

03063

7



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**POSGRADO EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN**

**COMPUTACIÓN EVOLUTIVA: SU APLICACIÓN  
COMO AUXILIAR EN EL DIAGNÓSTICO DE LA  
NEUROCYSTICERCOSIS HUMANA**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A :**

**MARCOS IVÁN QUINTANA HERNÁNDEZ**

**DIRECTOR DE TESIS: DR. JULIO SOLANO GONZÁLEZ**

México, D.F.

2000

03063



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**POSGRADO EN CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN**

**COMPUTACIÓN EVOLUTIVA: SU APLICACIÓN  
COMO AUXILIAR EN EL DIAGNÓSTICO DE LA  
NEUROCISTICERCOSIS HUMANA**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A :**

**MARCOS IVÁN QUINTANA HERNÁNDEZ**

**DIRECTOR DE TESIS: DR. JULIO SOLANO GONZÁLEZ**

**CO-DIRECTORA DE TESIS: DRA. KATYA RODRÍGUEZ VÁZQUEZ**

**A Lucía Susana**

*Noyolnantzin, yolchipahtzin,  
ma xihceli inin nocuic;  
noyoltech xochicueponi  
huan noyolouh itech yauh.*

*Xopan, tonalpan, cecuizpa  
tlemach tequitl otican  
ca zan nopampa otitoneul  
zan noyoca otimocauh.*

*Ma cemihcac tlazohnantzin  
ni mi temacaz mahuizotl,  
pampa aih tinech nencahua  
huan moyezo notech ca.*

*Fidencio Villanueva*

Madre de mi corazón, remedio del corazón,  
Recibe esta canción mía;  
Brotó la flor en mi corazón  
Y en mi corazón la acompaña.

En tiempos de verdor, de calor y frío  
Cualquier trabajo dejaste pues  
sólo por mi padeciste aflicción  
y por mi abandonaste.

Que siempre querida madre  
te de honra porque tu nunca  
me dejas desdichado  
y tu sangre está conmigo.

Fidencio Villanueva

**A Said Uriel**

*Nem i liliztli*

*Hichok nelli nemouna  
in haktikpak zan  
axika ye ni kan.  
Telkaxallhuti  
no chamani  
no teokuikatl  
in tlalpani  
anochipa tlaltikpak  
zanachika ye ni kan*

*Netzahualcóyotl*

**Pensamiento**

¿Es que en verdad se  
vive aquí en la tierra?,  
¡no para siempre aquí!  
Un momento en la tierra,  
si es de jade se hace astillas,  
si es de oro se destruye,  
si es plumaje de ketzalli se rasga,  
¡no para siempre aquí!  
Un momento en la tierra.

Netzahualcóyotl

## AGRADECIMIENTOS

Mis más profundos agradecimientos para:

El Dr. Julio Solano por haber aceptado dirigir esta tesis y por haberme encausado para adquirir los elementos académicos para desarrollarla.

La Dra. Katya Rodríguez por analizar minuciosamente cada una de las propuestas y por sus constantes recomendaciones para el desarrollo de un mejor trabajo.

El Dr. Fabián García Nocetti por haber hecho las gestiones necesarias para facilitarme espacio físico, equipo de cómputo y apoyo económico para la realización de este proyecto.

El Dr. Ernesto Bribiesca, por haberme dado la idea inicial del proyecto. También quiero expresarle mi reconocimiento, con respeto y admiración, por su ejemplo como investigador y como ser humano.

El M.C. Hermilo Sánchez por haberme facilitado los resultados de su trabajo relacionado a este proyecto y por explicarme pacientemente cada detalle técnico involucrado.

El Dr. Carlos Larralde y la Dra. Tzipe Govezensky, del Instituto de Investigaciones Biomédicas, por aclarar mis dudas y por facilitar el material para el desarrollo del proyecto.

La Dra. María Garza por gestionar apoyo para la divulgación de este trabajo.

La Dra. Susana Gómez por facilitarme un excelente equipo de cómputo en la recta final del desarrollo de esta tesis, que me permitió realizar experimentos mucho más robustos.

El personal del Departamento de Ingeniería en Sistemas Computacionales y Automatización del IIMAS por brindarme su amistad y su apoyo.

El Conacyt por haberme apoyado económicamente con una beca crédito para la realización de mis estudios y por apoyo material (proyectos 2146P-A9507, 27982-A y REDII 7350-858).

La UNAM que a través de la DGEP y la DGAPA me apoyó con becas complementarias.

La Señorita Enriqueta Barilla por su apoyo constante y por los diseños de estas páginas.

El profesor José Angel Ramírez Vallejo por haber cultivado en mi la disciplina para luchar por alcanzar todas las metas en la vida.

Por último, pero no menos importante, quiero agradecer al maestro Francisco José Espinosa Soberanes por haber despertado, quizá sin saberlo, mi vocación hacia la investigación.

## PUBLICACIONES

El trabajo presentado en esta tesis ha generado parte de la información publicada en:

a) Artículo en memoria de congreso internacional.

Quintana Hernández, M.I. y Solano González, J. 2000. "A Genetic Algorithm Approach to Auxiliary Diagnosis of Neurocysticercosis". In *Proceedings of the 2000 Genetic and Evolutionary Computation Congress. Workshop Program*. (Ed. Annie S. Wu, University of Central Florida). Las Vegas, Nevada, USA, julio 8. Pp. 301-304.

b) Capítulo en libro

Gómez, Susana; Solano, J.; Castellanos, L. y Quintana, M. I. 2000. "Tunneling and Genetic Algorithms for Global Optimization". **Convex Analysis and Optimization**. Kluwer Academic. (Capítulo Sometido)

# CONTENIDO

<b>DEDICATORIA .....</b>	<b>II</b>
<b>AGRADECIMIENTOS.....</b>	<b>III</b>
<b>PUBLICACIONES.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>CONTENIDO .....</b>	<b>V</b>
<b>I. INTRODUCCIÓN .....</b>	<b>1</b>
I.1. MOTIVACIÓN.....	1
I.2. OBJETIVO.....	3
I.3. CONTENIDO DE LA TESIS .....	3
I.4. APORTACIONES .....	4
<b>II. COMPUTACIÓN EVOLUTIVA.....</b>	<b>6</b>
II.1. INTRODUCCIÓN .....	6
II.2. TERMINOLOGÍA.....	7
II.3. ALGORITMOS GENÉTICOS .....	11
<i>II.3.1. Definición.....</i>	<i>11</i>
<i>II.3.2. Algoritmo Genético Simple.....</i>	<i>12</i>
II.3.2.1. Componentes Básicos.....	12
II.3.2.2. Operación Básica.....	12
<i>II.3.3. Representación.....</i>	<i>13</i>
<i>II.3.4. Operadores.....</i>	<i>14</i>
II.3.4.1. Selección .....	14
II.3.4.2. Cruzamiento .....	16
II.3.4.3. Mutación.....	18
<i>II.3.5. Algoritmos Genéticos Paralelos.....</i>	<i>19</i>
<i>II.3.6. Aplicaciones .....</i>	<i>20</i>
II.4. PROGRAMACIÓN GENÉTICA .....	22
<i>II.4.1. Definición.....</i>	<i>22</i>
<i>II.4.2. Tipos de Representación.....</i>	<i>25</i>
<i>II.4.3. Funciones de Aptitud.....</i>	<i>27</i>
II.4.3.1. Aptitud Estandarizada.....	28
II.4.3.2. Aptitud Ajustada.....	28
II.4.3.3. Aptitud Normalizada .....	29

II.4.4. Operaciones para Modificar Estructuras.....	29
II.4.4.1. Selección .....	29
II.4.4.2. Cruzamiento .....	30
II.4.4.3. Mutación.....	32
II.4.4.4. Permutación.....	33
II.4.4.5. Edición .....	34
II.4.4.6. Encapsulación.....	35
II.4.5. Aplicaciones .....	36
II.5. RESUMEN .....	38
II.6. REFERENCIAS .....	38
<b>III. NEUROCISTICERCOSIS.....</b>	<b>43</b>
III.1. INTRODUCCIÓN .....	43
III.2. PROBLEMA DE SALUD PÚBLICA .....	44
III.3. MÉTODOS DE DIAGNÓSTICO .....	45
III.3.1. Diagnóstico Clínico .....	46
III.3.2. Diagnóstico Radiológico.....	46
III.3.2.1. Radiografía Simple de Cráneo.....	46
III.3.2.2. Pneumoencefalografía, Ventriculografía y Angiografía.....	47
III.3.2.3. Tomografía Computarizada.....	47
III.3.2.4. Resonancia Magnética.....	48
III.3.3. Exámenes de Laboratorio .....	48
III.3.4. Inmunodiagnóstico.....	48
III.4. IMÁGENES WESTERN BLOT.....	50
III.4.1. Descripción del método.....	50
III.4.2. Interpretación de la prueba .....	52
III.4.3. Digitalización de las imágenes .....	53
III.4.4. Tratamiento previo y análisis.....	54
III.5. DEFINICIÓN DEL PROBLEMA .....	55
III.5.1. ¿Porqué es difícil de diagnosticar la neurocisticercosis?.....	55
III.5.2. Western blot como auxiliar en el diagnóstico.....	56
III.5.3. Propuestas de solución .....	58
III.5.4. Limitaciones.....	59
III.6. RESUMEN .....	59
III.7. REFERENCIAS.....	60
<b>IV. ALGORITMOS GENÉTICOS APLICADOS A IMÁGENES WESTERN BLOT: EL CASO DE LA NEUROCISTICERCOSIS.....</b>	<b>63</b>
IV.1. INTRODUCCIÓN.....	63



IV.2. DESCRIPCIÓN .....	64
IV.3. APROXIMACIÓN GENÉTICA (FORMULACIÓN) .....	67
IV.4. PRUEBAS Y RESULTADOS .....	69
IV.5. CONCLUSIONES .....	76
IV.6. REFERENCIAS.....	77
<b>V. ALGORITMOS GENÉTICOS APLICADOS A IMÁGENES WESTERN BLOT: DECISIÓN ENTRE NEUROCYSTICERCOSIS Y OTRAS ENFERMEDADES.....</b>	<b>78</b>
V.1. INTRODUCCIÓN .....	78
V.2. DESCRIPCIÓN .....	79
V.3. APROXIMACIÓN GENÉTICA (FORMULACIÓN) .....	79
V.4. PRUEBAS Y RESULTADOS .....	81
V.5. CONCLUSIONES .....	87
<b>VI. PROGRAMACIÓN GENÉTICA APLICADA A IMÁGENES WESTERN BLOT .....</b>	<b>88</b>
VI.1. INTRODUCCIÓN.....	88
VI.2. DESCRIPCIÓN .....	89
VI.3. PROGRAMACIÓN GENÉTICA (FORMULACIÓN).....	90
VI.4. PRUEBAS Y RESULTADOS.....	93
VI.5. CONCLUSIONES.....	97
VI.6. REFERENCIAS.....	97
<b>VII. CONCLUSIONES .....</b>	<b>99</b>
VII.1. CONCLUSIONES GENERALES .....	99
VII.2. TRABAJO FUTURO .....	101
<b>APÉNDICES.....</b>	<b>A-1</b>
APÉNDICE A: PROGRAMAS Y ARCHIVOS PARA ALGORITMOS GENÉTICOS .....	A-1
<i>Programa 1</i> .....	A-1
<i>Programa 2</i> .....	A-3
<i>Ejemplo de Archivo de Histograma</i> .....	A-4
<i>Programa 3</i> .....	A-6
<i>Funciones Objetivo</i> .....	A-10
Función 01 .....	A-10
Función 02 .....	A-11
<i>Parámetros</i> .....	A-13
APÉNDICE B: PROGRAMAS Y ARCHIVOS PARA PROGRAMACIÓN GENÉTICA .....	B-1
<i>Programa 1</i> .....	B-1
<i>Función Objetivo</i> .....	B-4

<i>Funciones para Creación de población</i> .....	<i>B-6</i>
<i>Funciones para Modificar Árboles</i> .....	<i>B-9</i>
Cruzamiento.....	<i>B-9</i>
Mutación.....	<i>B-11</i>
<i>Parámetros</i> .....	<i>B-13</i>

# I. INTRODUCCIÓN

## I.1. MOTIVACIÓN

La neurocisticercosis es la enfermedad parasitaria que se encuentra más frecuentemente en el sistema nervioso central del ser humano. Esta enfermedad representa un serio problema de salud pública en varios países del mundo, particularmente en los que se encuentran en vías de desarrollo, como el nuestro.

