo.lcc.uma.es/TutorialEA/semEC/main.html](http://neo.lcc.uma.es/TutorialEA/semEC/main.html)

- **Clásicas:** Se mantiene la representación de cadenas binarias, aunque algunas veces de longitud variable.
- **Ad hoc:** En este caso la representación es mucho más cercana al problema a resolver, y por ejemplo, en tareas de clasificación suele utilizarse la representación atributo valor.

Los sistemas que siguen la primera tendencia siguen muy fielmente el esquema clásico de algoritmo genético ya que la población consiste en diferentes soluciones posibles que compiten entre sí.

En general ha habido dos respuestas a la segunda de las preguntas, cada una de las cuales representa una escuela diferente en el uso de algoritmos genéticos de aprendizaje.

Dichas escuelas son:

- **Michigan:** En este enfoque los individuos de la población representan reglas individuales, por lo que la solución del problema consiste en toda la población.
- **Pittsburgh:** La población consiste en conjuntos de reglas de cardinalidad variable, por tanto, cada individuo representa una posible solución del problema a resolver.

Dentro del primer enfoque cabe destacar los **sistemas clasificadores** (classifier systems), cuyo estudio y análisis por si mismo constituye una disciplina aparte. Todos los sistemas que siguen este enfoque tienen un problema adicional cuya resolución es crítica

para su funcionamiento: Como la solución del problema es toda la población ¿cómo se evalúa la bondad de sus individuos?, Pues la bondad de la solución hace referencia a todos ellos.

**4.9.2.1 Método Michigan.** También llamados sistemas clasificatorios. Cada regla (clasificador) manipula un mensaje.

El sistema interactúa con el ambiente el cual manda mensajes (en este sentido es parecido a un sistema experto).

Las reglas son del tipo:

If condición (mensajes que satisface)

Then acción (manda un mensaje)

En general, todos los mensajes son del mismo tamaño sobre un alfabeto predefinido.

a. Componentes:

- Interfase de entrada: Convierte el estado del ambiente en mensajes
- Clasificadores o reglas: Definen el proceso de mensajes
- Lista de mensajes: mensajes de entrada y dados por las reglas disparadas
- Interfase de salida: Traduce los mensajes en acciones para modificar el ambiente

b. Ciclo de ejecución:

- Coloca todos los mensajes de la interfase de entrada en la lista de mensajes
- Itera:
  - Compara todos los mensajes con la lista de condiciones de los clasificadores.
  - Dispara todos los clasificadores que satisfacen sus condiciones (y tienen suficiente fuerza) formando con cada acción una lista de mensajes nuevos.
  - Reemplaza todos los mensajes por los mensajes nuevos.

La especificación de las condiciones de las reglas normalmente es por medio de  $\{0,1,\#\}$  y permiten tener un *Mensaje*.

El ambiente proporciona el refuerzo (premio/castigo) y se tiene que pensar en un método para asignar crédito a las reglas que intervinieron en la solución.

Uno de los métodos (por lo menos inicialmente) más usados es el *bucket brigade* el cual distribuye la ganancia entre las reglas que sirvieron para la solución.

A cada regla se le asigna una cantidad (fuerza) que se ajusta y se usa para competir.

c. La competencia se basa en oferta: Las reglas más fuertes son las que ganan.

En el proceso de oferta se toman en cuenta dos factores:

- Utilidad: Es la Fuerza de la regla  $s(C, T)$  = fuerza del clasificador  $C$  al tiempo  $t$ .
- Relevancia: Especificidad  $R(C)$  = número de no "#s" en la condición de la regla entre su longitud.

En el ciclo general se calcula la oferta (bid) de los clasificadores:

$$B(C, t) = b * R(C) * S(C, t)$$

Donde,  $b$  = constante  $< 1$ .

El tamaño de la oferta determina la probabilidad de que el clasificador ponga su mensaje (por ejemplo, la probabilidad puede decrecer exponencialmente con la oferta).

Bucket brigade trata a cada regla como un intermediario el cual se entiende con sus proveedores (reglas que hacen que se satisfagan sus condiciones) y sus consumidores (reglas que satisfacen sus condiciones con sus acciones).

Cuando una regla gana una oferta, paga parte a sus proveedores (si no gana, no paga nada) y puede recibir pago de sus consumidores. Su fuerza aumenta cuando recibe más de lo que da.

Un clasificador ( $C$ ) al depositar un mensaje paga:

$$s(C, t+1) = s(C, t) - B(C, t)$$

Los clasificadores ( $C$ ) que mandaron mensajes para que ganara el clasificador (proveedores) aumentan su fuerza:

$$s(C, t+1) = s(C, t) + a * B(C, t)$$

Donde,  $a = \frac{1}{\text{Elementos\_de\_}C}$

- Las reglas finales obtienen el pago del ambiente
- Las reglas malas son eliminadas por competencia
- Con el tiempo, las mejores reglas tienen más peso y son favorecidas a disparar por el mecanismo de resolución de conflictos

d. Puntos en general de los GA para reglas de producción:

Reglas no ordenadas.

- LHS: Algunos patrones de longitud fija usando  $\{0, 1, \#\}$ , aunque hay quienes argumentan que es demasiado restrictivo.

En general hay que establecer un balance entre simplicidad y poder expresivo.

También se ha sugerido que los patrones del LHS regresen un valor de apareo.

- Memoria de trabajo: Dejar o no que las reglas hagan cambios en la memoria de trabajo. Hay quienes están en contra de esto y argumentan que las aplicaciones no ameritan la generalidad y complejidad adicional (se tendría que restringir el número

de acciones sobre la memoria de trabajo) y la mayoría de los trabajos en aprendizaje de conceptos se han enfocado en reglas tipo estímulo-respuesta.

- Retroalimentación: A pesar de que se considere la representación adecuada y la arquitectura de sistemas de producción adecuados, la efectividad también depende en buena medida de la retroalimentación.

No todos los esquemas de premio/castigo son adecuados (por ejemplo, muy poco premio a menos que sea casi la solución no nos sirve)

El bucket brigade distribuye la retroalimentación. Pero también se tiene que considerar el número de reglas y el tiempo.

Se puede pensar en dar retroalimentación tanto a cada regla como al conjunto de reglas.

e. Conocimiento del dominio: En principio se ven a los GA como métodos de búsqueda independientes del dominio. Sin embargo conocimiento del dominio puede incorporarse en:

- Los operadores genéticos a utilizar
- La representación usada
- Poblaciones iniciales
- Retroalimentación

**4.9.2.2 Método Pittsburg.** Hay que considerar la representación. Cruce nos da nuevas combinaciones de reglas y mutación nuevas reglas.

Como los genes pueden asumir muchos valores podemos convergir prematuramente, por lo que hay que usar una representación binaria para que el cruce también produzca

nuevas reglas. Pero ahora tenemos que asegurar que produzcan reglas válidas y potencialmente útiles. Una forma es asegurarse que todas las reglas tienen una longitud fija.

También se pueden hacer operadores sensibles a la representación para hacer cambios de acuerdo a la sintaxis.

El otro punto es el tamaño de la población. Difícil justificar que el sistema de reglas tenga una longitud fija (aunque se podría justificar en términos de reglas redundantes).

Existen enfoques con poblaciones de longitud variable (Smith 80). En estas se debe que tener un balance entre el tamaño y su desempeño.

Ciclo de operación:

- Selección: Se observa el desempeño de cada individuo (conjunto de reglas) de la población con respecto a los otros individuos en la población y se selecciona (posiblemente varias veces) asegurándose que su número de descendientes es proporcional a su desempeño con respecto a los otros individuos.
- Recombinación: Se combinan pares de individuos por cruce, pero a diferentes niveles de granularidad (LS-1):
  - A nivel de listas de reglas (normal).
  - A nivel de reglas (con partes de pares de reglas). Se pueden ver como variantes de reglas de padres de alto desempeño.
  - Operador de inversión

## 5 MEJORAS PROPUESTAS A LOS ALGORITMOS GENÉTICOS TRADICIONALES

Los AG's se utilizan en problemas de optimización, y su campo de acción principal está donde no existen técnicas especializadas para la resolución de ciertos problemas como por ejemplo métodos para la integración numérica como el de Euler y Runge Kutta.

Los AG's clásicos trabajan con una codificación del conjunto de parámetros, no con estos directamente. Como por ejemplo en la optimización de una base de conocimiento o en la optimización de funciones.

### **Ejemplo:**

**Objetivo :** Optimizar una base de conocimiento.

**Población:** La base de conocimiento (conjunto de parámetros a codificar).

**Entorno:** El dominio del sistema basado en el conocimiento.

**Función de Aptitud:** Cada regla tiene asociado un grado de verdad en el entorno en el que actúa, dependiente de cómo se comporte en el sistema.

**Reproducción :** Generan conocimiento a partir del conocimiento existente.

Los Parámetros son las Reglas existentes en la base del conocimiento. Cada regla consta de uno o varios antecedentes separados por  $\text{Y}$  (AND) y un único consecuente  $\rightarrow$  (THEN).

Si la presión es Baja Y la temperatura es Alta	<b>Antecedentes</b>
<b>Entonces</b> Abrir la válvula un Poco	<b>Consecuente</b>

Las reglas se codifican de la siguiente manera (Tabla 1):

**Tabla 1. Ejemplo de Codificación de Reglas.**

Presión	A	000
Temperatura	B	001
Abrir	C	010
Cerrar	D	011
Alta	E	100
Baja	F	101
Mucho	Z	110
Poco	X	111

Según lo anterior la regla queda codificada de la siguiente manera:

Si la presión es Baja	Si A pertenece a F	000 101
Y la temperatura es Alta	Y B pertenece a E	001 100
<b>Entonces</b> Abrir la válvula un Poco	<b>Entonces</b> C pertenece a X	<b>010 111</b>

Si se tienen las siguientes reglas:

**Regla\_1:** Si la Presión es Baja **Y** la Temperatura es Alta **Entonces** abrir la válvula un poco.

**Regla\_2:** Si la Presión es Alta **Y** la Temperatura es Baja **Entonces** abrir la válvula mucho.

Su codificación será:

<b>Regla_1</b>	000	101	001	100	010	111
----------------	-----	-----	-----	-----	-----	-----

<b>Regla_2</b>	000	100	001	101	010	110
----------------	-----	-----	-----	-----	-----	-----

Cuando se tengan codificadas todas las reglas se realiza, el proceso de selección de las mejores, esto bajo el criterio de sí ayuda o no a mejorar la base de conocimiento (De acuerdo al motor de inferencia que se tenga).

Para realizar el cruce o reproducción se debe tener en cuenta la semántica de la regla. Los puntos de corte deben hacerse en las intersecciones (Y ó Entonces).

Para los dos cromosomas anteriores un posible cruce sería:

Regla_1'	000	101	001	101	010	110
Regla_2'	000	100	001	100	010	111

Después se evalúan las reglas de la nueva población.

La mutación debe hacerse teniendo en cuenta el significado de la frase perteneciente a una determinada regla, por ejemplo si mutamos el tercer bit del primer antecedente de la Regla\_1 (bit # 15 del primer cromosoma).

El Cromosoma perteneciente a la regla\_1:

Primer Antecedente				Segundo Antecedente							Consecuente						
18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	1	0	1	0	0	1	1	0	0	0	1	0	1	1	1


Bit a Mutar

Al ser mutado Tenemos:



Esta nueva regla no tiene ningún significado, y será descartada por el motor de inferencia cuando se esté evaluando la calidad de la base de conocimiento.

Una buena mutación sería:



Luego se evalúan las nuevas reglas generadas se realiza el proceso de selección, cruzamiento, mutación, y así hasta que se cumpla el criterio de terminación del Algoritmo.

La codificación es una herramienta poderosa y sencilla de los AG's sobre todo cuando se utilizan – como se vio en el ejemplo – en la mejora de una base de conocimiento y en el diseño de RNA, ó en problemas similares; pero cuando se utilizan para optimizar funciones o en el ajuste curvas, resulta mucho más efectivo trabajar con los parámetros directamente, sin codificarlos, esto es porque al realizar la codificación (binaria) se deja de explorar cierta información en el espacio de búsqueda, por ejemplo para un cromosoma de 3 bits tengo 8 posibles valores:

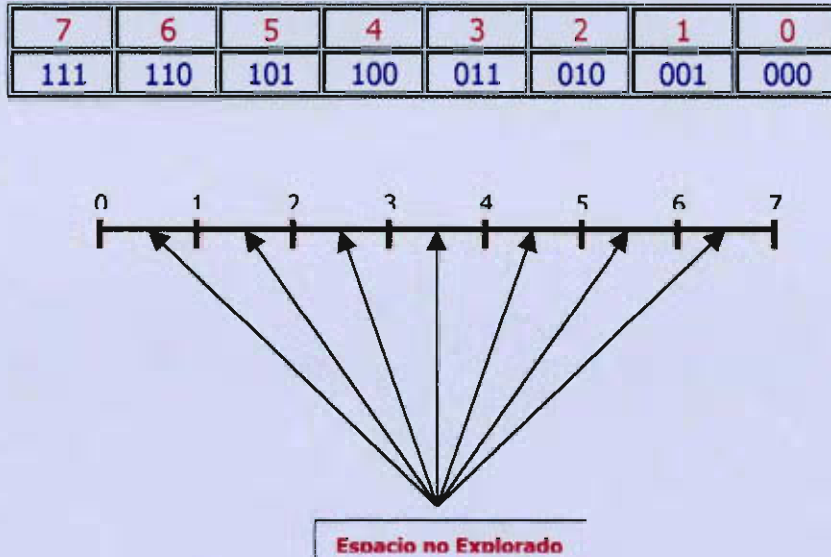


Figura 7. Ejemplo de posibles Valores para un cromosoma de 3 bits.

Si se necesitan generar números en un intervalo de 0 a 5 ( $X \in [0,5]$ ), se tiene que:

$$X_i = [5 * (\# \text{ codificado en binario})] / (2^3 - 1).$$

Los valores que se obtendrán con esta codificación son (Tabla 2):

Tabla 2. Ejemplo de Codificación de Parámetros.

Número	Codificación	$X_i$
0	000	0
1	001	0.7142857
2	010	1.4285714
3	011	2.1428571
4	100	2.8571428
5	101	3.5714285
6	110	4.2857142
7	111	5

Si el valor buscado es por ejemplo 2.54321 nunca se va a poder obtener con esta codificación, lo más aproximado sería 2.142857 ó 2.8571428.

Una posible solución sería aumentar los bits del cromosoma, y esto es a lo que se le conoce como cromosoma variable, pero de todas maneras se perdería información y no se podría encontrar una solución precisa si no se aumenta el número de cromosomas por generación, pero esto generaría un consumo mayor de recursos.

Según lo anterior los algoritmos genéticos se pueden clasificar también teniendo en cuenta la manera como manejan el conjunto de parámetros en:

- Algoritmos Genéticos con Codificación.
- Algoritmos Genéticos sin Codificación.

Los AG's sin Codificación serían ideales en la optimización de funciones, o en el ajuste de curvas o parámetros a un conjunto de datos; pero no servirían para resolver un proceso de mejoramiento de una base de conocimiento ó para el diseño de RNA, debido a que trabajan directamente con los parámetros del problema.

Los Algoritmos Genéticos, son simplemente una herramienta nueva con la cual es posible dar solución a muchos problemas en diversas áreas en especial de ingeniería.

Debido al gran auge que han tenido en los últimos años y el gran interés despertado por su manera de trabajar y por su gran campo de acción, la comunidad científica internacional se ha puesto en la tarea de investigar más a fondo esta técnica de búsqueda basada en la teoría de la evolución.

A continuación se proponen y exponen nuevas técnicas de generación de la población, selección, de mutación y cruce.

## 5.1 CROMOSOMA VARIABLE

La mayoría de los Algoritmos Genéticos trabaja con cromosomas de longitud fija, donde se conoce de antemano el número de parámetros de un problema. Pero hay problemas en los que esto no sucede, no conocemos el número de parámetros con los que vamos a trabajar por eso se hace necesario tener Cromosomas de Longitud variable, además también pueden ser utilizados en problemas donde conocemos el conjunto de parámetros, pero con el objetivo de mejorar la búsqueda y aumentar la precisión en la respuesta - para problemas de búsqueda y optimización de funciones -. Sin embargo en estos casos hablar de cromosoma variable resulta útil si el número de individuos de la población es al menos igual al número de divisiones del intervalo de búsqueda que se alcanza con la longitud (L) del cromosoma. Lo anterior será explicado con un ejemplo:

Supongamos que se tiene un cromosoma con codificación binaria de Longitud  $L=4$ , y que se genera una población inicial de 10 individuos, en un intervalo de  $-1$  a  $2$ .

Con un cromosoma de Longitud 4 se tienen 16 ( $2^4=16$ ) divisiones de este intervalo con una separación de 0.2 ( $Paso = \frac{2 - (-1)}{2^4 - 1} = 0.2$ ), pero si generamos sólo 10 individuos,

habrá 5 posibles valores que no se tendrán en cuenta. Entonces si se aumenta el tamaño del cromosoma de 4 a 5 ( $L=5$ ), se tiene 32 divisiones con un Paso de 0.096774193 (Ver Ecuación 14).

$$Paso = \frac{(b - a)}{2^l - 1}$$

**Ecuación 14. Paso con codificación binaria, en el intervalo (a, b).**

Pero al generar los mismos 10 individuos, no se está haciendo en realidad nada, porque así se tengan más divisiones del intervalo (Ver Figura 8), se están generando menos individuos que posibles puntos.