Este problema es atacado por las instituciones al servicio de la salud con diversas acciones, como son: proporcionar información a la población acerca de las medidas preventivas que deben tomar para evitar contraer el parásito, tratar médicamente a las personas que ya se encuentran infectadas, realizar estudios epidemiológicos para detectar zonas o sectores de la población particularmente sensibles e investigar nuevas alternativas para mejorar los tratamientos y métodos de diagnóstico existentes.

En la actualidad, el diagnóstico se basa en la experiencia del especialista al observar los síntomas presentados por el paciente y el apoyo de algunos métodos auxiliares. Sin embargo, aún no existe un método auxiliar que sea totalmente confiable por sí mismo.

Algunos de estos métodos como los estudios de tomografía computarizada y de resonancia magnética han proporcionado buenos resultados, pero son considerablemente caros. Por su parte, el estudio de líquido cefalorraquídeo ha demostrado en varios estudios su efectividad, pero tiene el inconveniente de ser peligroso y doloroso para el paciente, además de ser relativamente difícil de realizar.

Otra alternativa la constituyen los métodos inmunológicos, los cuales presentan mejores perspectivas para estudios a gran escala, debido a que se realizan con muestras de suero de los pacientes y son más económicos. Aunque los resultados obtenidos con los métodos inmunológicos no han sido lo suficientemente buenos, se sigue investigando como

utilizarlos para apoyar el diagnóstico, o bien como alternativa cuando no se cuenta con la posibilidad de realizar estudios más complejos.

Un método inmunológico que ha despertado particular interés en la comunidad científica es el método conocido como Western Blot (WB) debido a que es la prueba confirmatoria de diagnóstico del Síndrome de Inmunodeficiencia Adquirida. Por ese motivo, se han realizado diversas investigaciones tratando de utilizar el método WB como auxiliar en el diagnóstico de varias enfermedades, incluyendo la neurocisticercosis.

En la búsqueda de un análisis más profundo de la prueba WB se han utilizado densitómetros para obtener una versión digital de la prueba, a la cual se le conoce como imagen WB. Los análisis están enfocados a determinar características estadísticas, a la detección de bandas en las imágenes y a la búsqueda de un patrón auxiliar para diagnóstico.

En este trabajo se propone la utilización de técnicas de Computación Evolutiva para el análisis de las imágenes WB. Lo anterior debido a que algunas técnicas de Computación Evolutiva han demostrado su efectividad en la solución de problemas cuya complejidad difícilmente puede ser atacada con otros métodos.

De manera general, la Computación Evolutiva se compone de técnicas para búsqueda y optimización basadas en los mecanismos de la evolución y de la genética natural. A partir de una población inicial aleatoria de posibles soluciones a un problema dado, se busca la reproducción de los individuos más aptos durante varias generaciones. De tal forma que al evolucionar dichos individuos se encuentre uno cuya aptitud dé solución satisfactoria al problema considerado.

Particularmente las técnicas conocidas como Algoritmos Genéticos y Programación Genética han proporcionado buenas soluciones a problemas con espacios de búsqueda muy grandes y características de solución poco conocidas. Este es el caso del problema del análisis de las imágenes Western Blot que se aborda en esta tesis.

## I.2. OBJETIVO

El objetivo de esta tesis es proporcionar un método basado en imágenes WB y técnicas de Computación Evolutiva para auxiliar a los médicos especialistas en el diagnóstico de la neurocisticercosis humana. Los resultados que se obtengan en este trabajo:

- Darán más elementos al especialista para determinar el diagnóstico de un paciente en particular.
- Podrán ser utilizados como un paso previo a las pruebas radiológicas o a la extracción de LCR. O bien como sustituto en los lugares donde no se cuente con ellas.
- Proporcionarán elementos para purificar la prueba WB para neurocisticercosis.
- Abrirá nuevas líneas de investigación en la lucha contra esta enfermedad.
- Sentarán un precedente de los beneficios que representan las técnicas computacionales para auxiliar en la solución de este tipo de problemas médicos.

## I.3. CONTENIDO DE LA TESIS

El trabajo descrito en esta tesis se encuentra dividido en los siguientes capítulos:

**Capítulo I.** Presenta un panorama general del problema que se aborda, la motivación para resolverlo y las técnicas que se utilizaron como propuesta de solución. El objetivo general y el contenido de la tesis, así como sus principales contribuciones.

**Capítulo II.** Este capítulo muestra la teoría básica y los conceptos de los temas de la Computación Evolutiva que se abordan en la tesis, explica la terminología y proporciona una introducción general a los Algoritmos Genéticos y a la Programación Genética enumerando algunas de sus aplicaciones principales.

**Capítulo III.** En este capítulo se aborda desde una perspectiva general el tema de la neurocisticercosis como problema de salud pública, se mencionan los principales métodos de diagnóstico que existen y se aborda con más detalle el método Western Blot y su versión

digital. Además se expone el porqué de la complejidad para el diagnóstico de la neurocisticercosis y se hace un esbozo general de las propuestas de solución.

**Capítulo IV.** Se utilizan Algoritmos Genéticos en la búsqueda de imágenes Western Blot que sirvan como modelo para apoyar el diagnóstico de la neurocisticercosis. En este capítulo se muestran los resultados de experimentos preliminares y de un análisis de cómo afectan los intervalos de muestreo en la calidad de los resultados obtenidos y en el tiempo de procesamiento.

**Capítulo V.** Este capítulo presenta los resultados de varios experimentos con Algoritmos Genéticos para determinar un método auxiliar de diagnóstico basado en imágenes Western Blot compuestas de dos partes, una parte con características de las imágenes de pacientes con neurocisticercosis y otra parte con características de las imágenes de los pacientes con otras enfermedades.

**Capítulo VI.** En este capítulo se muestran los resultados preliminares de la búsqueda de un patrón auxiliar para el diagnóstico de la neurocisticercosis utilizando imágenes Western Blot y Programación Genética.

**Capítulo VII.** Se presentan las conclusiones finales de este trabajo y se muestran algunas alternativas futuras de investigación.

## I.4. APORTACIONES

Las alternativas de solución al problema del diagnóstico de la neurocisticercosis humana propuestas en esta tesis son innovadoras, abren la puerta a nuevas líneas de investigación y demuestran que es factible utilizar técnicas de la Inteligencia Artificial como medio de solución a problemas de investigación biomédica.

El trabajo propone un método auxiliar eficiente para el diagnóstico automático de la enfermedad a un bajo costo. Mas aún, la aplicación de las aproximaciones al método

propuesto se puede utilizar para el análisis de imágenes Western Blot en lugares que por sus características endémicas sean diferentes a las consideradas e incluso pueden reproducirse para modelar otras enfermedades con pruebas controladas.

Cabe señalar que el método propuesto de ninguna manera substituye al especialista, sino que le proporciona una herramienta de apoyo fácil de usar que le auxiliará en el diagnóstico acertado de la enfermedad.

Las imágenes utilizadas en las aproximaciones de solución propuestas corresponden a pacientes diagnosticados previamente con base en la experiencia de especialistas del Instituto Nacional de Neurología y han sido proporcionadas por el Instituto de Investigaciones Biomédicas (IIB) de la UNAM.

Es importante señalar que existen factores que limitan los resultados obtenidos. La calidad de los reactivos, la experiencia del laboratorista y el mismo proceso de digitalización de las imágenes, pueden propiciar la alteración de la información real. No obstante, los resultados demuestran la efectividad del método.

## II. COMPUTACIÓN EVOLUTIVA

### II.1. INTRODUCCIÓN

El término *Computación Evolutiva* se refiere al estudio de algunas técnicas heurísticas basadas en los principios de la evolución natural. Este estudio se puede dividir en tres grandes categorías: *Algoritmos Genéticos*, *Estrategias Evolutivas* y *Programación Evolutiva*. La división se basa en algunos detalles de desarrollo histórico de cada una de las técnicas y en algunas diferencias de su funcionamiento (Coello, 1998).

El origen de la Computación Evolutiva se produjo tratando de imitar el comportamiento de procesos que suceden en la naturaleza, tales como la evolución que se observa en los cromosomas de los organismos, los procesos evolutivos que se dan en la reproducción y el proceso de selección natural. Basados en estos procesos se desarrollaron independiente y casi simultáneamente los modelos de Computación Evolutiva mencionados anteriormente (Holland, 1962; Rechemberg, 1973; Fogel, 1965).

La Computación Evolutiva es considerada por muchos grupos como una parte de la disciplina conocida como Inteligencia Artificial. Entre las diferentes técnicas de aprendizaje de máquina existentes en la Inteligencia Artificial, la Computación Evolutiva presenta dos características interesantes: es capaz de generar por sí misma el conocimiento que requiere y selecciona aleatoriamente nuevas alternativas de solución para seguir generando nuevo conocimiento (Coello, 1998).

Un algoritmo evolutivo (propio de la Computación Evolutiva) es un proceso estocástico y repetitivo que opera sobre un conjunto de individuos llamado *población*. Cada individuo es una solución potencial para el problema particular a resolver. Inicialmente se genera una población aleatoria, se evalúa cada individuo en términos de qué tan bueno es para resolver el problema en cuestión y se entra a un proceso de selección, reproducción y reemplazo.



Durante la fase de selección deben obtenerse un mayor número de aquellos individuos que proveen una mejor solución que aquéllos con una peor. En la fase de selección se obtiene una nueva población diferente a la anterior tratando de conservar características deseables de los mejores individuos. Por último, en la fase de sustitución se cambian los individuos anteriores por los nuevos obtenidos de la selección, usualmente tratando de mantener a los mejores individuos y eliminando a los peores.

El proceso completo se repite hasta satisfacer un criterio de terminación o de convergencia. Es importante notar que este algoritmo equilibra la explotación de soluciones buenas con la exploración de nuevas zonas en el espacio de búsqueda. El comportamiento de estos algoritmos es estocástico, de tal forma que puede presentar diferentes soluciones en diferentes corridas de un mismo algoritmo.

Los Algoritmos Genéticos son los principales representantes de la Computación Evolutiva debido a que han tenido un acelerado desarrollo en los últimos años (Bäck, 1996; Buckles y Petry, 1992; y Mitchell, 1996). La Programación Genética puede ser considerada una técnica sucesora de los Algoritmos Genéticos y actualmente goza de mucha popularidad. En este capítulo se proporciona una introducción a los Algoritmos Genéticos (Goldberg, 1989) y a la Programación Genética (Koza, 1992).

## II.2. TERMINOLOGÍA

La Computación Evolutiva ha tomado muchos términos del campo de la Biología debido a su relación con las teorías de la evolución y de la selección natural. Sin embargo, estos términos tienen un significado diferente generalmente relacionado con fenómenos de la naturaleza. También se han tenido que incorporar algunos otros términos que se utilizan en otros ámbitos. A continuación se explica el significado de algunos términos frecuentemente usados con el objeto de que el lector tenga una mejor comprensión de dichos término en el contexto de la Computación Evolutiva.

- **Alelo:** Se refiere al valor de un gen. Por ejemplo, si se usa representación binaria, este valor puede ser 0 ó 1.
- **Aptitud:** Un valor asignado a un individuo que refleja qué tan bueno es éste para resolver una cierta tarea. Una *función de aptitud* se usa para mapear un cromosoma a un valor de aptitud. Un *paisaje de aptitud (fitness landscape)* es la hiper-superficie que se obtiene al aplicar la función de aptitud a cada punto del espacio de búsqueda.
- **Cromosoma:** Una estructura de datos que contiene una *cadena* de parámetros de diseño o genes. Esta cadena puede almacenarse, por ejemplo, como una cadena de bits u otra forma de representación (ej. enteros, punto flotante, etc.)
- **Cruzamiento:** Un operador de reproducción que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres. Más adelante se analizan algunas de las propuestas principales de cruzamiento que existen para codificación real y para estructuras de programas.
- **Elitismo (o estrategia elitista):** Es un mecanismo que se usa en algunos algoritmos evolutivos para asegurar que los cromosomas de los miembros más aptos de la población se pasen a la siguiente generación sin ser alterados por ningún operador genético. Usar elitismo asegura que la aptitud máxima de la población nunca se reducirá de una generación a la siguiente. El elitismo también suele acarrear una convergencia más rápida de la población, aunque en general puede decirse que no siempre mejora la posibilidad de localizar a un individuo óptimo.
- **Especiación:** Técnicas análogas al aislamiento geográfico que se utilizan en diversos algoritmos genéticos. Típicamente, la población se divide en sub-poblaciones y a los individuos sólo se les permite combinarse con otros que pertenezcan a la misma sub-población. Normalmente una pequeña población migrante es permitida.

- **Esquema:** Un patrón de valores de genes en un cromosoma que puede incluir estados *no importa* (*don't care*). Por tanto, en un cromosoma binario, cada esquema puede especificarse mediante una cadena de la misma longitud del cromosoma, en la que cada caracter es un elemento del conjunto {0, 1, #}. Un cromosoma en particular se dice que contiene un esquema determinado si corresponde al esquema en cuestión. Por ejemplo, el cromosoma 01101 corresponde al esquema #1#0#. El *orden* de un esquema es el número de posiciones que no sean 'no importa', mientras que la *longitud* es la distancia entre las 2 posiciones más alejadas de valores diferentes a *no importa*. Por lo tanto, #1##0# tiene un orden 2 y una longitud de definición de 3.

- **Exploración:** El proceso de visitar completamente nuevas regiones de un espacio de búsqueda, para ver si puede encontrarse algo prometedor. A diferencia de la explotación, la exploración involucra grandes saltos hacia lo desconocido. Los problemas que tienen máximos locales pueden resolverse algunas veces sólo usando este tipo de búsqueda aleatoria.

- **Explotación:** Cuando se atraviesa un espacio de búsqueda, la explotación es el proceso de usar la información obtenida de los puntos visitados previamente en el espacio de búsqueda para determinar qué lugares resulta más conveniente visitar a continuación. Un ejemplo de esto es la técnica *escalando la colina* (*hill climbing*), que investiga puntos adyacentes en el espacio de búsqueda y se mueve en la dirección que proporciona el incremento más grande de aptitud. Las técnicas de explotación son buenas para encontrar óptimos locales.

- **Gene (o Gen):** Una sub-sección de un cromosoma que (usualmente) codifica el valor de un solo parámetro.

- **Generación:** Una iteración de la medida de aptitud y la creación de una nueva población por medio de operadores de reproducción.

- **Genoma:** La colección completa de genes (y por tanto cromosomas) que posee un organismo.

- **Genotipo:** La composición genética de un organismo (la información contenida en el genoma).
- **Individuo:** Un solo miembro de la población. Cada individuo contiene un cromosoma (de manera más general, un genoma) que representa una solución posible a la tarea en cuestión, es decir, un solo punto en el espacio de búsqueda. Usualmente se almacena también otro tipo de información en cada individuo (su aptitud, por ejemplo).
- **Migración:** La transferencia de (los genes de) un individuo de una sub-población a otra.
- **Mutación:** Un operador de reproducción que forma un nuevo cromosoma a través de alteraciones (usualmente pequeñas) de los valores de los genes de un solo cromosoma padre.
- **Operador de Reproducción:** Un mecanismo que influencia la forma en que se pasa la información genética de padres a descendientes durante la reproducción. Los operadores de reproducción caen en 3 amplias categorías: cruzamiento, mutación y selección.
- **Operador Genético:** Un operador de búsqueda que actúa sobre una estructura de codificación que es análoga al genotipo de un organismo (por ejemplo, un cromosoma).
- **Población Panmítica (o Panmíctica):** Una población mezclada en la cual cualquier individuo puede reproducirse con otro con una probabilidad que depende solamente de su aptitud. Lo opuesto es una población dividida en grupos (llamados sub-poblaciones) donde los individuos sólo pueden reproducirse con otros de la misma sub-población. La mayoría de los algoritmos evolutivos convencionales consideran poblaciones panmíticas.
- **Sub-Población:** Una población puede subdividirse en grupos a los que se les denomina sub-poblaciones, donde los individuos sólo se pueden cruzar con los que pertenezcan al mismo grupo (esta técnica suele usarse cuando se manejan procesadores en paralelo). Tales

subdivisiones pueden influenciar marcadamente la dinámica evolutiva de una población. Las sub-poblaciones pueden definirse mediante varias restricciones de migración: islas con migración arbitraria limitada; lugares de paso con migración a islas cercanas; aislamiento por distancia, en la que cada individuo se cruza sólo con vecinos cercanos.

- **Variedad:** En algoritmos genéticos cuando una población no tiene *variedad*, el cruzamiento no será útil como operador de búsqueda, porque estará propensa a sólo regenerar a los padres.

## II.3. ALGORITMOS GENÉTICOS

### II.3.1. DEFINICIÓN

El concepto de Algoritmos Genéticos fue propuesto de manera formal por John Holland de la Universidad de Michigan en la década de los sesenta (Holland, 1962)

Un Algoritmo Genético puede definirse como un procedimiento de optimización y búsqueda que crea generaciones de individuos cada vez mejores como posibles soluciones a un problema dado. La búsqueda se basa en el valor de *aptitud* asociado a cada individuo en una población. El valor de aptitud da una medida de qué tan buena es una solución en particular, es decir, qué tanto se acerca al valor óptimo. Este valor representa, en el proceso de selección, la probabilidad de que un individuo (propuesta de solución) sea elegido para evolucionar. El valor de aptitud también se utiliza como una medida de ordenamiento.

Un Algoritmo Genético utiliza las operaciones básicas de la Computación Evolutiva que son: selección, cruzamiento y mutación; aunque se han desarrollado muchas variaciones de estos operadores y algunos operadores nuevos intentando mejorar la calidad de las soluciones. Además de las operaciones básicas también se han incorporado alternativas para el procesamiento paralelo de los Algoritmos Genéticos.

## **II.3.2. ALGORITMO GENÉTICO SIMPLE**

### **II.3.2.1. Componentes Básicos**

En forma general un Algoritmo Genético se compone de los siguientes elementos (Goldberg, 1989):

- 1) Una representación de las soluciones potenciales del problema,
- 2) Una población inicial de posibles soluciones (normalmente se generan en un proceso aleatorio),
- 3) Una función de evaluación que juegue el papel del ambiente en el que se tiene que “evolucionar”, clasificando las soluciones en términos de su "aptitud",
- 4) Operadores genéticos que alteren la composición de los descendientes que se producirán para las siguientes generaciones,
- 5) Valores para los diferentes parámetros que utiliza el Algoritmo Genético (tamaño de la población, probabilidad de cruzamiento, probabilidad de mutación, número máximo de generaciones, etcétera).

### **II.3.2.2. Operación Básica**

Para la operación básica de un Algoritmo Genético Simple usualmente se genera una población inicial aleatoria, se hace una evaluación a la población generada, a cada individuo se le asigna un valor de aptitud para resolver el problema específico, se procede a generar una nueva población utilizando los operadores de reproducción, la nueva población sustituye a la anterior y el proceso de generación y sustitución se repite hasta encontrar una solución satisfactoria.

La operación básica de un Algoritmo Genético puede ilustrarse a través del siguiente segmento de pseudo-código (Buckles y Petry, 1992):

```

begin
  generar población inicial, G(0);
  evaluar G(0);
  t:=0;
  repeat
    t:=t+1;
    generar G(t) usando G(t-1);
    selección,
    cruzamiento,
    mutación,
    evaluar G(t);
  until encontrar una solución;
end

```

Como puede observarse, el Algoritmo Genético Simple es muy fácil de programar, sin embargo, para poder hacer que el algoritmo encuentre soluciones a un problema dado es necesario elegir correctamente la forma de representación, los operadores genéticos, las probabilidades a utilizar para cada operador y el método de selección.

### II.3.3. REPRESENTACIÓN

Algunas de las formas en las que usualmente se representan los individuos de una población en un algoritmo genético son: cadenas binarias (forma tradicional), códigos de *gray* (codificación binaria), punto flotante (en codificación binaria), punto flotante (en codificación real), punto flotante (en codificación entera), expresiones "S" del lenguaje LISP (por este tipo de representación se considera que la Programación Genética es un tipo particular de Algoritmo Genético), listas binarias de longitud variable (*messy-GA*) (Koza, 1992).

Para el problema abordado en esta tesis se utiliza la representación de punto flotante (codificación real), por lo que a continuación se describen los operadores usualmente empleados para este tipo de codificación.

## II.3.4. OPERADORES

### II.3.4.1. Selección

En un Algoritmo Genético es necesario definir una estrategia para la selección de los individuos que van a sobrevivir generación tras generación. Idealmente debe preferirse un método que permita mantener características de los mejores individuos, que explore nuevos espacios de búsqueda y que evite llegar a una convergencia prematura (Banzhaf *et al.*, 1998).

No existe un método que sea eficiente para cualquier tipo de problema (Fogel, 1999), pero existen algunos métodos para seleccionar adecuadamente los operadores (DeJong, 1975).

Algunos de los métodos de selección más frecuentemente usados y su clasificación se desglosan en la siguiente lista:

- ◆ Selección Proporcional
  - Ruleta (De Jong, 1975)
  - Sobrante Estocástico (Booker, 1982; Brindle, 1981)
  - Universal Estocástica (Baker, 1987)
  - Boltzmann (Coello, 1998)
  - Escalamiento Sigma (Mitchell, 1996)
- ◆ Selección por Jerarquías (Baker, 1985)
- ◆ Selección mediante Torneo (Coello, 1998)
- ◆ Selección de Estado Uniforme (Whitley, 1989)

La Selección Proporcional describe a un grupo de esquemas de selección que eligen individuos de acuerdo a los valores de sus funciones objetivo. El valor esperado de un individuo es la aptitud del individuo dividida entre la aptitud promedio de la población.

Existen algunas variantes de Selección Proporcional, entre ellas la técnica de Selección Universal Estocástica (Baker, 1987). Esta técnica trata de minimizar la mala distribución de



los individuos en la población en función de sus valores esperados. En vez de girar la ruleta  $N$  veces para seleccionar  $N$  padres (De Jong, 1975), se gira la ruleta sólo una vez, pero con  $N$  punteros equidistantes que se usan para seleccionar  $N$  padres. El número de copias que cada individuo recibe se calcula contando el número de marcadores que caen dentro de su segmento.

La Selección por Jerarquías es una técnica cuyo propósito es también prevenir una convergencia prematura (Baker, 1985), los individuos de la población se ordenan de mejor a peor en base a su aptitud y cada uno se selecciona tantas veces como corresponda de acuerdo a una función de asignación no lineal. Se efectúa una selección proporcional de acuerdo a dicha asignación. Esta técnica ofrece la ventaja de prevenir la convergencia prematura al evitar usar las aptitudes reales, pero la desventaja es que no sabremos que tanto supera un individuo a otro en términos de aptitud.

La Selección mediante Torneo es más eficiente y más adecuada para implementarse en paralelo que la Selección por Jerarquías porque utiliza pares de datos para la selección (Coello, 1998). En este caso se escoge aleatoriamente un número  $P$  de individuos (típicamente 2) de la población y se hacen competir entre sí con base en su aptitud. El individuo más apto es el que resultará ganador del torneo y se selecciona como padre para la siguiente generación. La población normalmente se debe barajar  $P$  veces, con lo que el orden de los individuos es alterado a fin de que puedan volver a seleccionarse. Esta técnica garantiza que el individuo más apto será seleccionado  $P$  veces, y que el menos apto nunca será seleccionado.