Para evitar éste problema el número de individuos debe ser por los menos igual al número de divisiones del intervalo, para un  $L=4$  se deben generar como mínimo 16 individuos y para  $L=5$  serian 32 individuos, este cambio en el tamaño de la población se conoce como tamaño de la población variable.<sup>11</sup>

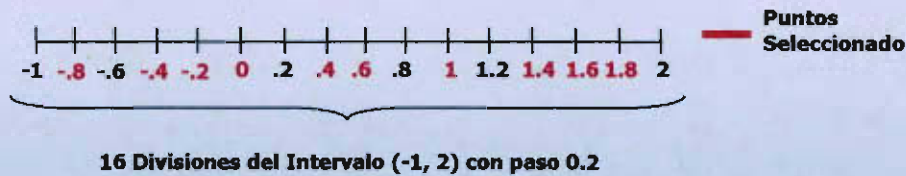
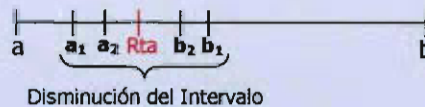


Figura 8. Intervalo de (-1, 2) con cromosoma de Longitud 4

**5.1.1 Intervalo Variable.** Esta es otra mejora propuesta, que junto con el tamaño de la población variable forman una analogía de lo que es y representa el cromosoma variable, puesto que todas estas metodologías pretenden optimizar el funcionamiento y rendimiento de un algoritmo genético, para llegar a una respuesta más rápida y precisa. El intervalo variable consiste en simplemente modificar la longitud del intervalo de búsqueda, ya sea aumentándolo o disminuyéndolo y conservando el tamaño de la población; se aumenta la longitud del espacio de búsqueda si después de explorarlo no se consigue ningún resultado satisfactorio – esto por lo general no ocurriría si el espacio de búsqueda está bien definido -. Por otro lado si el intervalo es el adecuado, y después de ejecutar el AG por cierto número de iteraciones podemos restringir el intervalo de

<sup>11</sup> Las ideas de las mejoras para proponer a los Algoritmos Genéticos existentes, surgieron del Profesor Fernando Ruiz en su clase de PNL, yo (El Autor) me encargue de probarlas, complementarlas y proponer otras nuevas.

búsqueda disminuyendo su longitud (ver figura 9), de acuerdo a los resultados obtenidos hasta ese momento.



**Figura 9. Intervalo Variable.**

Definir exactamente en que momento y cuantas veces se debe realizar la disminución del intervalo es complicado, y depende del problema que se este resolviendo. Una manera de hacerlo seria por ejemplo que cada  $n$  número de iteraciones se disminuya el intervalo, otra podría ser utilizando la desviación estándar de las aptitudes y mientras ésta se encuentre por encima o por debajo de un valor  $m$  establecido se realiza el cambio.

El intervalo variable es otra forma de realizar un cambio en el cromosoma (cromosoma variable), incluso resulta mucho más fácil de implementar, porque lo único que se necesita es cambiar los limites del intervalo de búsqueda y seguir generando el mismo número de cromosomas que se tenia, pero ahora en un espacio más reducido lo cual aumenta la precisión del algoritmo.

## 5.2 GENERACIÓN GAUSSIANA O NORMAL

Una Función importante dentro de un AG, es la generación de la población inicial de individuos y posteriores generaciones. Esta generación se realiza de manera aleatoria y uniforme en el espacio de búsqueda definido para el problema (Ver figura 10).

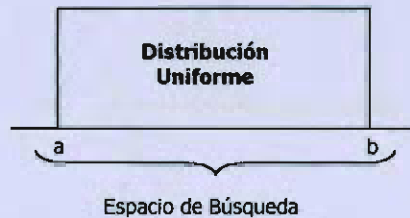
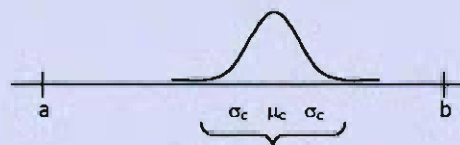


Figura 10. Generación Uniforme

La población inicial de individuos se genera de esta manera, debido a que en la mayoría de los casos se desconoce una aproximación a la solución, pero cuando el AG ha trabajado por cierto tiempo o número de poblaciones, ya se tiene una idea de hacia donde converge el algoritmo y de donde se encuentra la respuesta. En éste instante es donde se debe utilizar la generación de individuos utilizando la distribución de Gauss o Normal.

La generación Gaussiana consiste en generar individuos normalmente distribuidos con media  $\mu_c$  y desviación  $\sigma_c$  (Ver figura 11) dentro del nuevo espacio de búsqueda que se transforma en  $(-\sigma_c, \sigma_c)$ . Como parámetro  $\mu_c$  puede escogerse el mejor cromosoma de la población hasta ese momento o un promedio de los mejores individuos de las poblaciones anteriores, igualmente  $\sigma_c$  puede escogerse como la desviación de la aptitud de la población actual, o la desviación de las aptitudes de los mejores individuos de las poblaciones anteriores.



Distribución Normal  
Figura 11. Generación Gaussiana.

Para saber cuando se deben realizar las generaciones Gaussianas secundarias de individuos, se pueden establecer diversos criterios como:

- Cada cierto número  $m$  de poblaciones.
- Cada intervalo de tiempo  $t$ .
- También se puede utilizar la desviación estándar de las aptitudes de las poblaciones, y cuando ésta esté por encima o por debajo de un valor determinado ya sea fijo o aleatorio se realiza la generación Gaussiana.
- Cuando la desviación de los valores de los cromosomas de una población sea pequeña o este cercana a un valor determinado.

### 5.3 MUTACIÓN DIRIGIDA

El papel de la mutación en los Algoritmos Genéticos, es secundario, perfectamente podría no estar incluida dentro del Algoritmo que igual éste funcionaria; primero La Mutación tiene muy baja probabilidad de que ocurra porque de lo contrario el Algoritmo perdería eficiencia y se convertiría en una búsqueda de la solución al azar, y segundo porque es muy poco probable que en las veces que se realice una mutación esta sea efectiva, es decir nos acerque a la solución. Por este motivo y para darle un poco más de importancia a la función de la mutación dentro del Algoritmo, se plantea una Mutación Dirigida.

A diferencia de la mutación natural, que es al azar y ocurre con mínima frecuencia – Lo cual plantea La Teoría de la Evolución -, y con base en la cuál trabajan los Algoritmos Genéticos Clásicos y que en la mayoría de los casos no produce buenos resultados; La Mutación Dirigida es manipulada y como su nombre lo indica guiada hacia una solución supuestamente mejor, partiendo del hecho que la mutación se realiza al final o cuando el

algoritmo este cerca de una solución, y no sobre todo el espacio de búsqueda sino cerca de la solución que se obtuvo – Podría hacerse la analogía con la manipulación genética que se realiza actualmente para prevenir enfermedades hereditarias, mejoramiento de especies animales y vegetales, la clonación, entre otras -.

Se dice que se usa la mutación para afinación de la respuesta porque, al hacerla dirigida las probabilidades de encontrar la solución óptima aumentan, esto porque partimos del supuesto que si el valor actual ( $V_c$ ) del cromosoma es bueno, también lo será un valor cercano a él:

$$V_{cm} = V_c \pm \varepsilon .$$

**Ecuación 15. Nuevo valor al Aplicar Mutación dirigida.**

**5.3.1 Mutación de genes menos significativos con factor de Mutación.** Esta técnica consiste en establecer una política para saber cuando mutar y como mutar. Para saber cuando mutar establecemos un criterio, el cual plantea lo siguiente:

- Si dos generaciones tienen una aptitud general ( $\Sigma Apt_i$ ) muy similar o parecida; para ello se establece la medida del valor absoluto del error entre aptitudes (Ver Ecuación 16), y cuando este error este por debajo de un factor de Mutación ( $\omega$ ) que consiste en un número  $F_m$  - ( $F_m \in (0-100)$ ) - multiplicado por un número randómico (Ver Ecuación 17), se efectúa la mutación. Entre más grande sea el Factor de mutación, mayor número de mutaciones se tendrán.

$$E_{apt} = \left| \frac{Apt_{j-1} - Apt_j}{Apt_j} \right|$$

**Ecuación 16. Error entre Aptitudes**

$$\omega = F_m * rnd$$

**Ecuación 17. Factor de Mutación**

También se puede tomar como política de cuando mutar, al final, es decir cuando el algoritmo haya encontrado una posible solución.

Una vez establecido cuando mutar, se escoge uno o varias cromosomas al azar para ello, y se define como mutar. Para saber como mutar, se procede a realizar una mutación dirigida:

- Se efectúa una mutación de los genes (bits) menos significativos, es decir se escoge entre estos bits para realizar la mutación, esto porque si hay que mutar, la mayoría de cromosomas serán parecidos y se supone que se encuentran cerca del intervalo de respuesta. La escogencia de los bits menos significativos depende de la longitud del cromosoma, por ejemplo en un cromosoma de longitud 10 ( $L=10$ ), los bits menos significativos podrían ser los 3 ó 4 últimos (Ver figura 12).



**Figura 12. Ejemplo Mutación Dirigida**

**5.3.2 Mutación por Intervalo.** Esta técnica es similar a la anterior, pero se aplica a los Algoritmos Genéticos sin codificación, en lo que radica su diferencia es que con esta técnica no se mutan los bits menos significativos, sino que se establece un intervalo de longitud corta; una vez establecido el intervalo  $(-b, a)$  se escoge un número aleatorio ( $j$ ) que esté dentro del intervalo ( $j \in (-b, a)$ ) y este número se suma al valor actual del cromosoma que se escogió para efectuarle la mutación.

## 5.4 SELECCIÓN POR PROMEDIO

En realidad esta no es una mejora, más bien es una nueva técnica que se propone como alternativa a las ya existentes como Selección por torneo, Ruleta, entre otras.

Esta técnica, se puede utilizar siempre y cuando se pueda cuantificar la aptitud de cada cromosoma de la población - es decir tenga un valor numérico -. La selección por promedio, consiste en hallar el valor promedio de las aptitudes de los cromosomas de la generación (Ver Ecuación 18) y su desviación estándar (Ver Ecuación 19), posteriormente se compara cada aptitud de cada individuo ( $Apt_j$ ) con un factor de selección ( $\rho$ ) que es el promedio ( $\mu$ ) más/menos un número randómico multiplicado por la desviación (Ver Ecuación 20). Escogemos el Promedio como base para garantizar que los individuos que se van a escoger en su mayoría estén por debajo/encima de éste, sin embargo los que estén por encima/debajo del promedio también pueden ser escogidos.

$$\mu = \frac{\sum Apt_j}{N}$$

Ecuación 18. Promedio de las Aptitudes

$$\sigma = \sqrt{\frac{\sum (\mu - Apt_j)^2}{N}}$$

Ecuación 19. Desviación Estándar

$$\rho = \mu \pm rnd * \sigma$$

Ecuación 20. Factor de Selección

## 5.5 CRUCE POR VALOR PONDERADO DE LA APTITUD

La mayoría de técnicas de cruce que existen, están diseñadas para los Algoritmos Genéticos que utilizan la codificación de parámetros, como el cruce básico, cruce segmentado, cruce lineal entre otros. Esta es una técnica creada para trabajar con los Algoritmos Genéticos que trabajan directamente sobre el conjunto de parámetros del problema, y consiste en lo siguiente:

El valor del nuevo cromosoma que se obtiene al cruzar el par seleccionado, está dado por la ecuación:

$$V_{NC} = \frac{Apt_j * V_j + Apt_{j+1} * V_{j+1}}{Apt_j + Apt_{j+1}}$$

**Ecuación 21. Cruce Ponderado**

Con esta formula se garantiza que el valor del nuevo cromosoma, esté más cerca del valor del cromosoma mejor adaptado, es decir que tenga un valor de aptitud mayor.

## **6 PRUEBA DE LAS MEJORAS PROPUESTAS A LOS ALGORITMOS GENÉTICOS TRADICIONALES**

### **6.1 DEFINICIÓN DEL PROBLEMA ESCOGIDO PARA REALIZAR LAS PRUEBAS**

Para realizar las pruebas a las mejoras propuestas y realizar la comparación entre los Algoritmos Genéticos Clásicos y los Mejorados, se escogió el problema de encontrar los parámetros  $a$  y  $b$  de la ecuación de la línea recta ( $y=ax+b$ ) que mejor se aproxime a un conjunto de puntos. Como criterio de aproximación se tomó la sumatoria de los errores euclidianos<sup>12</sup> al cuadrado de los puntos a la recta, el conjunto de parámetros que presente un menor error se escoge como la mejor recta que aproxima a todos los puntos.

### **6.2 APLICACIÓN DESARROLLADA PARA REALIZAR LA COMPARACIÓN ENTRE LOS ALGORITMOS GENÉTICOS CLÁSICOS Y LOS MEJORADOS**

Para realizar la comparación entre algoritmos se desarrolló una aplicación en el lenguaje de programación Visual Basic 6.0 llamada RECT\_AG, en la cual se implementó el problema el problema descrito anteriormente, desarrollado por tres tipos de Algoritmo Genético que se describirán más adelante, además de realizar el ajuste utilizando la metodología de mínimos cuadrados.

La aplicación presenta un interfaz en la cual, se pueden introducir los puntos, escoger el tipo de algoritmo que se quiere utilizar para realizar el ajuste a los puntos introducidos

---

<sup>12</sup> El error euclidiano es simplemente la distancia perpendicular que existe entre el punto y la recta.

además de escoger el modo de operación con que se quiere trabajar ya sea manual, semi\_automático o automático; Por último presenta un modulo e comparación de Algoritmos y final presenta un resumen de los resultados obtenidos. A continuación se explicara con más detalle al funcionamiento de la aplicación.

**6.2.1 Funcionamiento de la Aplicación.** La aplicación consta de varias partes, La primera es la del ingreso de datos y donde se escogen todas las características con las que se va a trabajar como por ejemplo el tipo de algoritmo que se desea utilizar y el modo de operación de este. La segunda es el modulo donde se muestra la ejecución del algoritmo, es decir como va convergiendo hacia una posible solución, presenta la opción de volver a comenzar con la ejecución del algoritmo para un mismo conjunto de puntos. La tercera es donde se muestran los resultados parciales o totales de la ejecución del algoritmo. También se tiene un modulo de comparación entre algoritmos, donde se escogen cuales algoritmos se desean comparar y por ultimo muestra los resultados obtenidos de la comparación y los almacena en una base de datos.

**6.2.1.1 Ingreso de Datos.** En este módulo es donde se define las características con las cuales se va a desarrollar la aplicación (Ver Figura 13), se introducen los puntos, ya sea manualmente o se pueden cargar de un archivo (Ver Figura 14) existente; se puede elegir el tipo y modo de operación del Algoritmo, además de sí desea elegir un intervalo de búsqueda personalizado o si se desea realizar la comparación entre algoritmos.

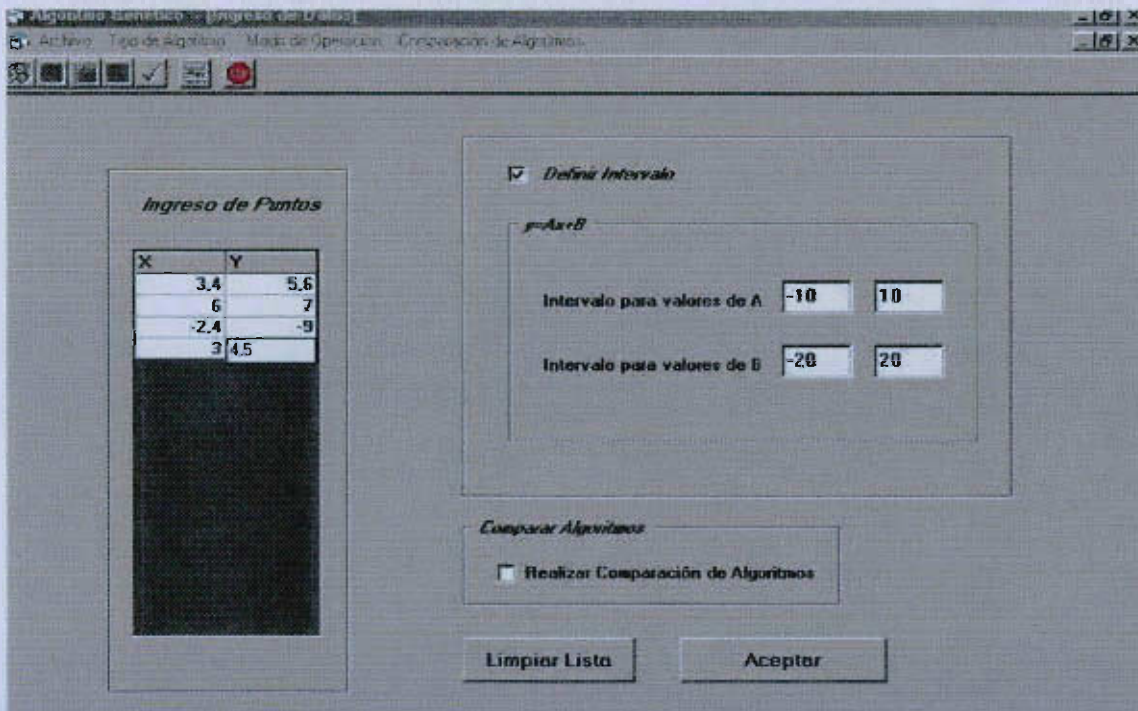


Figura 13. Ingreso de Datos.