En la Selección de Estado Uniforme (Whitley, 1989), sólo unos cuantos individuos son reemplazados en cada generación. Esta técnica suele usarse cuando se evolucionan sistemas basados en reglas (ej. sistemas clasificadores) y en la que los miembros de la población resuelven colectivamente (y no de manera individual) el problema en cuestión. Goldberg y Deb (1991) han demostrado que la técnica de selección de estado uniforme incluida en Genitor (Whitley, 1989) es  $O(n \log n)$ .

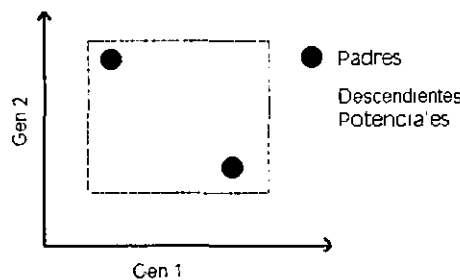
### II.3.4.2. Cruzamiento

Para los diferentes tipos de codificación existentes se utilizan diferentes tipos de cruzamiento y mutación, particularmente para codificación real se utilizan los métodos de recombinación intermedia, recombinación lineal y recombinación discreta (Chipperfield *et al*, 1994).

Recombinación intermedia: Funciona para valores reales codificados en estructura de cromosoma. Es un método que produce dos individuos a partir de los valores de los padres, estableciendo un área de posibles valores nuevos. La siguiente expresión define su comportamiento:

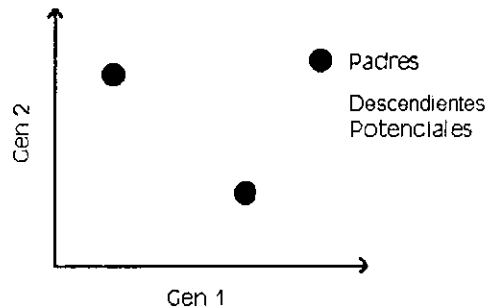
$$O_i = P_1 + \alpha(P_2 - P_1) \quad \dots\dots\dots (1)$$

donde  $\alpha$  es un factor de escalamiento seleccionado de forma aleatoria uniforme sobre algún intervalo, típicamente  $[-0.25, 1.25]$  y los cromosomas padres son  $P_1$  y  $P_2$ . Cada variable descendiente  $O_i$  es el resultado de la combinación de los padres de acuerdo con la expresión, donde se selecciona una  $\alpha$  para cada par de genes padres. En términos geométricos, la recombinación intermedia se puede visualizar como un espacio para generación de variables en forma de hipercubo ligeramente mayor que el formado por los padres pero limitado por el rango de  $\alpha$ .



**Figura Error! Unknown switch argument.:      **Recombinación****

Recombinación Lineal: Es similar a la recombinación intermedia, sólo que mientras en la recombinación intermedia se utiliza un hipercubo, en la recombinación lineal se utiliza una recta. Los descendientes pueden ser cualquier punto en la recta formada entre los padres, con las limitaciones impuestas por  $\alpha$ .



**Figura Error! Unknown switch argument.: Recombinación**

Recombinación discreta: Este tipo de recombinación es similar a un método para cruzamiento de cromosomas binarios llamado *cruzamiento uniforme*. El cruzamiento uniforme es una generalización de las formas más simples de cruzamiento para cadenas binarias (simple y multipunto). Para este tipo de cruzamiento se crea aleatoriamente una *máscara* binaria del mismo largo que la estructura del cromosoma. La paridad de los bits en la máscara indican cuál segmento de qué padre sustituirá al segmento correspondiente en los descendientes. Considere el siguiente ejemplo:

Padres

```
40.23 -17.17 28.95 15.38 %Padre 1
82.06 13.26 13.35 -9.09 %Padre 2
```

Máscara (generada aleatoriamente)

```
1 2 1 2 %Máscara 1
2 2 1 1 %Máscara 2
```

Descendientes

```
40.23 13.26 28.95 -9.09 %Máscara 1 y Padres 1 y 2
82.06 13.26 28.95 15.38 %Máscara 2 y Padres 1 y 2
```

### II.3.4.3. Mutación

En la evolución natural, la mutación es un proceso aleatorio donde un alelo de un gen es reemplazado por otro para producir una nueva estructura (Chipperfield *et al*, 1994). En Algoritmos Genéticos la mutación modifica elementos de los cromosomas y usualmente se utiliza con una baja probabilidad (Goldberg, 1989).

Para cada tipo de codificación se utiliza un operador de mutación adecuado. El operador de mutación para representaciones no binarias se puede aplicar modificando valores a un gen o con la selección aleatoria de nuevos individuos dentro del rango permitido (Chipperfield *et al*, 1994).

Se ha demostrado que los Algoritmos Genéticos de codificación real pueden obtener mejores resultados cuando utilizan una mayor probabilidad de mutación que los de codificación binaria en el mismo caso (Wright, 1991).

El operador de mutación para valores reales usa un término no lineal para la distribución del rango de mutación aplicado a los valores de los genes. La expresión para obtener un individuo mutante codificado en valor real (Mühlenbein, 1993), es la siguiente:

$$V_m = V * MutMx + R + \delta \quad \dots\dots\dots (2)$$

donde:

$V_m$  = Variable después de la mutación,

$V$  = Variable antes de la mutación,

MutMx = +1 ó -1 con igual probabilidad,

$R = 0.5 * \text{Dominio de la variable}$ ,

$$\delta = \sum_{i=0}^{m-1} \alpha_i 2^{-i}, \alpha_i = 1 \quad \text{con probabilidad } 1/m, \text{ en otro caso } 0, \text{ con } m=20.$$

Con  $m=20$  el operador de mutación puede localizar el óptimo con una precisión de  $R * 2^{-19}$ .

### II.3.5. ALGORITMOS GENÉTICOS PARALELOS

El acelerado desarrollo de las computadoras con arquitectura de varios procesadores ha propiciado que se busquen algoritmos y técnicas eficientes para aprovechar las ventajas que implica el utilizar esta tecnología. Los Algoritmos Genéticos proveen mejoras considerables al utilizar sub-poblaciones y ser implementados en varios procesadores, debido a que poseen características paralelas implícitas del propio algoritmo (Rodríguez, 1994).

La idea básica es utilizar un conjunto de Algoritmos Genéticos Nodales, cada uno de los cuales contiene sólo una parte de la población general. Cada nodo ejecuta un Algoritmo Genético Simple que evoluciona independiente a los demás.

Existen varios modelos para los Algoritmos Genéticos Paralelos (AGPs), entre los que destacan *farming*, *migración* y *difusión* (Rodríguez, 1994).

El modelo de *farming* (East y McFarlane, 1993) utiliza un proceso maestro y N procesos esclavos. El proceso maestro distribuye las tareas y posteriormente recibe los resultados. En los AGPs que utilizan este modelo un procesador maestro se encarga de supervisar la población genética total y de realizar la selección. Los procesadores esclavos se encargan de generar a los nuevos pobladores.

En el modelo de migración (Tanese, 1989), el conjunto de individuos con los cuales un miembro de una población puede cruzarse se le conoce como *deme*. Si el *deme* para todos los miembros de una población es igual a la población completa, entonces estamos hablando de una población panmítica. En este modelo usualmente se utiliza más de un *deme*, es decir, se utilizan sub-poblaciones. En intervalos arbitrarios de tiempo pueden moverse individuos de una sub-población a otra.

En el modelo de difusión (East y McFarlane, 1993) se pueden utilizar estructuras similares a las de migración, pero las sub-poblaciones no se encuentran totalmente aisladas, sino que

se tiene un acceso aleatorio a todos los individuos en algún tiempo. Los individuos están colocados dentro de una vecindad y todos pueden estar en continuo movimiento, lo cual permite el acceso para la recombinación.

Particularmente el modelo de migración ha proporcionado buenos resultados en aplicaciones de alto rendimiento (Rodríguez, 1994). Este modelo tiene la ventaja de que incluso utilizando una arquitectura de computadora de un sólo procesador se puede implementar una aplicación concurrente para simularlo (Chipperfield *et al*, 1994).

### **II.3.6. APLICACIONES**

Se han desarrollado una gran variedad de Algoritmos Genéticos y actualmente hay muchas líneas de investigación relacionadas. Algunos de los problemas que han sido abordados con Algoritmos Genéticos y sus relaciones con algunas otras ramas de la ciencia se mencionan a continuación:

**Optimización** : Se han usado en un gran número de problemas de optimización numérica y combinatoria, tales como el del agente viajero, asignación cuadrática, diseño estructural, diseño de plantillas de circuitos integrados, etcétera. (Haftka, 1998; Vázquez, 2000)

**Aprendizaje de Máquina:** Clasificación y predicción de tareas. Por ejemplo, predicción del clima o estructura de proteínas. También se han usado para evolucionar ciertos aspectos de algunos sistemas de aprendizaje de máquina, como por ejemplo los pesos de las redes neuronales, las reglas de aprendizaje de los sistemas clasificadores o los sistemas de producción simbólicos. (Golberg, 1989; Carse, 2000).

**Economía** : Se han usado para modelar procesos de innovación, el desarrollo de estrategias de regateo, y el comportamiento de mercados económicos emergentes. (Chen, 1998).

**Sistemas Inmunológicos:** Se han usado para modelar varios aspectos de los sistemas inmunológicos naturales, incluyendo la mutación somática durante el período de vida de un individuo y el descubrimiento de familias genéticas durante el tiempo de evolución. (Banzhaf *et al.*, 1998).

**Ecología :** Se han usado para modelar fenómenos ecológicos tales como la coevolución de los parásitos, la simbiosis, y el flujo de recursos. (Banzhaf *et al.*, 1998).

**Genética de población:** Se han usado para estudiar interrogantes acerca de la genética de población tales como "¿bajo qué condiciones es un gen de recombinación viable desde el punto de vista evolutivo?". (Mitchell, 1996).

**Evolución y Aprendizaje:** Se han usado para estudiar la forma en que interactúan el aprendizaje de un individuo y la evolución de su especie. (Hinton y Nowland, 1987; Nolfi, 1994)

**Sistemas Sociales:** Se han usado para estudiar aspectos evolutivos de los sistemas sociales tales como la evolución del comportamiento social en las colonias de insectos y, de manera más general, la evolución de la cooperación y la comunicación en los sistemas de multi-agentes. (Neeraj y Sandip, 1997).

**Minería de Datos:** Se han usado para optimizar consultas a sistemas de bases de datos. (Freitas, 2000).

**Procesamiento de Imágenes:** Se han usado para detección de ciertas características en las imágenes y para reconocimiento de patrones. (Bahnu *et al.*, 1995).

## II.4. PROGRAMACIÓN GENÉTICA

### II.4.1. DEFINICIÓN

La Programación Genética puede ser considerada como una sub-clase de los Algoritmos Genéticos con una estructura de representación más compleja y poderosa basada en árboles. Los árboles representan normalmente estructuras ejecutables (programas) de tamaño, contenido y forma variables.

Al igual que en el caso de los Algoritmos Genéticos, el paradigma de la Programación Genética continua con la tendencia de solucionar el problema de representación y lo hace incrementando la complejidad de las estructuras que buscan adaptarse.

En Programación Genética se utilizan poblaciones de cientos o miles de programas de computadora para reproducirse. La reproducción se hace siguiendo los principios de supervivencia de la teoría de Darwin. Como se verá más adelante, con esta técnica se puede obtener un programa que resuelve, o aproximadamente resuelve, un problema dado (Koza, 1992).

La Programación Genética inicia con una población inicial de programas de computadora los cuales son generados aleatoriamente y se componen de funciones y terminales definidos de acuerdo al contexto del problema. Las funciones pueden ser operaciones aritméticas, funciones de programación, funciones matemáticas, funciones lógicas o funciones de algún dominio particular. Dependiendo del problema, el programa de computadora puede proporcionar valores de tipo booleano, entero, complejo, vector, simbólico o múltiple.

Podemos pensar que la creación de la población inicial es una búsqueda aleatoria *ciega* del espacio de búsqueda del problema. Cada programa individual en la población, se evalúa en términos de que tan bien resuelve el problema en particular y a esta medida se le llama *aptitud*, la cual varía de acuerdo al tipo de problema.



Típicamente, se corre cada uno de los programas de la población sobre un número determinado de casos y la aptitud se mide como la suma o el promedio de varios casos representativos de diferentes situaciones. Por ejemplo, la aptitud de un programa en particular en la población puede ser medida como la suma del valor absoluto de las diferencias producidas por los resultados del programa y los resultados correctos (error absoluto).

El principio de Darwin para selección y supervivencia y la operación sexual de recombinación (cruzamiento) se usa para crear una nueva población de programas descendientes. El proceso genético de reproducción sexual entre dos programas de computadora *padres* se usa para crear nuevos programas de computadora descendientes. Los padres se seleccionan en proporción a su aptitud.

Los programas padre son típicamente de diferentes tamaños y formas. Los programas descendientes se componen de sub-expresiones (sub-árboles, sub-programas, sub-rutinas o bloques constructores) de sus padres.

Intuitivamente, si dos programas son eficientes para resolver un problema, entonces las partes que lo componen algún mérito deben de tener. Por medio de la recombinación aleatoria de algunas de las partes de programas de alguna forma eficientes, podemos producir programas que sean incluso mejores para resolver el problema particular dado.

Después de las operaciones de selección y cruzamiento en la población, la nueva generación obtenida sustituye a la anterior. Cada individuo en la nueva población de programas de computadora es evaluado de acuerdo a su aptitud, y el proceso se repite hasta satisfacer un criterio de paro.

A cada paso del proceso, el estado presente consistirá solamente de la población actual. Algunas características interesantes de esta técnica son que es un proceso intrínsecamente paralelo, controlado localmente y descentralizado.

Este modelo produce programas de computadora tales que, después de varias generaciones, tienden a mejorar incrementando su aptitud promedio de acuerdo al contexto del problema. Además, tiene la ventaja de que se pueden adaptar rápidamente a cambios en el ambiente de trabajo.

En resumen, el paradigma de la programación genética crea programas para resolver problemas siguiendo los siguientes pasos:

- 1) Generar una población inicial aleatoria de estructuras de funciones y terminales del problema (programas de computadora).
- 2) Realizar los siguientes sub-procesos hasta que el criterio de terminación quede satisfecho:
  - a) Ejecutar cada programa en la población y asignarle un valor de aptitud de acuerdo a su capacidad para resolver el problema.
  - b) Crear una nueva población de programas de computadora aplicando dos operadores primarios. Los operadores se aplican a programas seleccionados de la población con una probabilidad basada en su aptitud.
    - i) Copiar programas existentes a la nueva población.
    - ii) Crear nuevos programas de computadora por medio de recombinación genética seleccionando aleatoriamente partes de dos programas existentes.
    - iii) La mutación se aplica con probabilidad mínima ya que la variedad no se afecta con el operador de recombinación.
- 3) El mejor programa de computadora que haya aparecido en cualquier generación se designa como el resultado de la programación genética. Este resultado debe ser una solución (o una aproximación a la solución) del problema.

## II.4.2. TIPOS DE REPRESENTACIÓN

Las estructuras individuales utilizadas en Programación Genética como elementos a ser evolucionados, son programas de computadora estructurados jerárquicamente. El tamaño, la forma y el contenido de estos programas pueden cambiar dinámicamente durante el proceso.

El conjunto de posibles estructuras en Programación Genética es el conjunto de todas las funciones que puedan ser compuestas recursivamente del conjunto  $N_{func}$  de  $F=\{f_1, f_2, \dots, f_{N_{func}}\}$  y el conjunto de terminales  $N_{term}$  de  $T=\{a_1, a_2, \dots, a_{N_{term}}\}$ . Cada función particular  $f_i$  en el conjunto de funciones  $F$  tiene un número particular de argumentos  $z(f_i)$ . Al número de argumentos  $z(f_i)$  se le llama *aridad* de  $f_i$ .

El conjunto de funciones pueden incluir:

- operaciones aritméticas (+, -, \*, ...,etc.),
- funciones matemáticas (seno, coseno, exp, log, ..., etc.),
- operaciones booleanas (AND, OR, NOT, ..., etc.),
- operadores condicionales (ej. If-then-else),
- funciones iterativas (ej. Do-until),
- funciones recursivas,
- alguna otra función del dominio específico del problema que pueda ser definida.

Las terminales pueden ser variables (representando tal vez las entradas, sensores, detectores o el estado de algún sistema) o constantes (como el número 5 o la constante booleana NIL).

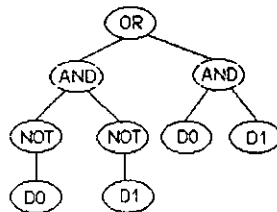
Como ejemplo considere el conjunto de funciones  $F=\{AND, OR, NOT\}$  y el conjunto de terminales  $T=\{D0, D1\}$  donde  $D0$  y  $D1$  son variables booleanas que sirven como argumentos para las funciones.

Se puede combinar el conjunto de funciones  $C = F \cup T = \{AND, OR, NOT, D0, D1\}$ . Entonces el conjunto combinado  $C$  tiene elementos terminales que requieren de cero argumentos para ser evaluados. Esto es, los cinco elementos del conjunto  $C$  requieren 2, 2, 1, 0 y 0 argumentos, respectivamente.

Siguiendo el ejemplo, consideremos la función “no XOR” (verdadero sólo cuando ambos valores coinciden), que es una función de 2 argumentos y retorna verdadero cuando un número par de sus argumentos son iguales, en otro caso retorna NIL. Esta función booleana puede expresarse en prefijo o forma normal disyuntiva (DNF, *Disjunctive Normal Form*) con la siguiente expresión  $S$  de LISP:

```
(OR (AND (NOT D0) (NOT D1)) (AND D0 D1))
```

O gráficamente con el árbol:



**Figura** Error! Unknown switch argument.:

Los cinco puntos internos son funciones, los cuatro puntos externos son terminales, la raíz del árbol está etiquetada con la función más externa de la expresión prefija (el OR). El árbol es equivalente al árbol de recorrido que la mayoría de los compiladores construye internamente para representar un programa de computadora dado.

### II.4.3. FUNCIONES DE APTITUD

La aptitud es la fuerza que controla la selección natural de Darwin. En la naturaleza, la aptitud de un individuo es la probabilidad de supervivencia hasta la etapa de reproducción. La aptitud se mide de alguna forma y usando entonces esta medida se controla la aplicación de las operaciones que modifican las estructuras de la población.

Existen varias formas de medir la aptitud, algunas de forma explícita, otras de forma implícita. La forma más común de medir la aptitud es crear una medida explícita de aptitud para cada individuo. A cada individuo en la población se le asigna un valor escalar como medida de aptitud que expresa la capacidad de ese programa para solucionar el problema.

Algunas medidas de aptitud que se han estudiado son: aptitud estandarizada (*standardized fitness*), aptitud ajustada (*adjusted fitness*) y aptitud normalizada (*normalized fitness*) (Koza, 1992). Estas medidas pueden ser aplicadas en Algoritmos Genéticos y en Programación Genética con la restricción de que en Algoritmos Genéticos están dadas en términos de qué tan bien se adapta un individuo en particular a la solución de la función dada y en Programación Genética se aplican normalmente en términos de minimización del error.

La aptitud  $r(i,t)$ , también conocida como *raw fitness*, queda expresada en la terminología natural del problema por sí misma. Por ejemplo, la aptitud en el problema clásico de la hormiga artificial (Koza, 1992) sería el número de piezas de comida que son transportadas, entre más comida mejor. Así, la aptitud para este problema tomaría valores enteros de 0 a 89 que es el máximo posible de piezas a transportar.

Cuando la función de aptitud es el *error*, la aptitud para cada función  $i$  en la población puede ser expresada en cualquier generación  $t$ :

$$r(i,t) = \sum_{j=1}^{N_s} S(i,j) - C(j) \quad \dots\dots\dots (3)$$

donde  $S(i,j)$  es el valor retornado por el programa  $i$ -ésimo para el caso de aptitud  $j$  (de  $N_c$  casos) y donde  $C(j)$  es el valor correcto para el caso de aptitud  $j$ .

### II.4.3.1. Aptitud Estandarizada

La aptitud estandarizada resta la aptitud de forma que el valor numérico menor es siempre el mejor valor. Para un problema particular donde a menor valor de la función es mejor aptitud, la aptitud estandarizada es igual a la aptitud tal cual. Esto es:  $s(i,t) = r(i,t)$ .

De la misma forma, en problemas donde a mayor valor de la función se tiene una mejor aptitud, la aptitud estandarizada se expresa como el mayor valor posible, denotado por  $r_{\max}$  menos el valor observado, esto es:  $s(i,t) = r_{\max} - r(i,t)$

### II.4.3.2. Aptitud Ajustada

La aptitud ajustada  $a(i,t)$  se obtiene a partir de la aptitud estandarizada  $s(i,t)$ , como sigue:

$$a(i,t) = \frac{1}{1 + s(i,t)} \dots\dots\dots (4)$$

La aptitud ajustada toma valores entre 0 y 1, y los valores mayores se dan a los mejores individuos de la población.

### II.4.3.3. Aptitud Normalizada

Si el método de selección usado es proporcional a la aptitud, también es necesario el concepto de aptitud normalizada. La *aptitud normalizada*  $n(i,t)$  se calcula a partir de la aptitud ajustada  $a(i,t)$  y el tamaño de la población  $M$  como sigue:

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^M a(k,t)} \dots\dots\dots (5)$$

La aptitud normalizada tiene tres características deseables:

- Valores entre 0 y 1.
- Mayor para mejores individuos en la población.
- La suma de aptitudes normalizadas es 1.

### II.4.4. OPERACIONES PARA MODIFICAR ESTRUCTURAS

Existen varias operaciones para modificar las estructuras (programas) que componen la población usada en Programación Genética. Las principales operaciones son *selección* y *cruzamiento*, y las operaciones secundarias, menos utilizadas son: *mutación*, *permutación*, *edición* y *encapsulación*.

#### II.4.4.1. Selección

La operación de selección pretende reproducir el mecanismo básico de selección natural Darwiniana y de supervivencia para Programación Genética. La operación de selección es asexual ya que opera con un sólo padre (expresión prefija o sub-programa) y produce un sólo descendiente (expresión prefija o sub-programa) cada ocasión que se realiza.

La operación se efectúa en dos pasos. Primero, se selecciona una expresión de la población de acuerdo a algún método de selección, y segundo, el individuo seleccionado se copia sin sufrir alteración de la población actual a la nueva población.

La forma más común de selección para la reproducción es la basada en la aptitud (Holland, 1975). Si  $f(s_i(t))$  es la aptitud del individuo  $s_i$  en la generación  $t$ , entonces bajo este esquema de selección, la probabilidad de que el individuo  $s_i$  sea copiado en la próxima generación de la población como el resultado de la operación de selección es

$$P(s_i(t)) = \frac{f(s_i(t))}{\sum_{j=1}^M f(s_j(t))} \dots\dots\dots (6)$$

Cuando la operación de reproducción se basa en selección proporcional a la aptitud se le llama simplemente *reproducción proporcional a la aptitud*.