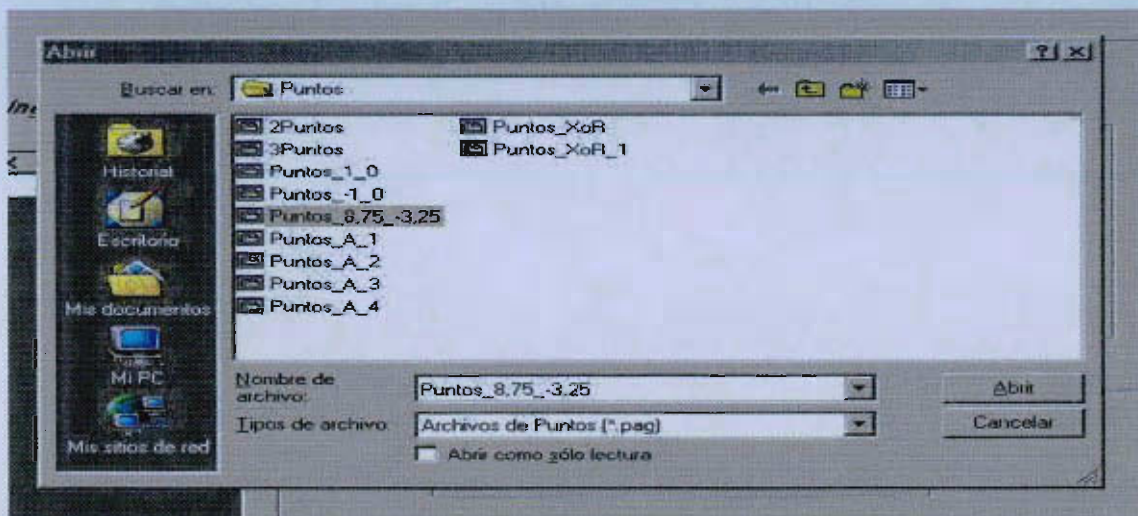


Figura 14. Cargar puntos desde archivo.

**6.2.1.2 Tipo de Algoritmo.** Esta opción se encuentra en el menú principal "Tipo de Algoritmo" (Ver figura 15) presenta 4 opciones para escoger:

- Algoritmo Genético Clásico.
- Algoritmo Genético Sub\_Mejorado.

- Algoritmo Genético Mejorado
- Mínimos Cuadrados.

El criterio de terminación para todos los Algoritmos Genéticos es el mismo y puede ser por convergencia de los parámetros buscados, por convergencia de los errores generados o por número de iteraciones.

Cuando se presiona sobre cualquier opción esta se selecciona automáticamente. A continuación se describirá la manera de cómo está diseñado cada tipo de algoritmo y como funciona.

**6.2.1.2.1 Algoritmo Genético Clásico.** Este algoritmo está diseñado con la filosofía clásica, esta compuesto por:

- Un cromosoma de 2 genes cada uno con 10 bits, donde se codifican los valores de los parámetros  $a$  y  $b$  de la ecuación de la recta.
- Realiza una generación de población inicial uniformemente distribuida en los valores de los intervalos  $(a_{\min}, a_{\max})$  y  $(b_{\min}, b_{\max})$ , ya sea definidos por el usuario o los estimados por la aplicación dependiendo del conjunto de puntos introducidos.
- La selección se realiza utilizando la técnica de selección por torneo binario.
- El cruzamiento se realiza con un factor de cruce del 80% y se utiliza la técnica de cruce básico.
- La mutación es al azar con una probabilidad del 3%.

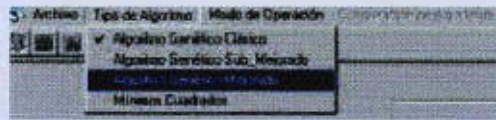
**6.2.1.2.2 Algoritmo Genético Sub\_Mejorado.** Este algoritmo posee algunas de las mejoras propuestas pero conserva algunas características del Algoritmo Genético Clásico, este presenta:

- Un cromosoma de 2 genes cada uno con 10 bits, donde se codifican los valores de los parámetros  $a$  y  $b$  de la ecuación de la recta.
- Primero realiza una generación uniforme en el intervalo establecido para los parámetros de la ecuación de la recta, y posteriormente realiza una segunda estimación de los intervalos de los parámetros y vuelve a realizar una generación uniforme de la población inicial.
- La técnica de selección es la técnica propuesta de la generación por promedio.
- El cruce es idéntico al utilizado por el Algoritmo Clásico, con la excepción que no presenta probabilidad de cruce, es decir todos los individuos seleccionados son reproducidos.
- La mutación es dirigida con probabilidad de mutación del 3%.
- 

**6.2.1.2.3 Algoritmo Genético Mejorado.** Este presenta las mejoras propuestas a los Algoritmos Genéticos Clásicos:

- Trabaja sin una codificación del conjunto de parámetros, trabaja con el conjunto de parámetros directamente.
- Realiza inicialmente una generación uniforme de la población inicial en el conjunto de intervalos establecidos para los parámetros.
- Posteriormente realiza generaciones secundarias de la población, utilizando una distribución Gaussiana.
- La técnica de selección utilizada es el método del torneo binario, al igual que el Algoritmo Genético Clásico.
- Para el cruzamiento se utiliza la técnica de cruce ponderado de la aptitud, que se diseñó para este tipo de algoritmos sin codificación de parámetros.

- Utiliza mutación dirigida, con la técnica de mutación por intervalo y se realiza en las últimas generaciones cuando el Algoritmo ha convergido.



**Figura 15. Menú de Algoritmos.**

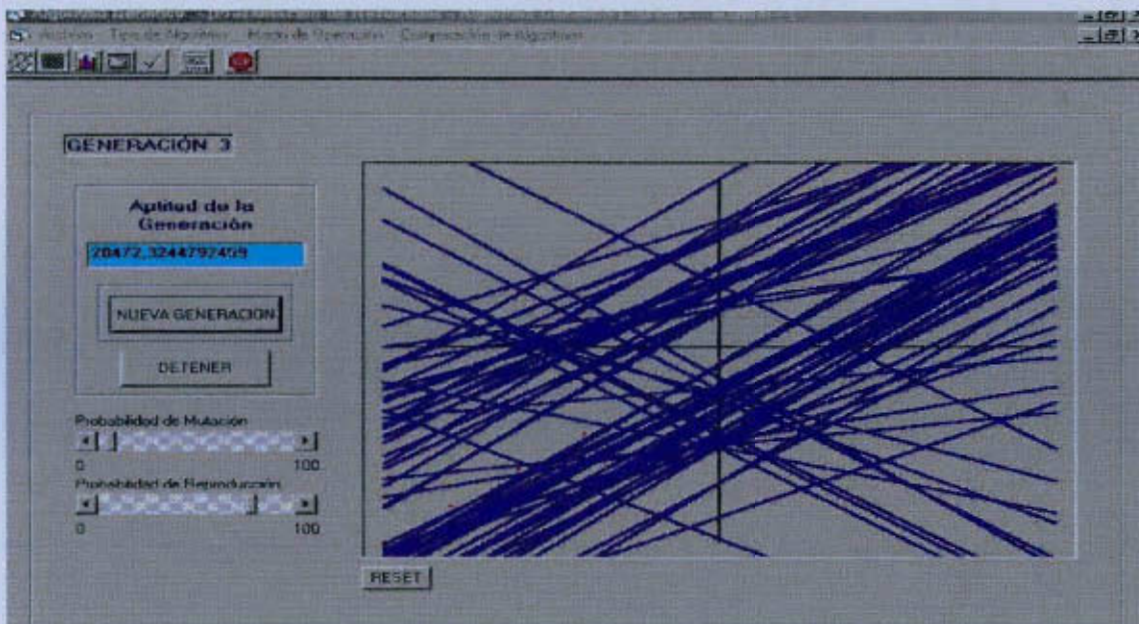
**6.2.1.2.4 Mínimos Cuadrados.** Esta parte activa la opción para ver la aproximación de los puntos por mínimos cuadrados (Botón 6 de la Barra de opciones). El Algoritmo crea el sistema de ecuaciones teniendo en cuenta el error en la coordenada "Y", y resuelve dicho sistema utilizando la eliminación Gaussiana. Este Algoritmo no hace parte de la comparación debido a que el criterio de evaluación del error es diferente al de los demás algoritmo, pero se incluye debido a que es un método antiguo y utilizado comúnmente para resolver este tipo de problemas.

**6.2.1.3 Modo de Operación.** Esta opción del menú permite escoger el modo de operación con que se quiere ejecutar el algoritmo.

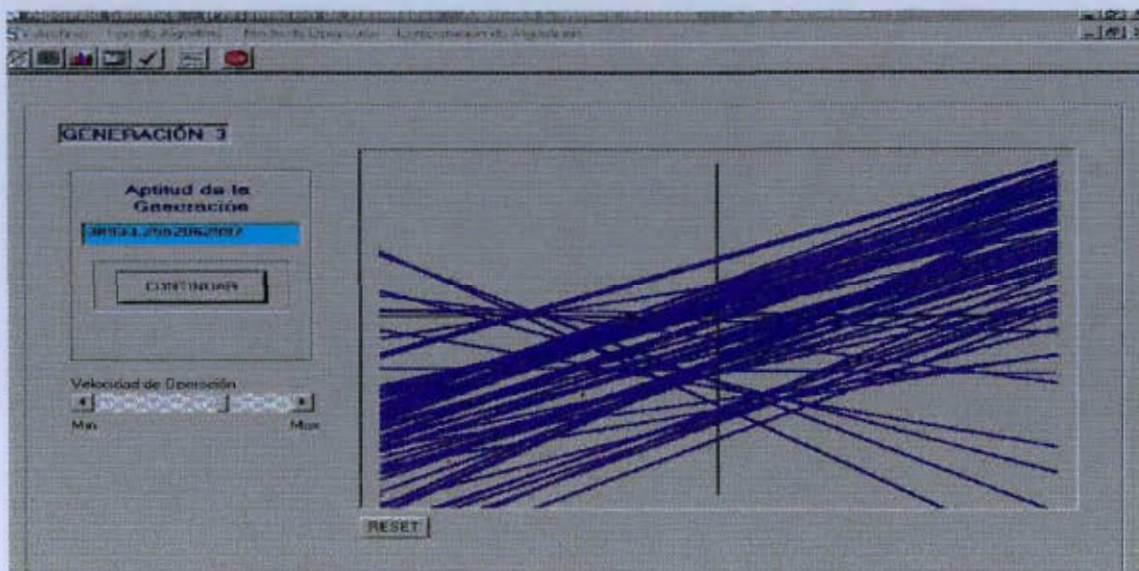
Existen tres tipos de operar el Algoritmo.

**6.2.1.3.1 Manual.** En este modo de operación el usuario puede ejecutar paso a paso la aplicación y ver el desarrollo del comportamiento del algoritmo (Ver Figura 16) presionando el botón con etiqueta "*Nueva Generación*", el botón "*Detener*" activa el botón de la barra de opciones para poder ver los mejores cromosomas de cada generación. Además para los tipos de Algoritmo Clásico y Sub\_Mejorado se pueden variar interactivamente las probabilidades de mutación y reproducción.

**6.2.1.3.2 Semi-Automático.** En este modo, el algoritmo se va ejecutando paso a paso pero sin que el usuario deba estar presionando ningún botón, aunque se puede detener en cualquier instante y ver los resultados obtenidos hasta ese momento y si desea puede continuar con la ejecución del algoritmo. Por otro lado el usuario también puede graduar a su gusto la velocidad de operación del algoritmo (Ver figura 17).

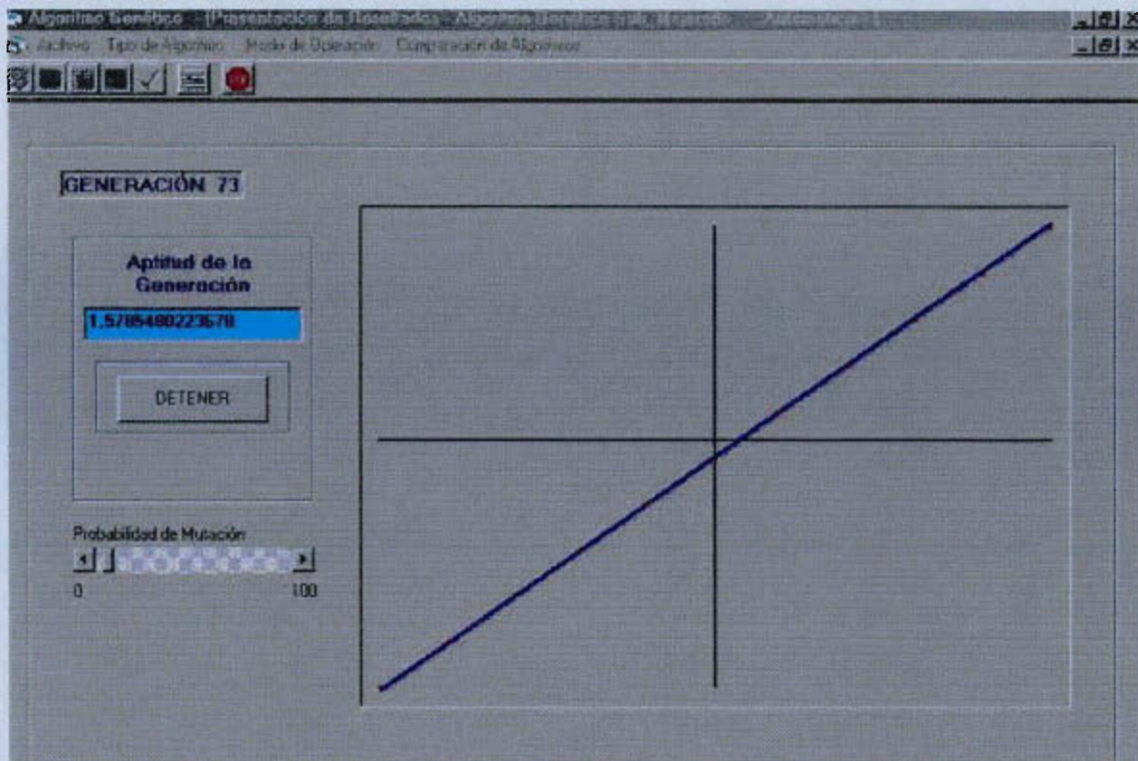


**Figura 16. Modo de Operación Manual para un Algoritmo Genético Clásico.**



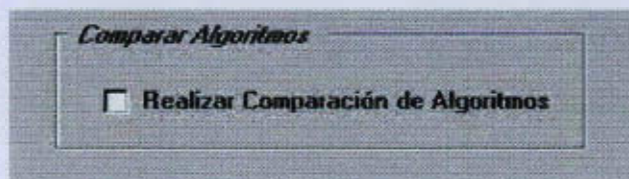
**Figura 17. Modo de Operación Semi-Automático para un Algoritmo Genético Mejorada.**

**6.2.1.3.3 Automático.** En esta modalidad a diferencia de los anteriores modos, el usuario no puede ver la ejecución paso a paso del Algoritmo, sino que por el contrario se muestran los resultados de la última generación y también se pueden ver los mejores cromosomas de cada generación así como el número de generaciones utilizadas para la convergencia del Algoritmo (Ver figura 18).



**Figura 18. Modo de Operación Automático para un Algoritmo Genético Sub\_Mejorado.**

**6.2.1.4 Módulo de Comparación de Algoritmos.** Esta es una opción que se selecciona en la parte del ingreso de datos, si se escoge esta opción se activa un menú para seleccionar entre cuales algoritmos se desea realizar la comparación (Ver Figura 19).



**Figura 19. Selección de Comparar Algoritmos**

Después que se seleccione cuales Algoritmos se desea comparar y se tengan cargados el conjunto de puntos, se presiona aceptar y después de unos segundos se presenta un formulario con la información obtenida de la comparación entre algoritmos, mostrando el tiempo gastado en la ejecución, el número de generaciones que se emplearon y el error que se obtuvo con cada tipo de algoritmo para el conjunto de puntos escogido anteriormente esta información se almacena en una bases de datos llamada Datps\_AG, también se muestran las rectas resultantes con cada Algoritmo (Figura 20).

**6.2.1.5 Barra de Opciones.** En esta barra se encuentran las opciones complementarias de la aplicación, los dos primeros botones son para ir al módulo de ingreso de datos y para comenzar la ejecución del algoritmo respectivamente; cuando el usuario se encuentra en la parte de ejecución del Algoritmo los tres siguientes botones sirven para ver la información de los cromosomas de la generación actual como valor del cromosoma, los parámetros  $a$  y  $b$ , y las aptitudes, además de una información gráfica de los errores y la información referente a los mejores cromosomas de cada generación hasta ese momento. El botón número 6 se activa cuando se ha seleccionado Mínimos Cuadrados del menú de "Tipo de Algoritmo"; el último botón es el de terminar la aplicación (Ver Figura 21).

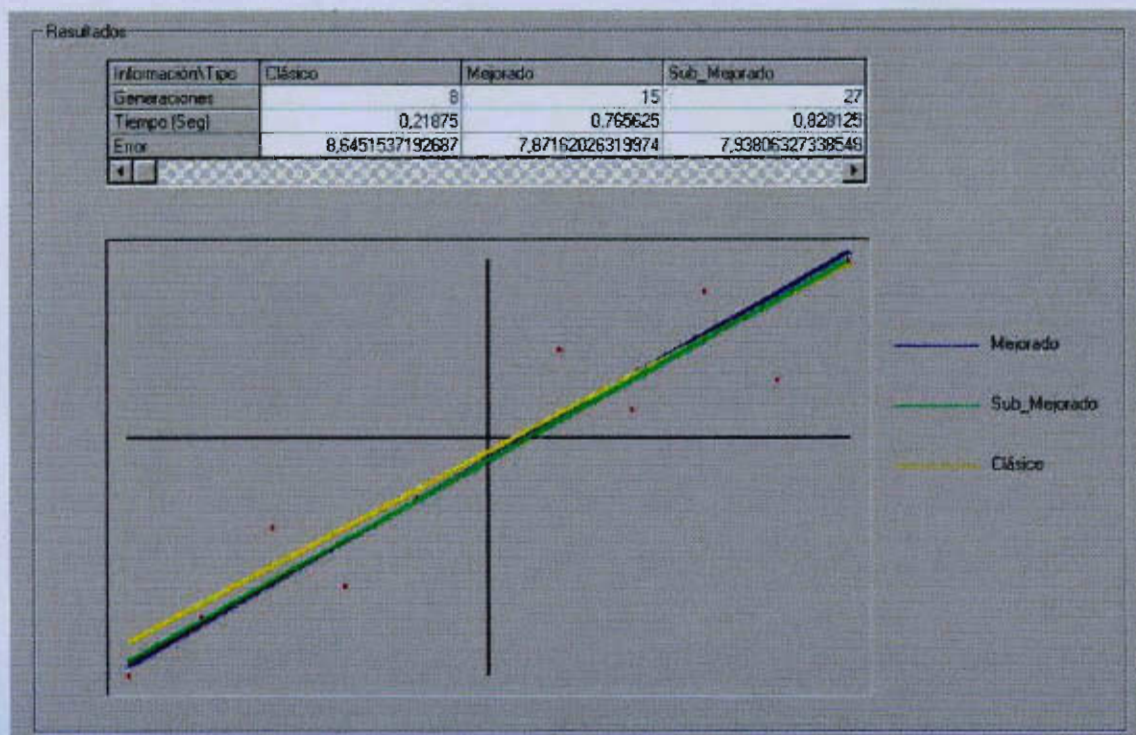


Figura 20. Resultados Obtenidos de Comparar los tres Algoritmos Genéticos.



Figura 21. Barra de Opciones.

### 6.3 RESULTADOS OBTENIDOS AL REALIZAR LA COMPARACIÓN ENTRE ALGORITMOS

Para realizar la comparación entre algoritmos se determinaron tres criterios básicos:

- Tiempo transcurrido en la ejecución del Algoritmo.
- Error en la respuesta final (Sumatoria de los errores euclidianos al cuadrado de los puntos a la recta).
- Número de Generaciones o Poblaciones utilizadas para alcanzar la respuesta.

Se realizaron pruebas de comparación entre algoritmos a 10 diferentes conjuntos de puntos (Ver Anexo C), y se tomaron varias muestras para un mismo conjunto de puntos - esto debido a la naturaleza aleatoria de los Algoritmos Genéticos -. Parte de los datos recopilados se muestran en la Tabla 3 (Para ver la totalidad de los datos recopilados ir al archivo Análisis\_AG.xls).

#### **6.4 ANÁLISIS DE RESULTADOS**

Una vez la información fue recopilada, se procedió con el análisis de los resultados; primero se promediaron y se sacaron las desviaciones estándar de los resultados obtenidos de los diferentes criterios de comparación para cada conjunto de puntos con cada Algoritmo Genético (Tabla 3). Posteriormente se realizó un análisis comparativo entre el Algoritmo Genético Mejorado y los otros dos Algoritmos Genéticos.

EL Algoritmos Genético Mejorado superó ampliamente a los otros dos tipos de Algoritmos Genéticos; en las pruebas que se realizaron:

- Siempre fue más rápido, produjo el menor tiempo para dar la respuesta.
- Más exacto, es decir produjo el menor error en la respuesta obtenida.
- Más preciso, las desviaciones estándar de las muestras de cada criterio de comparación fueron más pequeñas que la de los otros Algoritmos.
- Convergió siempre con un mínimo número de Generaciones. Esto se traduce en un ahorro de tiempo y de recursos.

Adicionalmente cuando todos los puntos pasaban por una recta el Algoritmo Genético Mejorado fue mucho más exacto, que los otros algoritmos, produciendo un error muy

pequeño del orden de  $1 \cdot 10^{-10}$  en promedio, pero cuando el conjunto de puntos era disperso aunque el Algoritmo Genético Mejorado producía el menor error, la diferencia con los otros Algoritmos no fue muy significativa en cuanto al error producido.

Los porcentajes de mejoramiento del Algoritmo Mejorado con respecto a los otros Algoritmos, para cada conjunto de puntos se muestran en la Tabla 4.

En General el Algoritmo Genético fue aproximadamente 20% más rápido, tuvo una disminución del 70% en el número de Generaciones y fue 40% más exacto que el Algoritmo Genético Clásico. Estos resultados junto con los de comparación del Algoritmo Genético Sub\_Mejorado se ilustran en la Tabla 5.

Tabla 3. Recopilación de Datos de la Comparación entre los Algoritmos

		Algoritmo Genético Clásico			Algoritmo Sub_Mejorado			Algoritmo Genético Mejorado		
Nombre Puntos	Tiempo	Error	Generaciones	Tiempo	Error	Generaciones	Tiempo	Error	Generaciones	
Puntos_8,75_-3,25.pag	0,5	0,000380391	24	1,5898438	0,11521933	75	0,44140625	5,87222E-10	14	
	0,4414063	0,001500934	23	1,4296875	0,13546964	67	0,37890625	1,0215E-09	13	
	0,4375	0,000380391	23	1,1523438	0,03598863	53	0,44140625	7,5038E-10	13	
	0,4414063	0,000421761	22	1,4882813	0,00205957	70	0,4296875	3,63974E-10	14	
	0,6015625	0,000380391	27	1,4804688	2,0017E-05	72	0,38671875	5,20061E-10	13	
	0,609375	0,000380391	29	1,4804688	0,00048515	71	0,37890625	5,30427E-10	13	
Promedios	<b>0,8952092</b>	<b>0,000874943</b>	<b>24,86888667</b>	<b>1,436849</b>	<b>0,04829798</b>	<b>64</b>	<b>0,40950521</b>	<b>6,28928E-10</b>	<b>13,33333333</b>	
Desviaciones	<b>0,0811021</b>	<b>0,000454384</b>	<b>2,732520204</b>	<b>0,1488977</b>	<b>0,06161762</b>	<b>7,79743848</b>	<b>0,03109499</b>	<b>2,29068E-10</b>	<b>0,516397779</b>	
Puntos_1_0.pag	Tiempo	Error	Generaciones	Tiempo	Error	Generaciones	Tiempo	Error	Generaciones	
	0,8203125	0,031948991	29	1,2109375	0,59658228	40	0,65625	5,80313E-10	14	
	0,8203125	0,015414538	28	1,9804688	0,00081886	66	0,6015625	6,83045E-11	13	
	0,7109375	0,023760706	24	2,140625	0,19490252	72	0,609375	1,97355E-10	14	
	1,5273438	0,002057922	52	1,4921875	0,01255895	49	0,609375	5,34761E-10	13	
	1,0390625	0,001856516	35	1,7109375	0,0078883	58	0,66015625	7,27285E-12	14	
	0,8789063	0,000397893	30	2,9809375	0,00152729	100	0,66015625	6,02302E-10	13	
Promedios	<b>0,9651458</b>	<b>0,012873781</b>	<b>33,19160168</b>	<b>0,13771384</b>	<b>0,16666667</b>	<b>64,16666667</b>	<b>0,6328125</b>	<b>3,31718E-10</b>	<b>13,5</b>	
Desviaciones	<b>0,2951395</b>	<b>0,013283909</b>	<b>9,959919678</b>	<b>0,6107844</b>	<b>0,23816473</b>	<b>20,9801493</b>	<b>0,02870496</b>	<b>2,71642E-10</b>	<b>0,547722558</b>	
Puntos_A_1.pag	Tiempo	Error	Generaciones	Tiempo	Error	Generaciones	Tiempo	Error	Generaciones	
	0,6015625	81,45922602	28	1,3203125	81,4847952	64	0,48828125	81,45342659	15	
	0,6601563	81,47626756	32	1,4296875	81,453489	61	0,4921875	81,4534266	15	
	0,8320313	81,45437415	39	0,9296875	81,493056	45	0,48828125	81,45342659	14	
	0,5507813	81,4547224	26	2,03125	81,4750166	100	0,546875	81,4534266	15	
	0,4921875	81,4547224	25	1,2070313	81,4536887	57	0,5	81,4534266	14	
Promedios	<b>0,6373438</b>	<b>81,4546628</b>	<b>30,1383938</b>	<b>0,1729091</b>	<b>81,4729091</b>	<b>65,4</b>	<b>0,503125</b>	<b>81,4534266</b>	<b>14,6</b>	
Desviaciones	<b>0,1301808</b>	<b>0,009387418</b>	<b>5,700877125</b>	<b>0,4070602</b>	<b>0,01780708</b>	<b>20,8470337</b>	<b>0,02492053</b>	<b>1,3487E-06</b>	<b>0,547722558</b>	

Tabla 4. % de Mejoramiento en cada conjunto de puntos.

% De Mejoramiento del AG Mejorado con respecto a los Otros Algoritmos Genéticos en cada conjunto de Puntos			
Puntos	Tiempo	Error	Generaciones
<b>Puntos_1,75_-3,25.psg</b>			
AG Clásico	18,94%	99,9999%	45,95%
AG SubMejorado	71,50%	100,00%	80,39%
<b>Puntos_1_0.psg</b>			
AG Clásico	34,50%	100,00%	59,09%
AG SubMejorado	66,97%	100,00%	78,96%
<b>Puntos_A_1.psg</b>			
AG Clásico	19,80%	0,01%	51,33%
AG SubMejorado	63,64%	0,02%	77,68%
<b>Puntos_A_2.psg</b>			
AG Clásico	14,22%	0,01%	47,90%
AG SubMejorado	66,36%	0,01%	79,58%
<b>Puntos_A_3.psg</b>			
AG Clásico	25,50%	0,0009%	51,63%
AG SubMejorado	65,53%	0,0005%	77,71%
<b>Puntos_A_4.psg</b>			
AG Clásico	19,35%	1,36%	44,23%
AG SubMejorado	58,75%	0,16%	72,81%
<b>Puntos_XoR.psg</b>			
AG Clásico	0,64%	0,0001%	37,50%
AG SubMejorado	56,67%	0,0007%	72,83%
<b>Puntos_XoR_1.psg</b>			
AG Clásico	14,34%	0,09%	47,42%
AG SubMejorado	62,35%	0,51%	78,57%
<b>2Puntos.psg</b>			
AG Clásico	25,00%	93,40%	30,11%
AG SubMejorado	45,23%	63,05%	55,43%
<b>Puntos_-1_0.psg</b>			
AG Clásico	37,09%	100,00%	58,44%
AG SubMejorado	69,63%	100,00%	80,49%

Tabla 5. % de Mejoramiento del AG Mejorado y los otros dos AG.

% De Mejoramiento del AG Mejorado con respecto a los Otros Algoritmos Genéticos en General			
	Tiempo	Error	Generaciones
AG Clásico	20,94%	39,48%	70,01%
AG SubMejorado	62,78%	38,38%	75,44%

## 7 USO DE LOS ALGORITMOS GENÉTICOS MEJORADOS

Una vez se probaron las mejoras propuestas y después de comprobar su efectividad, los Algoritmos Genéticos Mejorados, se aplicaron a dos problemas prácticos de ingeniería:

- Creación de una Herramienta Solver.
- Ajuste de los Parámetros de la Ecuación de Estado de Peng-Robinson a un problema específico.

### 7.1 CREACIÓN DE UN SOLVER

**7.1.1 Definición del Problema.** Existen ecuaciones para modelar procesos, comportamientos y en general cualquier cosa que sea de nuestro interés. Muchas veces estas ecuaciones resultan muy complicadas y de difícil manejo. Cuando se tiene el problema de encontrar el valor de una variable que no se puede despejar explícitamente dentro de una ecuación – por ejemplo la variable "T" dentro de la ecuación 22 -, se recurre a utilizar métodos iterativos de aproximación para encontrar el valor de esta variable, y en general de cualquier variable dentro de la ecuación, la única condición es que se debe asignar un valor a las demás variables de la ecuación.

Con los Algoritmos Genéticos se pretende dar un nuevo enfoque y metodología para resolver este problema.

Se diseñó un Algoritmo Genético Mejorado para resolver este problema y se implementó dentro de una aplicación llamada *Equation Solver* (Ver figura 22), la cual se explicará a continuación.

$$Y = \frac{(T + X^r)}{T}$$

**Ecuación 22. Variable T que no se puede despejar de la ecuación.**

**7.1.2 Implementación de la Aplicación.** Para resolver el problema descrito anteriormente se implementó una aplicación, con entorno gráfico similar al de una calculadora y consta de tres partes, donde el usuario puede introducir la ecuación que desea resolver, posteriormente introduce los valores de las variables dejando vacía la variable que desea despejar y por último utilizando un algoritmo genético mejorado la aplicación encuentra una respuesta para la variable buscada.

**7.1.2.1 Ingreso de la Ecuación.** El usuario puede introducir la ecuación utilizando el entorno gráfico que le brinda la aplicación (Ver Figura 23), o puede cargarla de un archivo existente (Ver figura 24). En general el tipo de ecuaciones que soporta la aplicación es muy diverso y variado, cualquier ecuación que el usuario desee introducir lo puede hacer, la única restricción que presenta es que la ecuación de la forma debe tener la forma de la ecuación 23, donde debe haber sólo una variable en función de las demás. Además la aplicación cuenta con un sistema de detección de errores de sintaxis, que si ocurren serán comunicados al usuario.

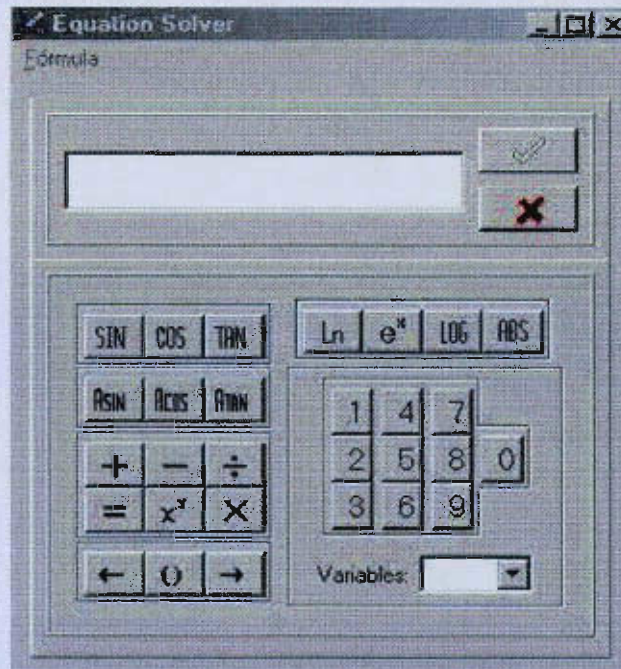


Figura 22. Equation Solver.

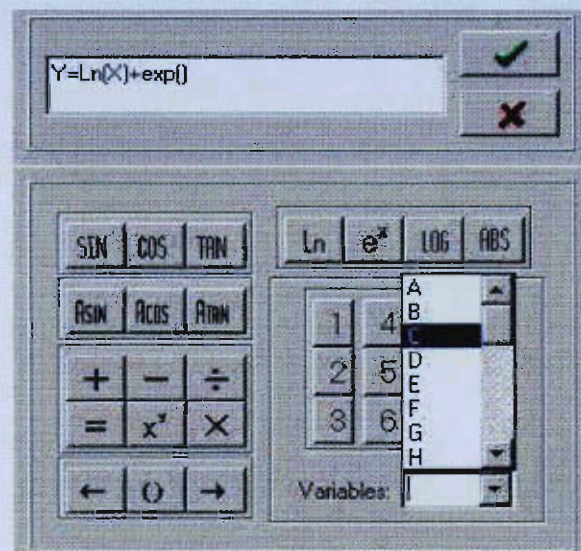


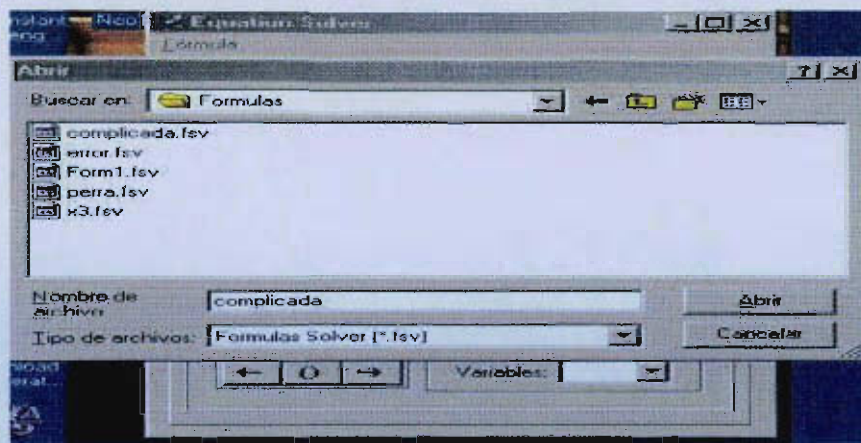
Figura 23. Ingreso de la Ecuación manualmente.