**II.4.4.2. Cruzamiento**

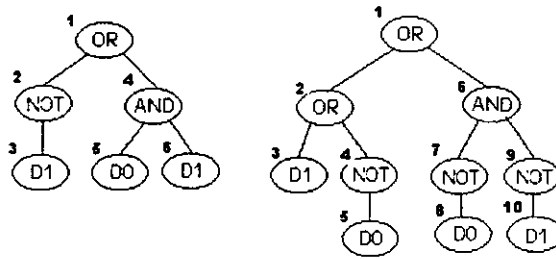
El operador de cruzamiento o recombinación sexual introduce diversidad en la población produciendo descendientes con partes tomadas de cada padre. La operación de cruzamiento comienza con dos padres (expresiones) que producen dos descendientes (expresiones). Los padres son seleccionados de la población de acuerdo al valor de su aptitud.

La operación comienza seleccionando aleatoriamente un punto de cruzamiento en cada uno de los padres. Dicha selección se hace usando una distribución de probabilidad uniforme. Nótese que los padres son típicamente de tamaños diferentes.

Los descendientes se producen eliminando los fragmentos seleccionados a partir del punto de cruzamiento en ambos padres y colocándolos como sub-árboles del otro padre a partir del punto de cruzamiento.



Por ejemplo, si consideramos dos padres:



**Figura Error! Unknown switch argument.: Ejemplo de Cruzamiento (Padres)**

Como expresión prefija:

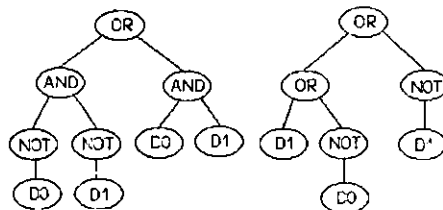
(OR (NOT D1) (AND D0 D1))

y

(OR (OR D1 (NOT D0)) (AND (NOT D0) (NOT D1)))

Asumiendo que los puntos de ambos árboles están numerados primero por la raíz y de izquierda a derecha. Supongamos que aleatoriamente quedan seleccionados el punto 2 y el punto 6 como punto de cruzamiento en el primero y segundo padre respectivamente.

Los descendientes quedarían descritos como:



**Figura Error! Unknown switch argument.: Ejemplo de Cruzamiento**

Como expresión prefija:

(OR (AND (NOT D0) (NOT D1)) (AND D0 D1))

y

(OR (OR D1 (NOT D0)) (NOT D1))

Los árboles que se producen por medio de cruzamiento son siempre válidos debido ya que son simplemente intercambios de segmentos de otras expresiones (Koza, 1992).

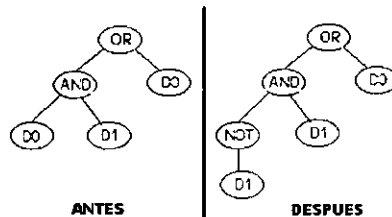
A diferencia del caso de Algoritmos Genéticos, el cruzamiento de dos individuos idénticos debe producir descendientes diferentes, debido a que el punto de cruzamiento rara vez es el mismo.

Debe establecerse un tamaño máximo permisible para los descendientes creados por medio de la operación de cruzamiento. El tamaño se mide de acuerdo a la profundidad del árbol o al número de nodos. Este límite previene que el tiempo de cómputo se haga excesivamente lento o que las expresiones S se alarguen demasiado. Si una operación de cruzamiento va a generar descendientes que excedan el tamaño permitido, la operación se cancela y uno de los padres (elegido aleatoriamente) se reproduce para la siguiente generación. Nótese que el otro descendiente necesariamente debe ser de tamaño permitido. Si el cruzamiento se cancela porque ambos descendientes son demasiado grandes, entonces ambos padres se reproducen para la siguiente generación.

#### **II.4.4.3. Mutación**

La operación de mutación introduce cambios aleatorios en las estructuras de la población. En los Algoritmos Genéticos convencionales la operación de mutación puede beneficiar dando diversidad a la población para evitar convergencia prematura. En Programación Genética se utiliza con baja probabilidad debido a que no es muy común la convergencia prematura.

En esta operación se selecciona aleatoriamente un *punto de mutación* en un sólo padre. El punto de mutación puede ser interno o externo (función o terminal). Entonces la operación de mutación elimina lo que esté en la sub-expresión asociada a ese punto y en su lugar pone otro segmento cualquiera de expresión.

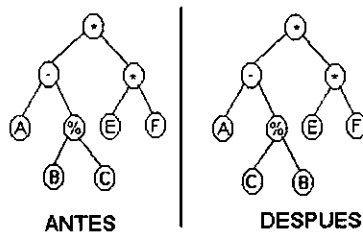


**Figura** Error! Unknown switch argument.: **Ejemplo de**

En la figura 6 se muestra un ejemplo en el que aleatoriamente es seleccionado el nodo D0 (de la rama izquierda) y se le aplica el operador de mutación. Como en otras operaciones, se debe controlar el tamaño máximo permitido, que típicamente es el mismo especificado como máximo en la población inicial.

#### **II.4.4.4. Permutación**

La operación de permutación es asexual, utiliza un sólo padre y produce un sólo descendiente. El individuo se selecciona de la misma forma que en las operaciones anteriormente descritas. La operación de permutación inicia con la selección de un punto correspondiente a una función (punto interno), si la función del punto seleccionado tiene  $k$  argumentos entonces se selecciona una permutación de un conjunto de  $k!$  posibles.



**Figura** Error! Unknown switch argument.: **Ejemplo de**

La figura 7 muestra la operación de permutación donde aleatoriamente se selecciona el punto de la función “%” y los operandos (B y C) son permutados. Obviamente, si la función seleccionada tiene la propiedad de ser conmutativa, la operación no producirá efecto alguno.

#### II.4.4.5. Edición

La operación de edición permite simplificar expresiones mientras el programa genético está corriendo. La operación de edición es asexual, un sólo padre y un sólo descendiente.

La operación de edición aplica recursivamente una serie de reglas preestablecidas de dominio dependiente a cada expresión en la población. La regla universal de dominio dependiente es la siguiente: si una función no afecta a otras, no depende del contexto y tiene solamente constantes como argumentos, será evaluada por la operación de edición y se le substituirá con el valor obtenido por la función.

Por ejemplo, la expresión numérica (+ 1 2) será reemplazada por 3 y la expresión booleana (AND T T) será reemplazada por T (*true*).

Para dominios booleanos algunas reglas de edición podrían ser:

$$(\text{AND } X \ X) \rightarrow X$$
$$(\text{OR } X \ X) \rightarrow X$$
$$(\text{NOT } (\text{NOT } X)) \rightarrow X$$

La operación de edición persigue dos objetivos, el primero es *cosmético*, es decir, hacer la salida del programa más comprensible, y el segundo es hacer que la salida obtenida en cada generación sea más simple y así disminuir el tiempo de ejecución del programa.

La operación de edición se controla con un parámetro de frecuencia que determina cada cuantas generaciones se edita. Si  $f_{ed}$  (frecuencia de edición) es 0, entonces no se realiza la operación, si es 1 entonces cada generación se edita, un número entero mayor a 1 indica la edición cada  $f_{ed}$  número de generaciones.

#### **II.4.4.6. Encapsulación**

La operación de encapsulación significa que automáticamente puede ser identificado un sub-árbol potencialmente útil, darle un nombre de forma que pueda ser referenciado posteriormente.

La función de encapsulación es asexual, un sólo padre y un sólo descendiente. Se selecciona un punto interno de la expresión aleatoriamente, se borra el sub-árbol seleccionado y se substituye por una etiqueta que representa un segmento de programa que posteriormente puede ser llamado.

La nueva función *encapsulada* no usa argumentos, simplemente substituye el segmento de programa que estaba anteriormente y facilita la comprensión en el resultado.

Por ejemplo, considérese la expresión:

```
(+ A (* B C))
```

Supongamos que aleatoriamente se selecciona el punto 3 (multiplicación) como el punto para aplicar encapsulación. La función E0 en notación de LISP se crea como sigue:

```
(defun E0 ()  
  (* B C)  
)
```

Y el resultado es:

```
(+ A (E0))
```

La ventaja de la encapsulación radica en que se pueden compilar nuevas funciones y subfunciones “al vuelo” durante la ejecución total del programa.

#### **II.4.5. APLICACIONES**

Una gran variedad de problemas, de campos muy diferentes entre sí, pueden ser resueltos encontrando un programa de computadora que produzca una salida deseada para una entrada en particular, es decir, programas por inducción (Koza, 1992).

La Programación Genética puede buscar en un espacio de posibles programas de computadora para encontrar uno solo de ellos que resuelva (o aproximadamente resuelva) el problema en cuestión.

Algunos de los problemas que pueden ser atacados con Programación Genética son:

**Control óptimo:** Requiere de la búsqueda de estrategias de control que utilicen el estado actual de las variables de un sistema para seleccionar valores de las variables de control y así pasar al estado deseado del sistema, minimizando algún valor de costo. (Koza, 1992).

**Planeación:** En Inteligencia Artificial y en Robótica se requiere encontrar un plan de acción o movimiento con la información recibida del medio ambiente por medio de sensores o detectores que indican el estado de varios objetos en el sistema y entonces se usa esa información para seleccionar acciones o movimientos que cambien el estado actual. (Miyashita, 2000).

**Estrategias de juegos:** Se requiere encontrar una estrategia que especifique que movimiento hacer en cada punto del juego, dando la información que se tiene acerca del estado actual. En el juego, la información puede ser una historia explícita de los movimientos hechos anteriormente por los jugadores, o implícita como el estado actual del juego (en ajedrez, por ejemplo, la posición de cada pieza en el tablero). (Barone y While, 2000).

**Regresión simbólica:** Implica encontrar una expresión matemática de forma simbólica que se aproxime lo más posible a un conjunto finito de muestras de valores de variables independientes y los valores correspondientes de las variables dependientes. Esto es, regresión simbólica implica encontrar un modelo apto para un conjunto de datos dado. (Koza, 1992).

**Programación automática:** Se refiere a una fórmula matemática para resolver un problema particular que inicia con ciertos valores dados (entradas) y produce ciertos resultados deseados (salidas). (Koza, 1992)

**Problemas inversos:** encontrar una función inversa de una curva dada implica encontrar una función matemática en forma simbólica, que defina el inverso de la curva. Se procede como en regresión simbólica y se busca una expresión matemática apta para los datos de una muestra finita. (Banzhaf *et al.*, 1998).

## II.5. RESUMEN

La Computación Evolutiva es una rama de la Inteligencia Artificial que trata de imitar el comportamiento evolutivo de la naturaleza para la solución a problemas complejos. Presenta dos características interesantes: es capaz de generar por sí misma el conocimiento que requiere y selecciona aleatoriamente nuevas alternativas de solución para seguir generando nuevo conocimiento.

La terminología utilizada en Computación Evolutiva tiene similitud con la utilizada en Biología, pero el contexto es diferente. Esta terminología se puede aplicar a dos de las técnicas que gozan de más popularidad actualmente: Algoritmos Genéticos y Programación Genética. Estas dos técnicas se han utilizado con éxito en aplicaciones de índole muy diversa y presentan características que las hacen deseables para el problema abordado en esta tesis.

En el siguiente capítulo se presenta el problema de la neurocisticercosis desde un panorama general, los métodos de diagnóstico existentes, el método Western Blot y algunas propuestas de solución que utilizan técnicas de Computación Evolutiva.

## II.6. REFERENCIAS

Bäck, Thomas. 1996. *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York.

Bahnu, B.; Lee, S. y Ming, J. 1995. "Adaptive Image Segmentation using a Genetic Algorithm". *IEEE Tran. on Systems, Man and Cybernetics*. Vol. 25, 12:1543-1567

Baker, J. E. 1985. "Adaptive selection methods for genetic algorithms", *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pp. 100-111.