**7.1.2.2 Ingreso de Valores a las variables.** Una vez el usuario ha introducido la ecuación sin errores de sintaxis, y presionar el botón de aceptar aparecerá un modulo donde se muestran las variables de la ecuación y la posibilidad de

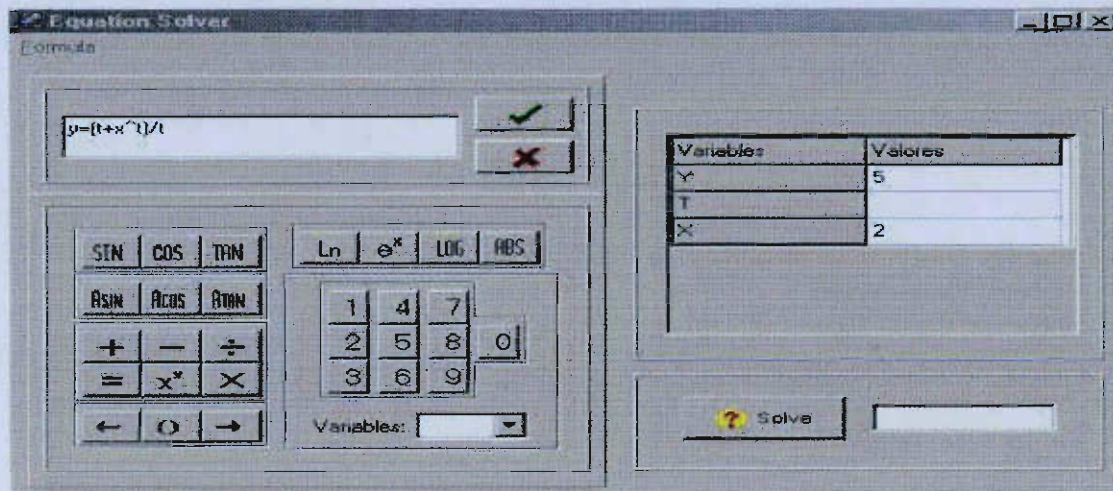
introducirlas los valores respectivos (Ver figura 25), teniendo en cuenta que se debe dejar sólo una variable vacía, de lo contrario la aplicación mostrará un mensaje de error.

$$\text{Variable} = \text{Expresión}$$

**Ecuación 23. Formato de las formulas Equation Solver**



**Figura 24. Cargar Formula desde Archivo.**



**Figura 25. Ingreso de Valores a la Variables.**

**7.1.2.3 Respuesta.** Después de haber introducidos los valores a las variables, y haber dejado sólo una vacía la aplicación esta lista para encontrar la solución. El funcionamiento del algoritmo se describe a continuación:

- La Primera parte se encarga de realizar una estimación del intervalo donde se encuentra la respuesta. Se parte buscando inicialmente en el intervalo  $(0, 1)$ , con una división del intervalo de 10 puntos. Si la respuesta no se encuentra en éste intervalo, se desplazan los límites del intervalo ya sea hacia delante o hacia atrás y se aumenta su longitud en múltiplos de 10 y conservando la división de 10 puntos (Ver Figura 26). Para definir hacia donde se debe desplazar el intervalo, se tiene en cuenta el criterio del mínimo error, es decir en el extremo donde se produzca el error más pequeño hacia ese se desplaza el intervalo.
- Después de encontrar y definir el intervalo de búsqueda de la respuesta, se procede a aplicar el algoritmo genético mejorado, este incluye:
  - Cromosoma Variable (Intervalo Variable).
  - Generaciones secundarias con distribución Gaussiana.
  - Selección por Promedio.
  - Mutación por Intervalo.

La respuesta se da con una precisión de  $1 \cdot 10^{-15}$ ; cabe anotar que si la variable incógnita puede tomar varios valores el algoritmo sólo encontrará uno, esto dependiendo de que tan cerca del intervalo  $(0, 1)$  se encuentre la respuesta. Por otro lado si la solución no tiene un valor real, la aplicación devolverá un mensaje de error como el de la figura 27.

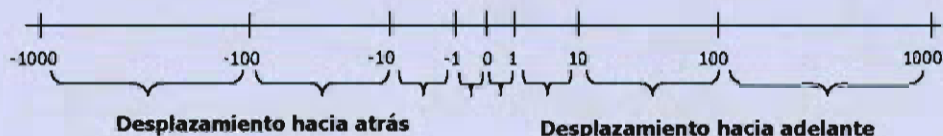


Figura 26. Búsqueda del Intervalo de respuesta.

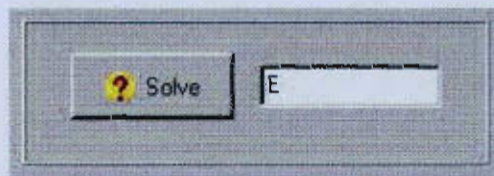


Figura 27. Mensaje de que no se encontró la respuesta.

**7.1.3 Resultados.** Para comprobar la eficiencia y eficacia del solver desarrollado se confrontó, con el solver que viene incorporado en la calculadora HP 48 GX, produciendo resultados muy similares en cuanto a precisión en la respuesta e igual comportamiento en las ecuaciones donde la variable no tenía un valor real.

En la siguiente tabla se muestran algunas ecuaciones evaluadas y los respectivos resultados con cada método.

Tabla 6. Comparación entre Herramientas Solver.

ECUACION	VALORES DE LAS VARIABLES				METODOS	
	Y	T	X		HP 48GX	AGM
$Y=(t+x^t)/t$	5	-3			E	E
	6,7		3,21		0,229201069	0,229201068
	-1		4,5		-0,312495629	-0,31249563
	-12,6		5,9		-0,06503499	-0,06503499
$y=\exp(x)-t$	Y	X	T			
	0		1		0	0
	-4		6		0,693147181	0,693147177
	3,75	-8			-3,749664537	-3,749664542
$y=t*\exp(t)-8*I+h$	Y	T	L	H		
	100		5	37	3,408466504	3,40846651
	50		-7	-20	1,964049113	1,964049091
	-15	3	5		-35,25661077	-35,25661075
	-27	-0,5		-6	2,587091834	2,587091843
$Y=(f+2)^r/r+s$	Y	F	R	S		
	3,78	1,2		32	-0,034059467	-0,034059467
	-8,9	2,5		-4	-0,160347718	-0,160347717
	1567	1,34		-1200	0,00036156	0,000361559
	2500	5	2,5		2448,143274	2448,143274
	2500	5	-3,7		2500,000202	2500,000202
	-569	3,6	-2,9		-568,9976673	-568,9976672
15		2	5	2,472135955	2,472135962	
34,7		3,8	3	1,528500847	1,528500842	

## 7.2 AJUSTE DE LOS PARÁMETROS DE LA ECUACIÓN DE ESTADO DE PENG-ROBINSON A UN PROBLEMA ESPECÍFICO

**7.2.1.Marco Teórico.** Una Ecuación de Estado (EOS) es la relación entre presión, volumen y temperatura, la cual permite estudiar el comportamiento físico de una sustancia con respecto a las variables mencionadas.

La ecuación de estado más conocida es la de los gases ideales:

$$P \cdot V = R \cdot T$$

**Ecuación 24. EOS para Gases Ideales.**

Donde :

- P: Presión absoluta a la que se encuentra el gas.
- V: Volumen que ocupa una mol de gas a las condiciones de P y T.
- R: Constante Universal de los gases.
- T: Temperatura absoluta a la cual se encuentra el gas.

La ecuación 23 tiene poca aplicación práctica pues rara vez un gas se comporta como ideal; esto hizo necesario que se buscara el desarrollo de una ecuación de estado (EOS) para gases reales.

Una de las ecuaciones de estado más usadas en la industria del petróleo en el estudio del comportamiento de fases, es la ecuación de Peng-Robinson la cual presenta la siguiente forma:

$$\left( P + \frac{a(T)}{V(V+b) + b(V-b)} \right) \cdot (V-b) = RT$$

**Ecuación 25. Ecuación de Peng-Robinson.**

Donde a y b son los parámetros de la ecuación.

$$a = a_c \cdot \alpha(T)$$

**Ecuación 26. Expresión para el cálculo del parámetro a.**

$$b = 0,07780 \cdot \frac{RT_c}{P_c}$$

**Ecuación 27. Expresión para el cálculo del parámetro b.**

Donde:

- $P_c$ : Presión Crítica de la Sustancia.
- $T_c$ : Temperatura Crítica de la Sustancia.
- $\alpha$ : Factor de Escalamiento.

Para aplicar la ecuación de Peng-Robinson a mezclas se debe conocer la composición de éstas. La composición de mezclas de hidrocarburos se da en fracción molar de los componentes más livianos y de una fracción pesada o fracción C7+ pues generalmente en el análisis que se hace a los hidrocarburos se identifica hasta el componente hexano.

Las propiedades físicas de los compuestos puros se pueden obtener de tablas existentes en la literatura, pero la fracción pesada es una mezcla indefinida a la cual se le deben conocer sus propiedades críticas (Presión, Volumen) y el factor acéntrico ( $\omega$ )<sup>13</sup> para poder calcular los parámetros a y b de la ecuación de estado para la mezcla.

Las principales dificultades que se presentan, para aplicar una ecuación de estado a una mezcla de hidrocarburos reales son:

- Manejo de la Fracción pesada.
- Obtención de los coeficientes de interacción binaria ( $K_{ij}$ ).

---

<sup>13</sup> El Factor acéntrico es una medida de la desviación, del Principio de los Estados Correspondientes, de una sustancia dada; es característico de cada sustancia, y aumenta con el peso molecular de ésta.

**7.2.2 Definición del Problema.** Las ecuaciones de estados son muy importantes en el estudio de Fases, una de las más utilizadas debido a su sencillez y aproximación en los resultados producidos es la Ecuación de Peng-Robinson. Uno de las propiedades físicas de la fracción pesada que más influye en el resultado de la ecuación de estado de Peng-Robinson, es el coeficiente de interacción binaria entre el Metano (C1) y la fracción pesada (C7+) ( $K_{C1-C7+}$ ), por ello se debe tener especial cuidado en el cálculo de dicho parámetro; otro parámetro de menor peso dentro del comportamiento de la ecuación de estado pero igual de importante es el factor acéntrico de la fracción pesada ( $W_{C7+}$ ).

Para resolver este problema se desarrolló una nueva metodología con Algoritmos Genéticos para ajustar el coeficiente ( $K_{C1-C7+}$ ) y el factor acéntrico ( $W_{C7+}$ ) para un gas condensado a una temperatura de 190 F, y cuya composición se muestra en la Tabla 7.

**7.2.3 Implementación de la Aplicación.** Para ajustar el coeficiente ( $K_{C1-C7+}$ ) y el factor acéntrico ( $W_{C7+}$ ) del gas condensando, se implementó un aplicación para encontrar los parámetros óptimos para este gas en particular, tomando como base la metodología utilizada en la aplicación PVTLAB, desarrollada por el Ingeniero de Petróleos Fernando Calvete (Para mayor información sobre esta aplicación y metodología refiérase al anexo D).

La aplicación tiene incluida la metodología utilizada en PVTLAB, para hallar los parámetros de la Ecuación de Peng-Robinson para éste gas, posteriormente se optimizan estos parámetros utilizando la metodología con Algoritmos Genéticos.

La Primera parte de la aplicación muestra los resultados de la fracción de volumen obtenidos por las dos metodologías y las confronta con los datos experimentales (Ver figura 28).

La segunda parte de la aplicación, presenta una comparación estadísticas de los datos obtenidos por las dos metodologías y los datos experimentales, además de graficar los datos para una mejor visualización de los resultados (Ver Figura 29).

**Tabla 7. Composición del Gas.**

<b>Composición del Gas 2</b>	
<b>Componente</b>	<b>Fracción</b>
<b>N2</b>	<b>0,0042</b>
<b>CO2</b>	<b>0,0061</b>
<b>H2S</b>	<b>0,0004</b>
<b>C1</b>	<b>0,5749</b>
<b>C2</b>	<b>0,1345</b>
<b>C3</b>	<b>0,0752</b>
<b>I-C4</b>	<b>0,02075</b>
<b>N-C4</b>	<b>0,02075</b>
<b>I-C5</b>	<b>0,01165</b>
<b>N-C5</b>	<b>0,01165</b>
<b>N-C6</b>	<b>0,0179</b>
<b>C7+</b>	<b>0,122</b>

**7.2.4 Resultados.** Los valores calculados por la aplicación corresponden a la fracción de volumen contra la presión que está dada en psig, (Tabla 8), además se comparó la eficiencia de las metodologías tomando medidas estadísticas como la desviación absoluta promedio, el error relativo entre los puntos obtenidos y los experimentales, y el porcentaje de error entre las presiones de saturación calculadas y la experimental (Tabla 9).

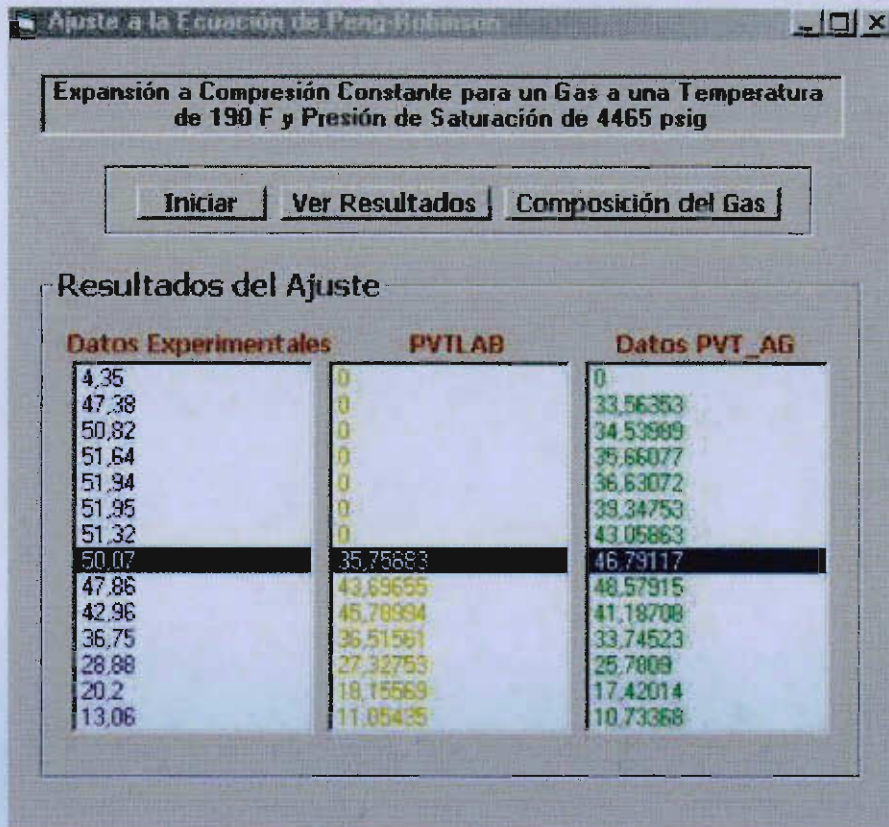


Figura 28. Presentación de Resultados.

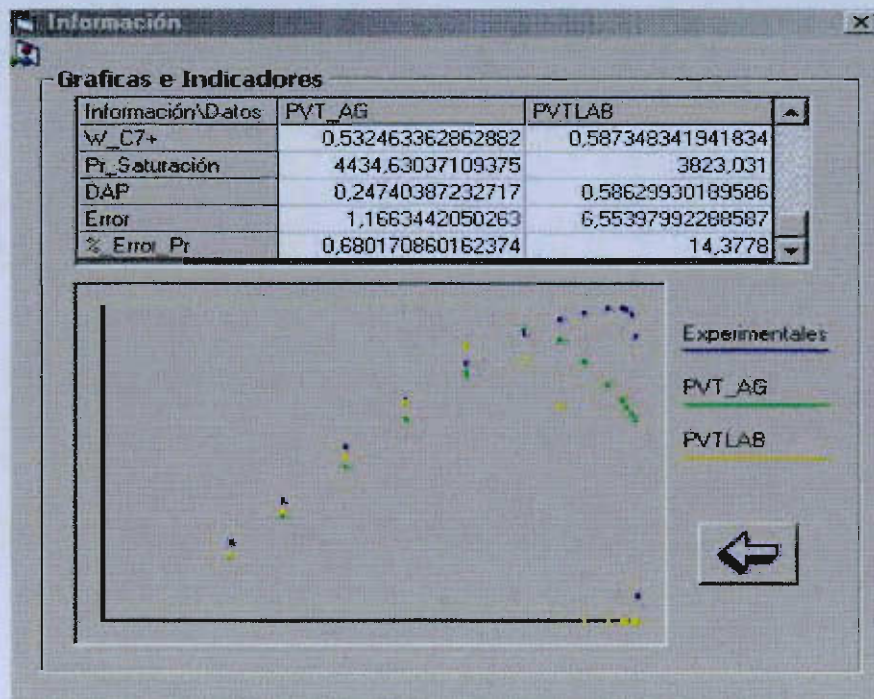


Figura 29. Comparación Grafica y Estadística de los Resultados.

Tabla 9. Comparación Estadística entre Metodologías.

	Experimental	PVTLAB	AG_M
$K_{C1-C7+}$		0,061441	0,10366484
$W_{C7+}$		0,58734834	0,53246336
Presión Sat	4465	3823,04	4434,63
DAP		0,5862993	0,24740387
Error Relativo		6,55397992	1,16634421
% Error_Psat		14,3778	0,68017086

Se puede concluir que el coeficiente de interacción ( $K_{C1-C7+}$ ), presenta una mayor sensibilidad en los cálculos de la ecuación de Peng-Robinson para el gas condensado (Tabla 7) que el factor acéntrico ( $W_{C7+}$ ).

Tabla 8. Datos Obtenidos por los dos métodos de la Aplicación.

Presión (psig)	% Fracción de Volumen		
	Experimental	PVTLAB	AG_M
4440	4,35	0	0
4420	47,38	0	33,56353
4385	50,82	0	34,53989
4339	51,64	0	35,66077
4300	51,94	0	36,63072
4180	51,95	0	39,34753
3993	51,32	0	43,05863
3780	50,07	35,75683	46,79117
3490	47,86	43,69655	48,57951
2998	42,96	45,78994	41,18708
2805	36,75	36,51561	33,54723
2000	28,88	27,32753	25,7809
1485	20,2	18,15569	17,42014
1058	13,06	11,05435	10,73368

## 8 CONCLUSIONES

- En los últimos años los Algoritmos Genéticos han dado o por lo menos plantean una alternativa de solución a diversos problemas del mundo real, a tal punto que su campo de acción se ha extendido considerablemente y se pueden encontrar aplicaciones en áreas como Medicina, Biología Molecular, Robótica, Criminología entre otras.
- Los Algoritmos Genéticos se pueden aplicar en diversas áreas, especialmente donde no existen técnicas o metodologías especializadas para la solución de un problema determinado.
- Los Algoritmos Genéticos presentan gran flexibilidad para trabajar conjuntamente con otros métodos o técnicas, creando algoritmos híbridos, para la solución de problemas específicos.
- No existe un Algoritmo Genético de Propósito general, es decir un único Algoritmo que se pueda aplicar a problemas de diversas áreas, por el contrario para cada problema se debe diseñar e implementar un Algoritmo Genético específico para resolver dicho problema.
- Cromosoma variable (Intervalo Variable), generación de la población utilizando la distribución de Gauss, mutación dirigida, fueron las mejoras que se propusieron a los

- Algoritmos Genéticos Clásicos, además de nuevas técnicas para la selección y el cruce como selección por promedio y cruce por valor ponderado de la aptitud respectivamente; la aplicación de estas mejoras produjo resultados alentadores para continuar con la investigación en el campo de mejoras a los Algoritmo Genéticos.
- EL cruce ponderado por valor de aptitud, se diseñó para ser utilizado en Algoritmos Genéticos donde el problema consiste en encontrar un valor numérico para la respuesta.
- Para probar las mejoras propuestas, se escogió el problema del ajuste de los parámetros de la ecuación de una recta ( $Y = ax + b$ ) a un conjunto de puntos; se implementó una aplicación para realizar las pruebas llamada RECT\_AG.