- Baker, J. E. 1987. "Reducing bias and inefficiency in the selection algorithm", *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14-21.
- Banzhaf, W.; Nordin, P.; Keller, R.E. y Francone, F.E. 1998. *Genetic Programming: An Introduction*. Morgan Kaufmann Publishers. San Francisco. California, EUA.
- Barone, L. Y While, L. 2000. "Adaptive Learning for Poker". *Proceedings of the Genetic and Evolutionary Computation Conference*. Darrel Whitley, *et al* editores. Págs. 566-573. Jul. 10-12. Las Vegas, Nevada, EUA. Morgan Kafumann.
- Booker, L. B. 1982. "Intelligent behavior as an adaptation to the task environment", PhD Thesis, University of Michigan at Ann Arbor.
- Brindle, A. 1981. "Genetic Algorithms for Function Optimization", PhD Thesis, Department of Computer Science, University of Alberta.
- Buckles, Bill P. y Petry, Frederick E. 1992. *Genetic Algorithms*, IEEE Computer Society Press.
- Carse B. y Orelan J. 2000. "A note on learning and evolution in Neural Networks". *Proceedings of the Genetic and Evolutionary Computation Conference*. Darrel Whitley, *et al* editores. Págs. 66-73. Jul. 10-12. Las Vegas, Nevada, EUA. Morgan Kafumann.
- Chen, S.D. 1998. "Evolutionary computation in financial engineering: a roadmap of GAS and GP". *Financial engineering news*. 2(4):1-10.
- Chipperfield A.; Fleming P.; Pohlheim H.; Fonseca C. 1994. *Genetic Algorithm Toolbox for use with Matlab. User's Guide*. Department of Automatic Control and Systems Engineering. University of Sheffield. Reino Unido.

Coello, Carlos A. 1998. "Introducción a la Computación Evolutiva".

(Documento Web).

<http://www.lania.mx/~ccoello/genetic.html>

25 de abril del 2000.

De Jong, Kenneth A. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD Thesis, University of Michigan at Ann Arbor. Michigan. EUA.

East, I.R. y Macfarlane, D. 1993. "Implementation in OCCAM of Parallel Genetic Algorithms on Transputer Network". *Parallel Genetic Algorithms: theory and Applications*. Joachim Stender (Ed.), IOS Press.

Fogel, L. J.; A. J. Owens y M. J. Walsh. 1965. "Artificial Intelligence through a Simulation of Evolution", En A. Callahan, M. Maxfield & L. J. Fogel (Editores), *Biophysics and Cybernetics Systems*, Spartan Books, Washington, D.C., pp. 131-156.

Fogel, D.B. 1999. "Some Recent Important Foundational Results in Evolutionary Computation". *Evolutionary Algorithms in Engineering and Computer Science*. Wiley.

Freitas, A. 2000. "Data mining with Evolutionary Algorithms". *2000 Genetic and Evolutionary Computation Conference, tutorial Program*. Jul. 10-12. Las Vegas, Nevada, EUA. Pp. 514-537.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning* Addison-Wesley.

- Goldberg, David E. y Deb Kalyanmoy. 1991. "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms". *Foundations of Genetic Algorithms*, (Gregory J. y Rawlins E. Eds.) Morgan Kaufmann Publishers, San Mateo, California, pp. 69-93.
- Hafta, R.T. 1998. "Genetic Algorithms for optimization of composite laminates". En Mota *et al* (Eds.). *Mechanics of composite materials and structures*, 2, NATO ASI, Troia, Portugal, Pp. 141-152.
- Hinton, G.E. y Nowlan, S.J. 1987. "How learning can guide evolution". *Complex Systems*. 1:195-502.
- Holland, John H. 1962. "Outline for a logical theory of adaptive systems", *Journal of the Association for Computing Machinery*, 3:297-314.
- Holland, John H. 1975. *Adaptation in Natural and Artificial Systems*, University of Michigan Press. (Editado desde 1992 por el MIT Press).
- Koza, John R. 1992. *Genetic Programming: On the programming of Computers by Means of Natural Selection*. The MIT Press. Boston, Massachusetts, EUA.
- Mitchell, Melanie. 1996. *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Massachusetts.
- Miyashita, K. 2000. "Job-Shop Scheduling with Genetic Programming". *Proceedings of the Genetic and Evolutionary Computation Conference*. Darrel Whitley, *et al* editores. Págs. 505-512. Jul. 10-12. Las Vegas, Nevada, EUA. Morgan Kafumann.
- Mühlenbein, H. Y Schlierkamp-Voosen. 1993. "Predictive Models for the Breeder Genetic Algorithm". *Evolutionary Computation*. Vol. 1, No. 1, pp. 25-49.

- Neeraj A. y Sandip S. 1997. "Resolving social dilemmas using Genetic Algorithms: Initial results". In *Proceedings of the 7<sup>th</sup> International Conference on Genetic Algorithms*. Pp. 689-695. San Mateo, Ca. Morgan Kauffmann.
- Nolfi, S. 1994. "Learning and evolution in neural networks". *Adaptive Behaviour*. 3:5-28.
- Rechemberg, I. 1973. *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog Verlag, Stuttgart.
- Rodríguez Vázquez, Katya. 1994. *Algoritmos Genéticos Paralelos en Estimación Espectral Doppler*. Tesis de Licenciatura. Ingeniería en Computación. Facultad de Ingeniería. UNAM. DF. México.
- Tanese, R. 1989. "Distributed Genetic Algorithms". *Proceedings of Third International Conference on Genetic Algorithms*. San Mateo. California. Morgan Kauffman. Pp. 434-439.
- Vázquez M. y Whitley. 2000. "A Hybrid Genetic Algorithm for the Quadratic Assignment Problem". *Proceedings of the Genetic and Evolutionary Computation Conference*. Darrel Whitley, *et al* editores. Págs. 135-142. Jul. 10-12. Las Vegas, Nevada, EUA. Morgan Kafumann.
- Whitley, D. 1989. "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best", *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 116-121.
- Wright, A.H. 1991. "Genetic Algorithms for Real Parameter Optimization". In *Foundations of Genetic Algorithms*, J.E. Rawlins (Ed.), Morgan Kauffman, pp. 205-218.

### III. NEUROCISTICERCOSIS

#### III.1. INTRODUCCIÓN

La cisticercosis es una enfermedad que afecta a seres humanos y a otros mamíferos (ej. cerdos). Es provocada por la presencia de un parásito conocido comúnmente como cisticerco (*Cysticercus cellulosae*), siendo este una forma intermediaria de vida de la solitaria (*Taenia solium*). Cuando esta enfermedad se presenta en el sistema nervioso central es conocida como neurocisticercosis (Aluja *et al.*, 1987).

La sintomatología de esta enfermedad es variada, desde totalmente asintomática, pasando por la manifestación de diversas molestias como migrañas, cefaleas atípicas, convulsiones, trastornos emocionales discretos, cuadros epilépticos e hipertensión craneal hasta enfermedades bien definidas como la esclerosis múltiple, meningitis o trastornos mentales graves como la esquizofrenia. Cifras estadísticas muestran que si la enfermedad se desarrolla libremente puede sobrepasar el 50% de mortalidad (Robles, 1997; Aluja, 1987).

Se ha calculado que un número significativo de individuos con cisticercosis, probablemente más del 50%, permanece asintomático o con mínimos síntomas durante el curso de la enfermedad. Esto puede provocar que la enfermedad nunca se detecte o que sea detectada demasiado tarde.

Existen varios métodos de diagnóstico para la cisticercosis, como son la radiografía simple de cráneo, la tomografía computarizada, el estudio del líquido cefalorraquídeo y algunos tipos de inmunodiagnóstico que han dado relativamente buenos resultados. Todos estos métodos presentan deficiencias para poder hacer un diagnóstico confiable debido a la complejidad de la sintomatología que presenta la enfermedad y a las características del parásito.

Los métodos serológicos, como también se le conoce al inmunodiagnóstico, son los que ofrecen mejores perspectivas para la vigilancia epidemiológica ya que se realizan con muestras de sangre o de líquido cefalorraquídeo (LCR) y son potencialmente sensibles y específicos (Aluja *et al.*, 1987). Por tanto, se busca mejorar estos métodos debido a su mayor viabilidad al abarcar grandes grupos de la población y a su bajo costo.

Otra de las ventajas de mejorar el inmunodiagnóstico de esta enfermedad es que permitiría al especialista, cuyas sospechas apuntan a la neurocisticercosis, solicitar una prueba serológica como sustituto o como un paso previo a la tomografía computarizada, a la radiografía o, en el caso de que se realice sólo con muestras de sangre, a la extracción de LCR. Al substituir los dos primeros métodos, el inmunodiagnóstico representa una ventaja económica, mientras que al substituir la extracción de LCR implica ventajas adicionales: no es tan doloroso, peligroso y es más fácil de realizar.

### III.2. PROBLEMA DE SALUD PÚBLICA

La cisticercosis es la enfermedad parasitaria más frecuente del sistema nervioso central. De las autopsias realizadas en hospitales públicos de México se detecta la presencia de cisticercos en un 2.5% de los casos, de los cuales, sólo la mitad han tenido manifestaciones clínicas. Con estas cifras se puede estimar que en nuestro país existen aproximadamente 2 millones de portadores. Por lo tanto estamos hablando de un serio problema de salud en México y en otros países en vías de desarrollo (Robles, 1997).

No existen estudios en el ámbito mundial acerca de la preponderancia exacta de la neurocisticercosis debido a que es muy difícil de determinar en vista de la poca especificidad de sus manifestaciones clínicas y de la falta de una prueba completamente confiable y segura que pueda ser utilizada en estudios epidemiológicos a gran escala.

En México esta enfermedad prevalece en la zona geográfica denominada “El Bajío”, lugar donde extensas plantaciones de frutas y vegetales alternan con grandes ranchos de ganado porcino.

Por otra parte, diversos estudios revelan que la neurocisticercosis es causa importante de admisiones hospitalarias y de procedimientos neuroquirúrgicos en México. La cisticercosis también es endémica en Sudamérica, principalmente en Brasil, Colombia, Ecuador y Perú. En dichos países, la neurocisticercosis es causa importante de epilepsia de inicio tardío. Al igual que en Asia y Africa, la endemia de la taeniosis/cisticercosis en América Latina se debe a las pobres condiciones socioeconómicas de la mayoría de sus habitantes, así como al desconocimiento de la naturaleza de esta enfermedad y de su forma de adquisición (Del Brutto, 1998).

### III.3. MÉTODOS DE DIAGNÓSTICO

El diagnóstico de la neurocisticercosis sigue siendo un reto. Mientras las modernas técnicas inmunológicas y de estudios de neuroimagen permiten la detección de un número de casos cada vez mayor, muchas veces estas técnicas no se encuentran disponibles en las áreas endémicas donde son realmente necesarias.

Además, el hecho de que muchos pacientes permanecen asintomáticos, aun en los casos de infección severa, provoca confusión cuando se desea utilizar solamente el criterio clínico para diagnóstico.

Se requieren criterios de diagnóstico más exactos para determinar la existencia de la enfermedad en diferentes poblaciones, para comparar exámenes epidemiológicos, para evaluar nuevas técnicas terapéuticas y para eliminar los diagnósticos erróneos en pacientes hospitalizados (Del Brutto *et al.*, 1996).

En esta sección se presenta una visión general de los principales métodos de diagnóstico existentes y sus limitaciones.

### **III.3.1. DIAGNÓSTICO CLÍNICO**

La sintomatología de la neurocisticercosis es muy compleja, ya que presenta una serie de combinaciones de síntomas que pueden describirse como caleidoscópicas y cuyo análisis cuidadoso permite a veces reconocer la localización del parásito en el tejido nervioso (Aluja *et al.*, 1987).

Los datos clínicos que pueden provocar la sospecha de neurocisticercosis son: vivir o haber vivido en zona endémica, antecedentes familiares o personales de teniasis, presencia de cefalea atípica, aparición de crisis convulsivas tardías (después de los 21 años), aparición de síntomas neurológicos asociados a hipertensión endocraneana y aparición de deterioro mental en un individuo no senil.

Es ampliamente recomendable auxiliarse, en lo posible, con otros métodos de diagnóstico que permitan confirmar o desechar la sospecha de la presencia de cisticercos en el sistema nervioso central.