Los criterios para la evaluación de las mejoras fueron:

- Tiempo de proceso.
- Exactitud en la respuesta.
- Número de Poblaciones utilizadas para alcanzar la respuesta.

Bajo estos criterios en todas las pruebas realizadas, los algoritmos genéticos que incluían las mejoras propuestas superaron a los algoritmos genéticos clásicos.

- Los Algoritmos Genéticos clásicos tuvieron una disminución del 20% en los tiempos de proceso, tuvo un 40% más de exactitud en la respuesta, y la obtuvo con una disminución del 70% en el número de Poblaciones utilizado.

- Se creó un nuevo algoritmo híbrido, entre un algoritmo de búsqueda de intervalo y un algoritmo genético mejorado, para encontrar el valor de una variable dentro de una ecuación, esto se conoce como Solver.
- El nuevo algoritmo se implementó dentro la aplicación Equation Solver. Los resultados obtenidos con este algoritmo se compararon con los producidos por la Herramienta Solver que viene incorporada dentro de la calculadora HP 48GX, produciendo comportamientos y resultados muy similares en la respuesta.
- Se desarrolló una nueva metodología conjunta, entre Los Algoritmos Genéticos Mejorados y la metodología utilizada en PVTLAB, para el ajuste de los Parámetros de la Ecuación de Estado de Peng-Robinson.
- La nueva metodología se implementó dentro de una aplicación llamada PVT\_AG, produciendo una disminución del error relativo total, una mejor aproximación al valor experimental de la presión de saturación en comparación con los resultados obtenidos con PVTLAB, el único inconveniente es que la nueva metodología aumenta los tiempos de proceso considerablemente, pero en este tipo de problemas de ajuste se requiere una mayor exactitud y precisión en la respuesta.

## 9 RECOMENDACIONES

- Ya que los resultados obtenidos fueron satisfactorios, se recomienda seguir y profundizar la investigación en campos como áreas de aplicación y mejoras de los Algoritmos Genéticos.
- Se podría entrar en el campo de investigación de nuevos métodos numéricos utilizando los Algoritmos Genéticos, como por ejemplo en la búsqueda de raíces para polinomios, en la solución de sistemas de ecuaciones no lineales entre otros.
- El nuevo algoritmo creado para encontrar el valor de una variable dentro de una ecuación, se podría mejorar para que encontrara todos los valores que pudiese tomar esta variable, además de encontrar soluciones complejas.
- La aplicación RECT\_AG, se diseñó con un objetivo educativo e ilustrativo más que práctico, se podría mejorar agregándole un módulo que realizara regresiones no sólo lineales sino polinomiales.
- En el nuevo método creado para ajustar los parámetros de la ecuación de estado de Peng-Robinson, se podrían incluir muchos más parámetros para realizarles el ajuste como la Presión, Volumen y temperatura crítica de la Fracción pesada.

- La aplicación PVT\_AG podría modificarse para que realizara los ajustes y comparación a muchos más tipos de compuestos.

## BIBLIOGRAFIA

Andrade, Juan Pablo. , Naranjo Abel. Estudio de Comportamiento de Fases usando Ecuaciones de Estado. Universidad Nacional Medellín, 1994.

Bethke, A.D. Genetic Algorithms as Function Optimizers. Ph. D. Thesis, University of Michigan, Ann Arbor. 1980.

Conway, E. W. Maxwell, W. L., and Miller, L. W. Theory of Scheduling. Adisson-Wesley, Reading, Massachusets, 1967.

Chakraborty, U.K y Dastidar, D.G. Using reiliability analysis to estimate the number of generations to convergence in genetic algorithms, Information Processing Letters, 4b, 199-209, 1993.

Davis, L. Applying Adaptive Algorithms to Epistatic Domains. Proceedings of the 9<sup>th</sup> InternationalmJoint Conference on Artificial Intelligence. Morga Kauffman Publisher. Los Altos, California, 162-164, 1985.

Davis L. Genetic Algorithms and Simulated Annealing, primera edición. Van Nostrad Reinhold, 1987.

Donford, U., Pesch, E. Combining Genetic and Local Search for Solving the job shop Sheduling Problem. APMOD93 Preprints. 142-149, 1993.

Eiben, A.E, Aarts, E.H.L. y Van Hee, K.M. Global convergence of genetic algorithms: An infinite Markov chain, analysis, Computing Science Notes, Eindhoven University of Technology, The Netherlands, 1990.

Goldberg, D., Y Richardson, J. Genetic Algorithms with sharing for multimodal function optimization Genetic Algorithms and their applications : Proceedings of the second International Conference on Genetic Algorithms. Erlbaum, 1987.

Goldberg, David. Genetic Algorithms in search, optimization, and machine learning. USA: Adison Weslwy, 1989.

Goldberg, David. Los placeres existenciales de los Algoritmos Genéticos. Department of General Engineering, University of Illinois at Urbana-Champaing.

Goldberg, D. Deb, k., Kargupta, H., y Harik, g. rapid, accurate optimization of difficult problems using fast messy genetic algorithms. IliGAL Report N° 93004, Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois, Urabana, 1993.

Holland, J.H. Adaptation in Natural and Artificial Systems. University of Migihan Press, 1995.

Jeremiah, B., Lalchandai, A., y Schrage, L. Heuristic Rules Toward Optimal scheduling. Research report. Department of Industrial Engineering. Cornell University, 1964.

Jones, T., y Forrest, S. An Introduction to SFI Echo. Working paper 93-12-074, Santa Fe Institute, 1993.

Kitano, Hiroaki. Designing neural networks using genetic algorithms with graph generation system. Complex Systems. Vol 4:461-467, 1990.

Koza, Jhon. R. Genetic Programming: On the Programming of Computers by means of natural Selection. MIT Press, 1992.

Laguna, Manuel y Moscato, Pablo. "Algoritmos Genéticos", Cap. 3 del libro 'Optimización Heurística y Redes Neuronales'. Editorial Paraninfo, Madrid, España, 1996.

Lucius, C. B., y Kateman, g. Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufman, 1989.

Martínez, José J. Fundamentos de Algoritmos Genéticos, II Congreso Colombiano de Neurocomputación, Facultad de Ingeniería, Universidad Nacional de Colombia, Bogotá, 1997.

Merzs, Peter. Genetic Algorithms for Combinational Optimization Problems, Master`s Thesis, Department Of Electrical Engineering and Computer Science, University of Siegen, 1996.

Mitchell, Melanie An Introduction to Genetic Algorithms, USA: MIT Press, 1998.

Moscato, Pablo. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report 826, 1989.

Naranjo, A.A. Estudios de Comportamiento de Fases usando Ecuaciones de Estado. Universidad Nacional (Medellín) – ICP, 1993.

Nilson, Nils J. Inteligencia Artificial, Una nueva síntesis. Editorial Mac Graw Hill, Madrid España 2001.

Rudolph, C. Convergence analysis of canonical algorithms, IEEE Transactions on Neural Networks, 96-101, 1994.

O. Cordón, F. Herrera, E. Herrera-Viedma, m. Lozano. Genetic Algorithms and Fuzzy Logic in Control Processes. Technical Report N° DECSAI-95109, ETS de Ingeniería Informática, Universidad de Granada, 1995.

Syswerda, G. Schedule optimization using Genetic Algorithms. Davis (ed.), 332-349, 1991.

Valdivieso, Margarito Teran, y Otros. Análisis numérico de la Ecuación de Estado de Peng-Robinson, Instituto Mexicano del Petroleo, 1984.

Whitley, L.D. An executable model of a simple genetic algorithms. Foundations of Genetic Algorithms 2. Morgan Kauffman, 1993.

Direcciones de Internet.

[www.iamnet.com/users/jcontre/genetic/ag.htm](http://www.iamnet.com/users/jcontre/genetic/ag.htm).

[www.cs.qub.ac.uk/~M.Sullivan/ga/ga\\_index.html](http://www.cs.qub.ac.uk/~M.Sullivan/ga/ga_index.html).

[www.Tigre.aragon.unam.mx/geneticos/indice.htm](http://www.Tigre.aragon.unam.mx/geneticos/indice.htm).

[www.neo.lcc.uma.es/TutorialEA/semEC/main.html](http://www.neo.lcc.uma.es/TutorialEA/semEC/main.html).

[www.Biological\\_Ground/principal.htm](http://www.Biological_Ground/principal.htm).

<http://ccp.servidores.net/genetico.html>

[www.cinefantastico.com/nexus7/ia/ageneticos.htm](http://www.cinefantastico.com/nexus7/ia/ageneticos.htm)

[http://delicias.dia.fi.upm.es/cursos/algoritmos\\_gen\\_ticos.html](http://delicias.dia.fi.upm.es/cursos/algoritmos_gen_ticos.html)

[http\\moebius.bio.ucm.es/pie/enlaces/algoritmos-geneticos.html](http://moebius.bio.ucm.es/pie/enlaces/algoritmos-geneticos.html)

[www.electa.uta.cl/~revista/algor-hd.pdf](http://www.electa.uta.cl/~revista/algor-hd.pdf)

[www.uarg.unpa.edu.ar/proy\\_invest/proyectos/nro29-a070.html](http://www.uarg.unpa.edu.ar/proy_invest/proyectos/nro29-a070.html)

[www.aircenter.net/gaia/ce/ce\\_c.htm](http://www.aircenter.net/gaia/ce/ce_c.htm)

[www.psycologia.com/articulos/ar-h\\_gascon01.htm](http://www.psycologia.com/articulos/ar-h_gascon01.htm)

[http\\bayes.escet.urjc.es/jmuruzabal/evocomp.html](http://bayes.escet.urjc.es/jmuruzabal/evocomp.html)

[http\\geneura.ugr.es/~jmerelo/DegaX/GenAlg.html](http://geneura.ugr.es/~jmerelo/DegaX/GenAlg.html)

[http\\cacic2001.info.unlp.edu.ar/cacic2000/Gallard.html](http://cacic2001.info.unlp.edu.ar/cacic2000/Gallard.html)

[http\\editorial.cic.ipn.mx/acm/coello.htm](http://editorial.cic.ipn.mx/acm/coello.htm)

[http\\slwww.epfl.ch/~acis/redcic/](http://slwww.epfl.ch/~acis/redcic/)

# **ANEXOS**

## ANEXO A

### Ejemplos

#### **Ejemplo 1: Funcionamiento de un Algoritmo Genético Simple.**

Se parte de una función  $f(x)$  muy sencilla:

- $f(x)=x^2$

Se desea encontrar el valor de  $x$  que hace que la función  $f(x)$  alcance su valor máximo, pero restringiendo a la variable  $x$  a tomar valores comprendidos entre 0 y 31. Aún más, a  $x$  sólo se le permite tomar valores enteros, es decir: 0,1,2,3,...,30, 31. Obviamente el máximo se tiene para  $x = 31$ , donde  $f$  vale 961. No se necesita saber algoritmos genéticos para resolver este problema, pero su sencillez hace que el algoritmo sea más fácil de comprender.

- Lo primero que se debe hacer es encontrar una manera de codificar las posibles soluciones (posible valores de  $x$ ). Una manera de hacerlo es con la codificación binaria. Con esta codificación un posible valor de  $x$  es 01011. ¿Cómo se interpreta esto? Muy sencillo: multiplica la última componente (un 1) por 1, la penúltima (un 1) por 2, la anterior (un 0) por 4, la segunda (un 1) por 8 y la primera (un 0) por 16 y a continuación se realiza la suma: 11. Observe que 00000 es equivalente a  $x = 0$  y que 11111 equivale a  $x = 31$ .

A cada posible valor de la variable  $x$  en representación binaria se le llamará individuo.

Una colección de individuos constituye lo que se denomina población y el número de

individuos que la componen es el tamaño de la población. Una vez se tenga codificada la solución, se debe escoger un tamaño de población. Para este ejemplo ilustrativo se escogerán 6 individuos.

Se debe partir de una población inicial. Una manera de generarla es aleatoriamente: se coge una moneda y se lanza al aire; si sale cara, la primera componente del primer individuo es un 0 y en caso contrario un 1. Repite el lanzamiento de la moneda y tendremos la segunda componente del primer individuo (un 0 si sale cara y un 1 si sale cruz). Así hasta 5 veces y se obtendrá el primer individuo. Se repite ahora la secuencia anterior para generar los individuos de la población restantes. En total se tiene que lanzar  $5 * 6 = 30$  veces la moneda.

- El siguiente paso es hacer competir a los individuos entre sí. Este proceso se conoce como selección.

La tabla A.1 resume el proceso.

(1)	(2)	(3)	(4)	(5)
1	01100	12	144	6
2	10010	18	324	3
3	01111	15	225	2
4	11000	24	576	5
5	11010	26	676	4
6	00001	1	1	1

**Tabla A.1 - Selección**

Cada fila en la tabla A.1 está asociada a un individuo de la población inicial. El significado de cada columna de la tabla es el siguiente:

(1) = Número que le asignamos al individuo.

(2) = Individuo en codificación binaria.

(3) = Valor de  $x$ .

(4) = Valor de  $f(x)$ .

Observe que el mejor individuo es el 5 con  $f = 676$ . Se calcula la media de  $f$  y se obtiene  $f_{med}=324.3$ . Una manera de realizar el proceso de selección es mediante un torneo entre dos. A cada individuo de la población se le asigna una pareja y entre ellos se establece un torneo: el mejor genera dos copias y el peor se desecha. La columna (5) indica la pareja asignada a cada individuo, lo cual se ha realizado aleatoriamente. Existen muchas variantes de este proceso de selección, aunque este método sirve para ilustrar el ejemplo.

- Después de realizar el proceso de selección, la población que se tiene es la mostrada en la columna (2) de la tabla A.2. Observe, por ejemplo, que en el torneo entre el individuo 1 y el 6 de la población inicial, el primero de ellos ha recibido dos copias, mientras que el segundo cae en el olvido.

(1)	(2)	(3)	(4)
1	01100	5	1
2	01100	3	3
3	10010	2	3
4	10010	6	1
5	11010	1	1
6	11010	4	1

Tabla A.2. - Cuce

Tras realizar la selección, se realiza el cruce. Una manera de hacerlo es mediante el cruce 1X: se forman parejas entre los individuos aleatoriamente de forma similar a la selección. Dados dos individuos (pareja) se establece un punto de cruce aleatorio, que no es más que un número aleatorio entre 1 y 4 (la longitud del individuo menos 1). Por ejemplo, en la pareja 2-3 el punto de cruce es 3, lo que significa que un hijo de la pareja conserva los tres primeros bits del padre y hereda los dos últimos de la madre, mientras que el otro hijo de la pareja conserva los tres primeros bits de la madre y hereda los dos últimos del padre. La población resultante se muestra en la columna (2) de la tabla A.3.

(1)	(2)	(3)	(4)
1	01010	10	100
2	11100	28	784
3	01110	14	196
4	10000	16	256
5	11010	26	676
6	10010	18	324

Tabla A.3.- Población tras el Cruce

En la columna (3) se tiene el valor de  $x$ ; en la siguiente se tiene el valor de  $f$  correspondiente. Ahora el valor máximo de  $f$  es 784 (para el individuo 2), mientras que antes de la selección y el cruce era de 676. Además  $f_{med}$  ha subido de 324.3 a 389.3. ¿Qué quiere decir esto? Simplemente que los individuos después de la selección y el cruce son mejores que antes de estas transformaciones.

El siguiente paso es volver a realizar la selección y el cruce tomando como población inicial la de la tabla A.3. Esta manera de proceder se repite tantas veces como número de iteraciones se fije. Y ¿cuál es el óptimo?. En realidad un algoritmo genético no te garantiza la obtención del óptimo pero, si está bien construido, proporcionará una solución razonablemente buena. Puede que se obtenga el óptimo, pero el algoritmo no te confirma que lo sea. También es buena idea ir guardando la mejor solución de todas las iteraciones anteriores y al final quedarse con la mejor solución de las exploradas.

- **Consideraciones adicionales**

En problemas reales en los que se aplican los algoritmos genéticos, existe la tendencia a la homogenización de la población, es decir a que todos los individuos de la misma sean idénticos. Esto impide que el algoritmo siga explorando nuevas soluciones, con lo que podemos quedar estancados en un mínimo local no muy bueno.

Existen técnicas para contrarrestar esta "deriva genética". El mecanismo más elemental, aunque no siempre suficientemente eficaz, es introducir una mutación tras la selección y el cruce. Una vez que se ha realizado la selección y el cruce se escoge un número determinado de bits de la población y se alteran aleatoriamente. En este ejemplo consiste simplemente en cambiar alguno(s) bit(s) de 1 a 0 ó de 0 a 1.

### **Ejemplo 2: Resolver un problema de optimización**

Un grupo de financieros colombianos ha resuelto invertir 10 millones de pesos en la nueva marca de aguardiente "Carrumbo". Así pues, en 4 ciudades de las principales de Colombia se decide iniciar una vigorosa campaña comercial: Bogotá en el centro, Cali en el sur, Medellín en el occidente y Villavicencio en el oriente. A esas 4 ciudades van a corresponder las zonas comerciales I, II, III y IV. Un estudio de mercado ha sido realizado en cada una de las zonas citadas y han sido establecidas curvas de ganancias medias, en millones de pesos, en función de las inversiones totales (almacenes, tiendas de venta, representantes, publicidad, etc.). Estos datos se ilustran en la tabla A.4. Para simplificar los cálculos, supondremos que las asignaciones de créditos o de inversiones deben hacerse por unidades de 1 millón de pesos. La pregunta es: ¿en dónde se deben de asignar los 10 millones de pesos de los que se dispone para que la ganancia total sea máxima?.

- **Representación:** Para poder aplicar el algoritmo genético, lo primero que necesitamos determinar es cuál será el esquema a utilizarse para representar las posibles soluciones del problema. En este caso necesitamos 4 bits ( $2^4 = 16$ ) para representar cada solución, porque cada una admite 11 valores posibles (de 0 a 10).

Inversión (en millones)	Beneficio I	Beneficio II	Beneficio III	Beneficio IV
0	0	0	0	0
1	0.28	0.25	0.15	0.20
2	0.45	0.41	0.25	0.33
3	0.65	0.55	0.40	0.42
4	0.78	0.65	0.50	0.48
5	0.90	0.75	0.62	0.53
6	1.02	0.80	0.73	0.56
7	1.13	0.85	0.82	0.58
8	1.23	0.88	0.90	0.60
9	1.32	0.90	0.96	0.60
10	1.38	0.90	1.00	0.60

Tabla A.4. Datos obtenidos con la investigación de mercado en cada una de las regiones en estudio

Como existen 4 valores independientes (uno por cada zona de estudio), se requieren entonces 16 bits (4 x 4) por cada cromosoma. Es importante hacer notar que se requiere una función de codificación (que transforme el valor de la inversión a binario) y una de decodificación (que realice el proceso inverso). Debido a que en este caso los 4 bits utilizados para representar una solución pueden producir más valores de los que se necesitan, se usará una función de ajuste que haga que los resultados producidos siempre se encuentren en el rango válido.

- **Función de Aptitud:** Dado que el objetivo es obtener las inversiones que sumen 10, y que tengan un beneficio máximo, podemos usar la siguiente función de aptitud penalizada:

$$F(x) = \frac{c1 + c2 + c3 + c4}{500 * V + 1}$$

Donde c1, c2, c3 y c4 son las ganancias por zona, que se calculan de acuerdo a los valores de la tabla 2, y v es el valor absoluto de la diferencia entre la suma obtenida de las inversiones y 10. Nótese que cuando no se viole ninguna restricción (cuando la suma de inversiones sea exactamente 10) la función de aptitud no será "castigada".

- **Operadores:** Se usará una cruce de 2 puntos. La probabilidad que se dará a la misma será del 80%. En cuanto a la mutación, se le asignará una probabilidad baja, que será del 1%. El tamaño de población manejado para este ejemplo será de 50 cromosomas, y se correrá el algoritmo genético durante 20 generaciones.
- **Resultados:** El resultado obtenido en una corrida típica es un beneficio de 1.81 millones de pesos, correspondiente a invertir 4 millones en la zona I, 3 millones en la zona II, 1 millón en la zona III y 2 millones en la zona IV. Esta es la solución óptima, la cual se obtuvo originalmente mediante programación dinámica.

Aunque éste es sólo un sencillo ejemplo del uso del algoritmo genético para resolver problemas de optimización con restricciones, puede alcanzarse a percibirse el poder de la técnica en comparación con los métodos tradicionales de búsqueda.

### Ejemplo 3: Implementación de un algoritmo genético

- **Problema a evaluar:** La función  $f(x,y,q,w) = x+y+q+w$  (la meta va a ser que la función sea igual a **6**).
- **Codificación:** Binaria de 3 posiciones (por ejemplo  $7 = 111$ ,  $3 = 011$ , etc).
- **Adaptación( $p_i$ ):** Evaluar valor absoluto  $(f(p_0, p_1, \dots, 0_i, \dots, p_3) - 6)$  donde  $0_i =$  un valor de cero en el parámetro  $i$ . Esta fórmula mide qué tanto se modifica la desviación de la meta cuando se quita el parámetro  $p_i$ . Cuanto mayor sea este número significa que es más importante por lo tanto está más "adaptado".
- **Reproducción( $p_i$ ,  $p_j$ ):** Seleccionar un número aleatorio " $m$ " de una distribución uniforme de 1 a 2 (el número de posiciones de la codificación binaria - 1), dividir a  $p_i$

desde la posición 1 hasta m y luego desde la siguiente posición hasta el final. Concatenar a la primer parte de  $p_i$ , la parte de  $p_j$  que va desde la posición  $m+1$  hasta la posición final. Y concatenar la parte de  $p_j$  que va desde 1 a m a la segunda parte de  $p_i$ . Por ejemplo ( $m=2$ ,  $p_i=101$ ,  $p_j=100 \rightarrow p_k=100$ ,  $p_l=101$ ).

- **Mutación:** Seleccionar un rango pequeño entre 0 y 1, por ejemplo (0,4 – 0,41) y para cada parámetro, generar un número aleatorio uniforme entre 0 y 1, si el número está dentro del rango, mutar el parámetro cambiando todas las posiciones a un valor de 0 o 1 generado aleatoriamente. También se muta siempre que todos los números son iguales o si la generación actual es igual a la anterior.
- **Meta:** si  $f(p_0, p_1, p_2, p_3)=6$  se llegó a la meta, si no, entonces no se llegó.

- **Comenzar algoritmo:**

Generación\_0: (3,4,5,7)

Selección	Reproducción	Mutación / Evaluación
Adaptación de 3: $(f(0,4,5,7)-6) = 10$	Elegidos: <b>3,4</b>	No-mutación.
Adaptación de 4: $(f(3,0,5,7)-6) = 9$	<input type="checkbox"/> Número aleatorio: 1 $3 = 011 \Rightarrow 000 = 0$	$f(3,4,0,7)=14$ diferente de 6.
Adaptación de 5: $(f(3,4,0,7)-6) = 8$	$4 = 100 \Rightarrow 111 = 7$	
Adaptación de 7: $f(3,4,5,0)-6 = 6$	<b>Generación_1:</b> <b>(3,4,0,7)</b>	
Adaptación de 3: $(f(0,4,0,7)-6) = 5$	Elegidos: <b>3,4</b>	No-mutación
Adaptación de 4: $(f(3,0,0,7)-6) = 4$	<input type="checkbox"/> Número aleatorio: 2 $3 = 011 \Rightarrow 010 = 2$	$f(3,4,2,3)=12$ diferente de 6
Adaptación de 0: $(f(3,4,0,7)-6) = 8$	$4 = 100 \Rightarrow 011 = 3$	
Adaptación de 7: $f(3,4,0,0)-6 = 1$	<b>Generación_2:</b> <b>(3,4,2,3)</b>	
Adaptación de 3: $(f(0,4,2,3)-6) = 3$	Elegidos: <b>3,2</b>	No-mutación.
Adaptación de 4: $(f(3,0,2,3)-6) = 2$	<input type="checkbox"/> Número aleatorio: 2 $3 = 011 \Rightarrow 010 = 3$	$f(3,3,2,3)=11$ diferente de 6
Adaptación de 2: $f(3,4,0,3)-6 = 4$	$2 = 010 \Rightarrow 011 = 3$	
Adaptación de 3:	<b>Generación_3:</b>	

(f(3,4,2,0)-6) = 3 Adaptación de 3: (f(0,3,2,3)-6) = 2 Adaptación de 3: (f(3,0,2,3)-6) = 2 Adaptación de 2: (f(3,3,0,3)-6) = 3 Adaptación de 3: (f(3,3,2,0)-6) = 2 Adaptación de 3: (f(0,2,2,3)-6) = 1 Adaptación de 2: f(3,0,2,3)-6) = 2 Adaptación de 2: (f(3,2,0,3)-6) = 2 Adaptación de 3: (f(3,2,2,0)-6) = 1 Adaptación de 2: (f(0,2,2,2)-6) = 1 Adaptación de 2: f(2,0,2,2)-6) = 1 Adaptación de 2: (f(2,2,0,2)-6) = 1 Adaptación de 2: (f(2,2,2,0)-6) = 1 Adaptación de 2: (f(0,3,2,2)-6) = 1 Adaptación de 3: (f(2,0,2,2)-6) = 0 Adaptación de 2: (f(2,3,0,2)-6) = 1 Adaptación de 2: (f(2,3,2,0)-6) = 1	<b>(3,3,2,3)</b> Elegidos: <b>3,2</b>  <input type="checkbox"/> Número aleatorio: 1 3 = <b>011</b> => <b>010</b> = 2 2 = <b>010</b> => <b>011</b> = 3  <b>Generación_4:</b> <b>(3,2,2,3)</b> Elegidos: <b>2,2</b>  <input type="checkbox"/> Número aleatorio: 2 2 = <b>010</b> => <b>010</b> = 2 2 = <b>010</b> => <b>010</b> = 2  <b>Generación_5:</b> <b>(2,2,2,2)</b> Elegidos: <b>2,2</b>  <input type="checkbox"/> Número aleatorio: 2 2 = <b>010</b> => <b>010</b> = 2 2 = <b>010</b> => <b>011</b> = 2  <b>Generación_6:</b> <b>(2,2,2,2)</b> Elegidos: <b>2,2</b>  <input type="checkbox"/> Número aleatorio: 1 2 = <b>010</b> => <b>010</b> = 2 2 = <b>010</b> => <b>010</b> = 2  <b>Generación7(2,2,2,2)</b>	No-mutación. f(3,2,2,3)=10 diferente de 6  No-mutación. f(2,2,2,2)=8 diferente de 6  Mutación parámetro 1 2 = 010 => 011 = 3 f(2,3,2,2)=9 diferente de 6  Mutación parámetro 3 2 = 010 => 000 = 0 <b>f(2,2,2,0)=6 igual a 6.</b> <b>FIN</b>
--	---	--

Tabla A.5. Secuencia del Algoritmo Genético.

Del ejemplo anterior podemos obtener varias conclusiones acerca del uso de algoritmos genéticos para la solución de problemas:

- Si ya se tiene una forma analítica o matemática para la solución del problema NO se deben usar algoritmos genéticos.
- Una de las partes más importantes del algoritmo es la función de "adaptación" ya que de acuerdo a esta evolucionan los parámetros indicados.

- Si la mutación en los individuos, se pueden "contaminar", es decir ya se cruzan individuos con los mismos genes (todos los parámetros son iguales) y por lo tanto no hay evolución, y por otro lado la necesidad de cambiar el rumbo de las generaciones si estas son las generaciones presentes son iguales a las pasadas.

## **ANEXO B**

### **Aplicaciones de los Algoritmos Genéticos.**

#### **Reconocimiento de Criminales.**

Uno de las más novedosas aplicaciones de los Algoritmos Genéticos a la fecha, ha sido creada por Caldwell y Johnson (Caldwell, 1991). Este sistema es usado para ayudar al testigo en la reconstrucción de la descripción facial del criminal. Este sistema está basado en el hecho, de que el testigo puede identificar fácilmente al sospechoso si lo vuelve a ver, pero le resulta más complicado hacer la descripción facial del sospechoso.

El sistema contiene una librería de características faciales básicas. Por ejemplo el sistema contiene imágenes de narices, ojos, orejas, etc. El sistema contiene los siguientes bloques para la construcción de los rostros:

- Frentes
- Ojos y distancia de separación
- Narices
- Bocas
- Mentones

El sistema utiliza cadenas de 35 bits para codificar dichas características y crea una población inicial de 20 rostros. El testigo estudia los 20 rostros, y les da a cada uno una

calificación (0 a 9), la cual sirve como el valor de aptitud de cada rostro, entonces se crea una nueva población, utilizando la selección, el cruce y la mutación.

### **Generación de la Trayectoria de un brazo robótico.**

Este Algoritmo fue creado por Yuval Davidor. Él muestra como al aplicar los Algoritmos Genéticos sirven para planear el camino de la puesta en movimiento de un brazo robótico de un punto a otro. Davidor utiliza cromosomas de longitud variable en su solución, además de novedosos e interesantes operadores de cruce.

### **Sistemas Dinámicos no lineales, y Seguridad Internacional.**

Creado por Stephanie Forrest y Gottfried Mayer-Kress. Concierno a un problema planteado por la investigación actual en modelos caóticos de procesos reales.

Modelos Caóticos de Grupos Internacionales y competición económica buscan modelar algunas características de procesos del mundo real mejor que las técnicas tradicionales usadas para ello. Los autores usan el Algoritmo Genético para encontrar el mejor conjunto de parámetros de los modelos de Mayer-Kress, que optimicen el rendimiento de dichos modelos.

### **Síntesis Genética de la Arquitectura de una Red Neuronal.**

Propuesto por Steven A. Harp y Tariq Samad. Describe la técnica para codificar la arquitectura de una red neuronal en cromosomas binarios. Los autores utilizan cromosomas de longitud variable y una variedad de nuevas técnicas.

**Método para Pronosticar Múltiples Fallas.**

Creador por Gunar E. Liepens y W. D. Potter. Discute el uso de Algoritmos Genéticos para encontrar la combinación más razonable de causas para un sistema de comunicación de microondas. Los autores usan cromosomas binarios para representar soluciones a un problema que ellos muestran es un tipo de cobertura del problema.

**Análisis Conformacional de DNA.**

Por C. B. Lucasius, M. J. J. Blommers, L. M. C. Buydens, y G. Kateman. Este es el desarrollo de un Algoritmo Genético para determinar la estructura de una muestra de ADN, basada en datos espectrométricos de esta. Una interesante técnica de evaluación en cascada que realza la eficiencia de la función de Evaluación es usada. Los autores utilizan cadenas de bits para codificar las estructuras moleculares. La función de evaluación mide el grado que cada estructura decodificada conforma los datos que han sido tomados de la muestra. El Algoritmo Genético desarrolla una descripción de la estructura molecular que está en concordancia con los datos recolectados.

## ANEXO C

Puntos Utilizados para realizar las pruebas de las mejoras propuestas.

Puntos_ 8,75_-3,25		Puntos_ 1_0		Puntos_A_1		Puntos_A_2	
X	Y	X	Y	X	Y	X	Y
0	-3,25	1	1	-1	2	1	3
1	5,5	5	5	1	4	2	1
2	14,25	6	6	2	6	3	5
3	23	-2	-2	-2	-3	4	2
4	31,75	-10	-10	5	7	5	6
5	40,5	7	7	4	5	-1	-2
-1	-12	-3	-3	-5	3	-2	-5
-2	-20,75	8	8	-8	8	-3	-3
-3	-29,5	13	13			-4	-6
-4	-38,25	9	9			-5	-8
-5	-47						

Puntos_A_3		Puntos_A_4		Puntos_XOR		Puntos_-1_0	
X	Y	X	Y	X	Y	X	Y
-1	3	0,9	0,8	-1	-1	-1	1
-2	5	-0,87	3,5	1	1	-2	2
-3	6	-6,7	2,45	-1	1	-5	5
-4	1	-4,67	-2,6	1	-1	-8	8
-5	7	<b>2Puntos</b>		<b>Puntos_XOR_1</b>		1	-1
1	-5	1,756	5,32	-1	-1	5	-5
2	-2	-4,78	-7,3	1	1	7	-7
3	-8			-1	1	10	-10
4	-12			1	-1	5,8	-5,8
5	-3			2	2	-4,7	4,7

Estas tablas se encuentran en el archivo Puntos\_Prueba.xls.

## **ANEXO D**

### **Artículo Ecuación de Estado.**

#### **SIMULACION DE EXPERIMENTOS PVT PARA YACIMIENTOS DE GAS CONDENSADO Y ACEITE VOLATIL**

**Calvete González Fernando E. y Rangel Cortés Nohora Patricia  
Universidad Industrial de Santander**

### **RESUMEN**

Este trabajo se planteó con el objetivo de Implementar y desarrollar un programa interactivo que simule experimentos PVT para yacimientos de gas condensado y petróleo volátil, con el cual se logre reproducir de manera práctica las condiciones que se presentan en el curso de la prueba y se muestre en forma didáctica el procedimiento para realizarla en laboratorio.

Además de ofrecer esta reproducción funcional del laboratorio se encontró importante predecir en todo momento el comportamiento de variables termodinámicas como presión, temperatura, volumen y fracciones gas - líquido, obteniendo así resultados previos a la fase experimental, que ayudan a fijar la atención en puntos claves donde la medición debería ser más precisa para lograr una buena predicción del comportamiento PVT de los fluidos de yacimiento.

La interactividad del programa permite a los usuarios familiarizarse con las técnicas experimentales utilizadas, observar y corregir los errores cometidos durante el transcurso de la prueba, por lo cual servirá como material de entrenamiento de personal.

## **INTRODUCCION**

Los datos obtenidos de las pruebas PVT son necesarios en el manejo integral del yacimiento, sin embargo, normalmente no se está muy familiarizado con los procedimientos experimentales utilizados para conseguirlos, por consiguiente, se consideró de gran importancia, proporcionar un mecanismo que facilite el aprendizaje de las técnicas de laboratorio utilizadas en estos experimentos.

En el caso de pruebas para aceite volátil y gas condensado, se requiere trabajar a altas presiones y temperaturas. La utilización de esta herramienta software, permitirá reproducir de una manera segura dichas condiciones, evaluando así de manera aproximada el comportamiento de fases de dichos fluidos.

La evaluación matemática del desarrollo de un experimento PVT se dificulta, en ocasiones debido a la inestabilidad de fases en el equilibrio y a la pérdida de validez de las ecuaciones de balance de energía en zonas de alta presión. Por ello se utilizaron correlaciones y procedimientos matemáticos que aseguren la predicción de las variables involucradas en el comportamiento de fases.

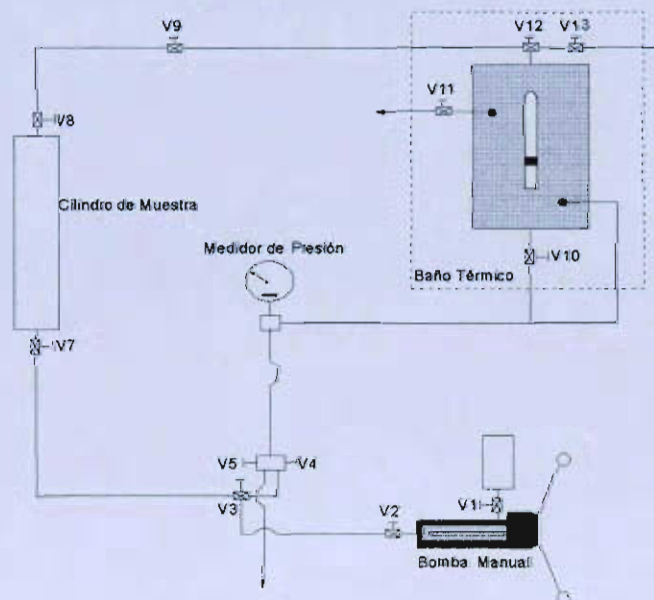
La herramienta fundamental para la predicción de las condiciones de la muestra durante la prueba es la ecuación de estado y los procedimientos matemáticos que giran en torno a ella. Uno de los problemas que hacen más sensibles los resultados, es la descripción adecuada de la fracción pesada, debido a la forma gráfica en que opera el simulador, se utilizó procedimientos standard de manejo de la fracción pesada y para suplir la imprecisión que pueda generar el hecho de no caracterizar dicha fracción de una manera más extendida se utilizaron mejores correlaciones para calcular sus propiedades físicas y una modificación de los valores de los parámetros que la definen en la ecuación de estado.

Uno de los objetivos de la simulación de las pruebas es predecir en cada instante la presión total del sistema, la cual está afectada principalmente por las propiedades del fluido que se esté analizando. En laboratorio, se controla el curso de la prueba fijando el volumen para cada paso y se lee la presión en el Manómetro, mientras que en las ecuaciones que describen matemáticamente el proceso ocurre lo contrario, partiendo de una presión conocida se obtienen resultados volumétricos, por ello se debió crear e implementar un método que a partir de ecuaciones refleje lo que ocurre realmente en el laboratorio.

El aparato que el software representa gráficamente es el equipo de comportamiento de fases (JEFRI PHASE BEHAVIOR) que se encuentra en los laboratorios tecnológicos de Producción del Instituto Colombiano del Petróleo ( I.C.P.).

La representación de este equipo se muestra en la figura 1.

**Fig. 1 Equipo PHASE BEHAVIOR**



Las celdas PVT están diseñadas especialmente para la medición de las propiedades volumétricas de los fluidos y para el estudio de su comportamiento de fases. Las mediciones de interés pueden incluir diversas propiedades de las fases vapor y líquido tales como densidad, relación vapor/líquido. Estas celdas son recipientes cuyo volumen interno es conocido para un amplio rango de presiones y temperaturas y que pueden mantenerse a temperatura constante. La celda esta equipada con un pistón flotante que permite eliminar el mercurio como medio desplazante reemplazándolo por aceite mineral. Además de un sistema de doble ventana visual que permiten ver las muestras y con fluido de sobrecarga que asegura que el volumen del cilindro sea constante durante todo el rango de presiones.

### **Baño de Aire**

El baño térmico es un horno calentado eléctricamente, que mantiene constante la temperatura a la que se realiza la prueba.

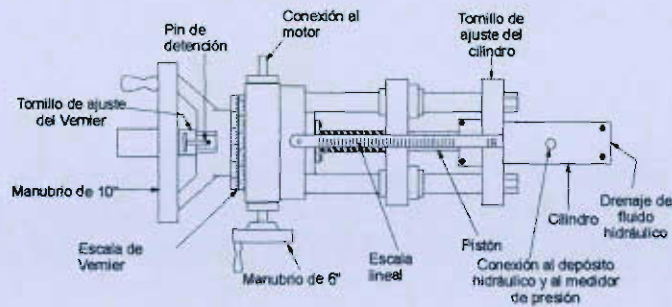
### **Bomba Manual de Desplazamiento Positivo**

Está diseñada para levantar la presión de los cilindros de muestra y de las celdas visuales transfiriendo aceite mineral a ellas por medio de las líneas.

### **Mecanismo de Agitación**

Imparte un movimiento de oscilación a las celdas, agitando los fluidos contenidos en ellas, para alcanzar el equilibrio.

**Fig.2 Bomba Manual de Desplazamiento Positivo**



### **Cilindros de Muestra**

Son cilindros de acero inoxidable, diseñados para transferir y almacenar las muestras de fluidos a alta presión. Tienen un pistón libre que separa el fluido de trabajo del fluido.

### **Catetómetro**

Es una reglilla graduada que mide alturas. Por medio de una constante de calibración la altura leída de los fluidos involucrados en la celda, puede traducirse a mediciones de volumen.

## **EXPERIMENTOS PRESION-VOLUMEN**

Los experimentos PVT fundamentales realizados en sistemas de gas condensado y aceite volátil son el Agotamiento a Volumen Constante (AVC) y la Expansión a Composición Constante (ECC).

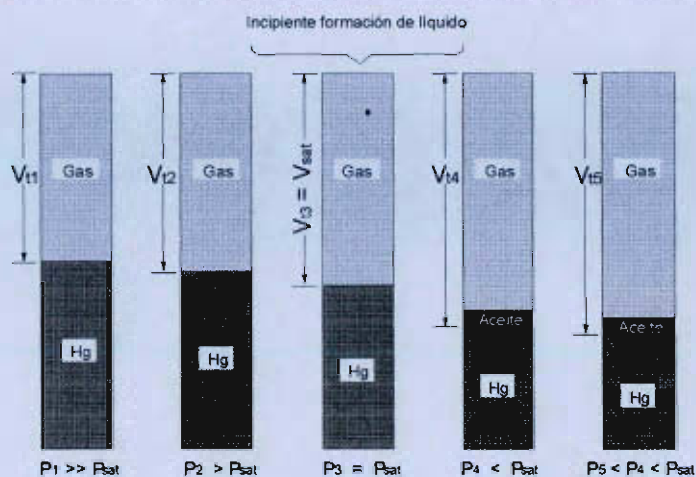
### **Expansión a Masa Constante (EMC)**

Esta prueba se conoce también como Expansión a Composición Constante o vaporización flash.

Se usa para medir el volumen total de fluido y su compresibilidad para un amplio rango de presiones desde la presión inicial de yacimiento hasta valores por debajo de la del separador, también es utilizada en la determinación de la presión de saturación a condiciones de yacimiento.

Esta prueba comienza con un fluido de yacimiento representativo (recombinando gas y aceite del muestreo). La Temperatura es la del yacimiento y la presión es igual o superior a la que soporta el fluido en la roca. El proceso de la prueba se puede ver en la figura 3.

**Fig. 3 Expansión a Masa Constante (EMC) para un Gas Condensado**



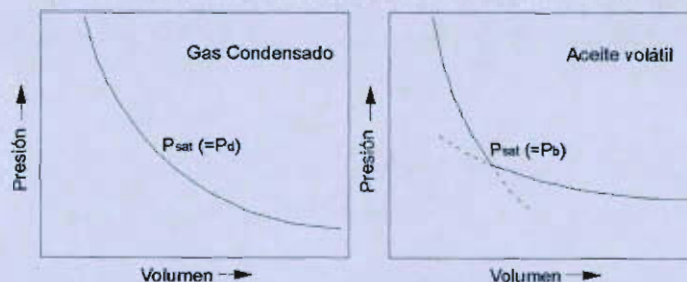
La presión se reduce paso a paso. Se observa en la celda visual el sistema y se registra el volumen en cada cambio de presión, según las lecturas de la bomba manual.

En el caso de fluidos volátiles cercanos al punto crítico, puede ser difícil determinar si la presión de saturación es un punto de rocío o de burbuja, porque las dos fases son cercanamente idénticas. Los cambios de fases son rápidos y exhiben grandes variaciones en su comportamiento para pequeños cambios de presión y temperatura. Una vez la presión de saturación del sistema se ha encontrado, y el estudio a masa constante se ha

realizado, el programa de trabajo continúa de la misma manera que para un gas condensado.

Este comportamiento se puede ver en la figura 4.

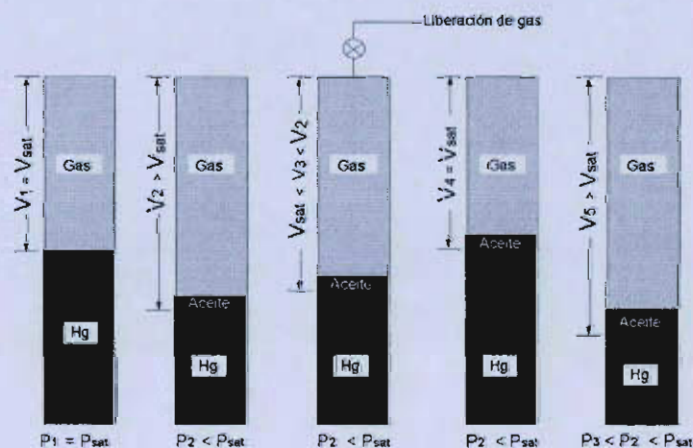
**Fig. 4 Comportamiento P-V para gas condensado y aceite volátil**



### Agotamiento a Volumen Constante (AVC)

Este experimento se realiza para simular el comportamiento del fluido en el yacimiento durante la producción. El procedimiento de la prueba se muestra en la figura 5.

**Fig. 5 Agotamiento a Volumen Constante (AVC) para Gas Condensado**



De nuevo el experimento comienza con una muestra de fluido en una fase, de volumen conocido, a presión de saturación y temperatura de yacimiento. De la misma manera que

el experimento anterior, la presión se reduce paso a paso, resultando una expansión del volumen en cada uno. Después de alcanzar el equilibrio, se retira fluido de la celda hasta alcanzar el volumen inicial. El gas removido en cada etapa, se lleva a condiciones estándar y se analiza para determinar la composición y el peso molecular del  $C_7^+$ . Después de la etapa final, el líquido se remueve de la celda para analizarlo.

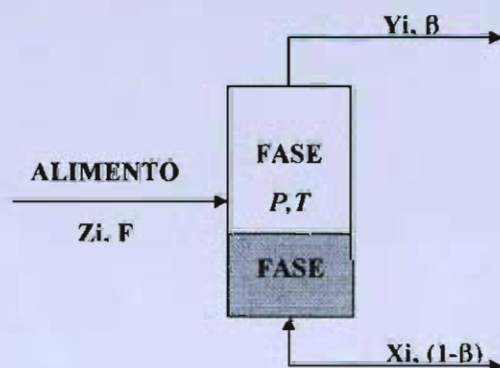
Los datos que son determinados en la prueba son los volúmenes de gas y líquido a presión y temperatura, la cantidad de gas producida acumulada, el factor Z del gas y la composición total de la corriente del pozo.

## CALCULOS FLASH

Los procesos FLASH son variaciones de las condiciones fisicoquímicas de una mezcla de fluido que se efectúan de una manera instantánea en los cuales se mantienen constantes algunas propiedades termodinámicas, para poder predecir el comportamiento del fluido como una función de las variables que afectan la mezcla.

El Flash que ocurre en los experimentos PVT es de tipo (P,T) y es una expansión isotérmica del fluido, como se presenta en la Figura 6 :

Fig. 6 Esquema de un Flash (P,T)



Los resultados de un Flash P-T son:

- Número de fases presentes en el equilibrio.
- Cantidades molares de cada fase.
- Composición molar de cada fase.
- Densidades de fases.
- Cantidades molares de Entropía, Entalpía y capacidad calorífica de cada fase.

En resumen, se requiere resolver el siguiente sistema de ecuaciones simultáneas no lineales:

- ◆ Balance de materia:

$$Z_i = \beta \times Y_i + (1 - \beta) \times X_i \quad \dots\dots\dots (1)$$

$$i=1,2,\dots,N_c$$

- ◆ Restricción de las fracciones:

$$\sum_{i=1}^{N_c} (Y_i - X_i) = \sum_{i=1}^{N_c} \frac{Z_i \times (K_i - 1)}{1 + \beta \times (K_i - 1)} = 0 \quad \dots\dots\dots (2)$$

- ◆ Condición de equilibrio:

$$f_i^V = f_i^L \quad \dots\dots\dots (3)$$

Donde:  $\Phi_i^V = \frac{f_i^V}{y_i p}$     y     $\Phi_i^L = \frac{f_i^L}{x_i p}$

Por lo tanto:

$$Y_i \times \Phi_i^V = X_i \times \Phi_i^L \quad \dots\dots\dots (4)$$

$$i=1,2,\dots,N_c$$

Resultando así un sistema de  $(2N_c + 1)$  ecuaciones con  $(2N_c + 1)$  incógnitas:  $X_i, Y_i, \beta$ .

## MODELAMIENTO DEL EQUILIBRIO DE FASES

El proceso de Flash isotérmico se puede tomar como un procedimiento de equilibrio de fases el cual a su vez, solo es una parte de un procedimiento termodinámico mucho mayor.

Una manera de resolver el sistema de ecuaciones que se a formado en el proceso de Flash, se basa en un esquema numérico de solución de sistemas de ecuaciones no-lineales, y este a su vez es planteado como un procedimiento iterativo basado en una de las herramientas termodinámicas más valiosas con la que contamos los ingenieros: *La Ecuación de Estado*.

El corazón del simulador de pruebas es la ecuación de estado de Peng-Robinson (EDE-PR), la cual produce cálculos volumétricos y de fugacidades sobre la base de parámetros físicos y críticos de los componentes de la mezcla. Además se utilizaron procedimientos correctivos de esta EDE para mejorar su predicción volumétrica y la entrada de los parámetros de la fracción pesada.

El cálculo del equilibrio líquido - vapor usando ecuaciones de estado genera un sistema de ecuaciones no lineales que debe resolverse iterativamente. Comúnmente el procedimiento usado es el de sustituciones sucesivas, pero exhibe una rata de convergencia pobre cerca al punto crítico. Para solucionar este problema, se utilizó el método de L. X. Nghiem<sup>9</sup>, quien creó un algoritmo robusto para cálculos flash que combina una versión especial del método de sustituciones sucesivas con el procedimiento de Powell<sup>11</sup> si se observa una

convergencia lenta. Con esta metodología se busca asegurar la convergencia del cálculo flash. El esquema de sustituciones sucesivas se modificó de manera que la región de una sola fase pueda ser detectada sin tener que calcular la presión de saturación.

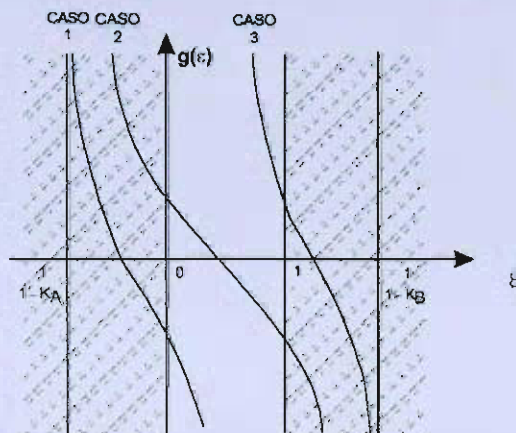
En el enfoque tradicional, al resolver las ecuaciones resultantes del flash no necesariamente se asegura que las fases se encuentren en equilibrio y por lo tanto se puede presentar una inestabilidad del sistema. Por esta razón pueden existir soluciones de dicho sistema de ecuaciones donde  $\beta$  no se encuentre en el rango normal de cero a uno. Para asegurar que esto no suceda se utilizó el criterio del flash negativo de L. X. Nghiem<sup>9</sup>.

Este criterio se basa en la minimización de una función monótonica y decreciente  $g(\varepsilon)=g(\beta)$ :

$$g(\beta) = \sum_{i=1}^{N_c} (Y_i - X_i) = \sum_{i=1}^{N_c} \frac{Z_i \times (K_i - 1)}{1 + \beta \times (K_i - 1)} = 0 \quad \dots\dots\dots (5)$$

Esta minimización da como resultado tres casos, como se muestra en la figura 7:

**Fig. 7 Condiciones para la existencia de la solución de  $g(\varepsilon)$  en  $[0,1]$ .**



Caso1:  $g(0) < 0$  y  $g(\epsilon_0) < 0$ : En este caso la mezcla se encuentra en estado monofásico (líquido).

$\beta$  se asume igual a 0.5 en la primera iteración y a 0 en la subsecuentes iteraciones.

Caso2:  $g(0) \geq 0$  y  $g(1) \leq 0$ : entonces  $g(\epsilon_0) \in [0,1]$ .

Existe más de una fase en equilibrio.  $\beta$  se obtiene utilizando un método iterativo como el de Rachford y Rice<sup>14</sup>

Caso3:  $g(1) > 0$  y  $g(\epsilon_0) > 1$ : La mezcla se encuentra en estado monofásico (gaseoso).

Estos criterios nos aseguran que el algoritmo se encuentre iterando en la región correcta y así poder asegurar la *estabilidad* de las fases.

En el desarrollo de las pruebas se requiere conocer en cada instante la presión del sistema, a partir del volumen, pero las ecuaciones manejan esta relación de forma inversa, por lo cual fue necesario diseñar e implementar un método que nos permita invertir esta relación. Para lograrlo y mantener la velocidad de calculo del simulador se utilizó el criterio desarrollado por Rachford y Rice<sup>14</sup> pero iterando en función de la presión, como se puede ver en la figura 8.

Fig. 8 Comportamiento de la función de presión en el esquema de R&R.