### **III.3.2. DIAGNÓSTICO RADIOLÓGICO**

Los procedimientos radiológicos permiten una muy buena aproximación al diagnóstico de la neurocisticercosis, sin embargo ninguno de ellos es totalmente seguro. Hay lesiones que ninguno de estos procedimientos es capaz de detectar debido a la densidad de las lesiones o bien a los espacios entre cortes tomográficos.

#### **III.3.2.1. Radiografía Simple de Cráneo**

Muestra en ocasiones imágenes densas que sugieren parásitos calcificados cuando miden entre 3 y 6 milímetros de diámetro. Estas imágenes deben ser distinguidas de las calcificaciones fisiológicas y de otros tipos de calcificaciones patológicas. Se ha visto y



comprobado, por hallazgos de autopsia que no siempre se detectan parásitos en la etapa nodular calcificada utilizando este método.

### **III.3.2.2. Pneumoencefalografía, Ventriculografía y Angiografía**

Estos procedimientos han proporcionado ayuda al diagnóstico de la enfermedad con limitaciones para cada uno de los procedimientos, ya que sólo detectan la presencia de parásitos en forma indirecta, al mostrar desplazamientos ventriculares o vasculares.

### **III.3.2.3. Tomografía Computarizada**

Es el procedimiento que permite lograr el diagnóstico más preciso de la neurocisticercosis. Las imágenes que se obtienen y permiten el diagnóstico o la sospecha de presencia de neurocisticercosis, incluyen: imágenes “quísticas” hipodensas, imágenes moderadamente hiperdensas que se refuerzan intensamente con el contraste, imágenes nodulares hiperdensas, hiperdensidad difusa y dilatación ventricular.

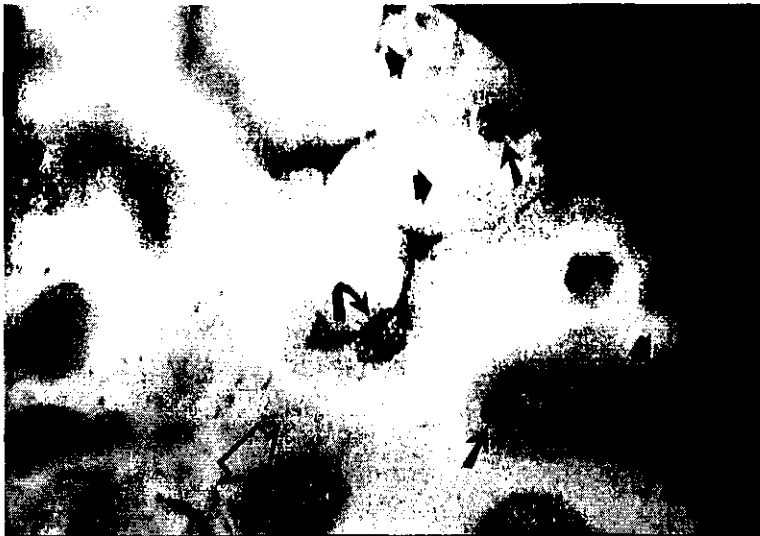


Figura Error! Unknown switch argument.:

#### **III.3.2.4. Resonancia Magnética**

Este método de diagnóstico por imágenes es un procedimiento no invasivo de bastante exactitud. Permite obtener imágenes que hasta hace pocos años no se soñaban en el diagnóstico neurológico, desafortunadamente es caro (Robles, 1997).

#### **III.3.3. EXÁMENES DE LABORATORIO**

El estudio de líquido cefalorraquídeo (LCR) apoya el diagnóstico de neurocisticercosis, sin embargo no es una prueba concluyente, ya que ciertas características presentadas por la enfermedad pueden aparecer en otros tipos de meningitis crónicas y, en el caso de no presentar las características buscadas, tampoco excluyen el diagnóstico de la neurocisticercosis.

#### **III.3.4. INMUNODIAGNÓSTICO**

Para la búsqueda de anticuerpos contra el cisticerco se han empleado varios métodos inmunológicos, tales como la técnica de fijación de complemento, la inmunoelectroforesis, la hemaglutinación pasiva, la inmunofluorescencia, la doble inmunodifusión, el ensayo inmunoenzimático y la prueba Western Blot.

Cada método presenta cierto porcentaje de sensibilidad y de especificidad en diferentes experimentos ya sean de laboratorio, pruebas controladas o en población abierta. El porcentaje puede variar sensiblemente de acuerdo al tipo de experimento, al antígeno utilizado y a la muestra del paciente.

Se entiende por *sensibilidad* a la capacidad de algún método para detectar la presencia del parásito y por *especificidad* a la capacidad del método para no confundir el diagnóstico con otra posible patología.

Los resultados indican que cualquiera de las técnicas descritas, cuando se realiza correctamente, es un importante apoyo para el diagnóstico de la neurocisticercosis, especialmente cuando la tomografía computarizada o los estudios de gabinete no están al alcance, no son concluyentes o son inaplicables. Tal es el caso de un estudio epidemiológico.

Cada uno de estos métodos presenta problemas técnicos para su realización, esto puede limitar el diagnóstico ya que dependerá de la experiencia del laboratorista. Además, se sabe que en un grupo de cisticercosis confirmada, la positividad no es del 100%, y esto se puede deber a que existen pacientes que no producen respuesta inmune contra el cisticerco o bien que es tan baja que no es posible la detección. Otra causa puede deberse a que la prueba se realiza en diferentes periodos de la relación huésped-parásito (Aluja *et al.*, 1987).

Particularmente para el método de ELISA se ha mostrado que alcanza un 85% de efectividad en el diagnóstico cuando se utiliza suero como muestra, un 95% cuando se utiliza LCR y un 100% si se combinan los resultados de las muestras de suero y de LCR (Plancarte *et al.*, 1989).

En contraste, otros estudios han concluido que ELISA ha fracasado en el diagnóstico de la neurocisticercosis al utilizar suero con un 50% de sensibilidad y un 70% de especificidad, y se argumenta que el fracaso se debe a que la infección por cisticercos es frecuente en la población general de zonas endémicas y a que la cisticercosis en el sistema nervioso produce reacción con inflamación local sin niveles de detección de anticuerpos específicos en suero (Sotelo, 1989).

El hecho de lograr un inmunodiagnóstico por vías serológicas (sin utilizar LCR) representa muchas ventajas. La extracción de muestras de sangre es más simple, menos dolorosa y

peligrosa para el paciente que la extracción de LCR. Adicionalmente presenta el atractivo de que pueden ser utilizadas en población abierta para estudios epidemiológicos. Esto significaría, pruebas de menor costo, mayor seguridad y mayor factibilidad de operación.

Una alternativa para auxiliar el diagnóstico, que cumpla con las características citadas, la constituye la prueba denominada Western Blot (WB). Dicha prueba permite visualizar anticuerpos dirigidos contra cada proteína viral (University of Arizona, 1998).

### III.4. IMÁGENES WESTERN BLOT

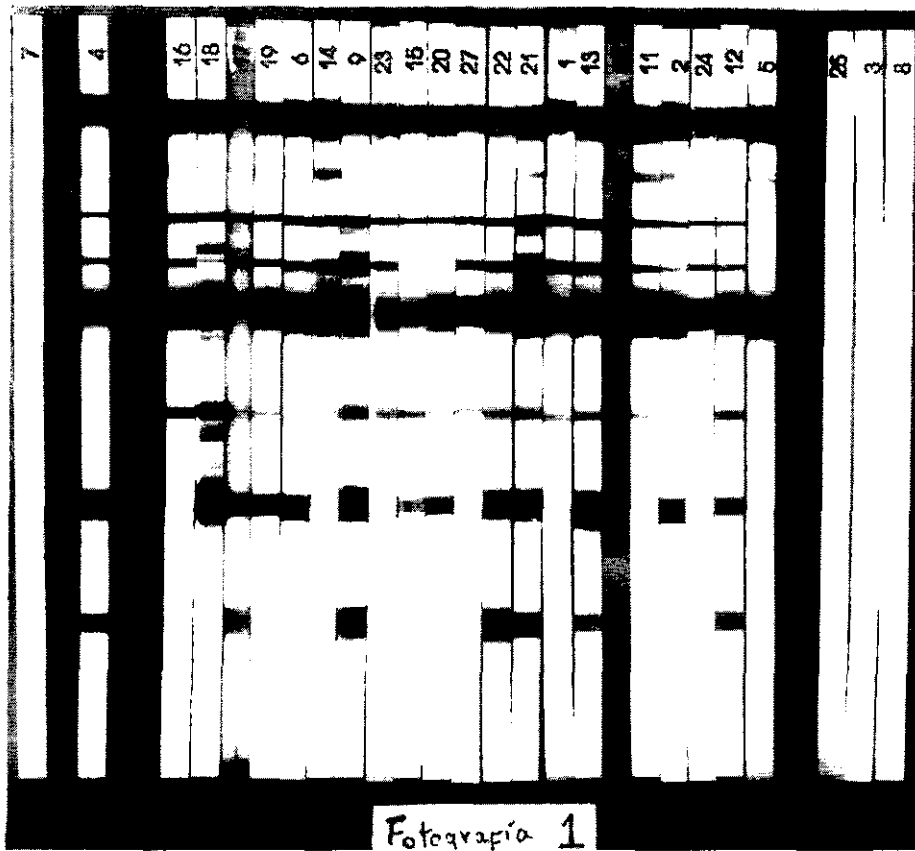
El método inmunológico de diagnóstico conocido como Western Blot (WB) ha sido utilizado con éxito para detectar los anticuerpos que genera el sistema inmunológico contra el Virus de Inmunodeficiencia Humana (VIH) y también ha sido utilizado como auxiliar en el estudio de otras enfermedades como la tuberculosis, la neurocisticercosis y la amibiasis (Bribiesca *et al.*, 2000).

En esta sección se realiza una descripción del método WB, se analiza la digitalización de las tiras y su razón de ser, se explica como se interpretan las imágenes resultantes, así como el tratamiento previo y análisis que se ha logrado con trabajos anteriores.

#### III.4.1. DESCRIPCIÓN DEL MÉTODO

La prueba WB, también conocida como *immunoblot*, detecta anticuerpos en el suero de un paciente, estos anticuerpos reaccionan contra algunas proteínas específicas de una enfermedad en particular. La prueba se realiza separando por peso molecular todas las proteínas que se tienen en un extracto crudo mediante un gel elaborado con un polímero de poros de tamaño controlado. Las proteínas se colocan verticalmente en una película de gel, se agrega el suero y entonces los anticuerpos existentes buscarán unirse a algunos antígenos en particular. En esta etapa de la prueba se incorpora una enzima de detección la cual se fija

sobre los anticuerpos. Finalmente se utiliza un químico como colorante cuando entra en contacto con las capas de proteína – anticuerpo - enzima (Koh, 1996).



**Figura Error! Unknown switch argument.: Ejemplo**

Una vez separadas las proteínas pueden ser transferidas a una membrana sólida para análisis WB, esto se logra aplicando una corriente eléctrica al gel, y posteriormente haciendo la transferencia en nitrocelulosa. El método WB proporciona gran ayuda en el análisis de mezclas de antígenos porque combina una fina resolución de la electroforesis en gel con una alta sensibilidad de anticuerpos (Larralde *et al.*, 1989).

La figura 9 muestra varias tiras después del proceso descrito, corresponden a una prueba WB para VIH, las tiras 7, 25, 3 y 8 pertenecen a individuos sanos y la tira 4 es de un individuo previamente diagnosticado como positivo (para control de la prueba). Como puede observarse, el patrón para diagnóstico se puede detectar a simple vista.

La prueba Western Blot evidentemente no es perfecta, pueden ocurrir errores en las lecturas de la prueba debido a múltiples razones. El proceso de análisis es realizado por un técnico de laboratorio el cual se encarga de observar la intensidad de las bandas presentes en las tiras resultantes, por tanto la prueba está limitada a la subjetividad del ojo humano y la luz que incide en el laboratorio puede jugar un papel importante en dicho análisis. Actualmente se utilizan densitómetros para auxiliar a los técnicos en la realización de la prueba.

El proceso que se utiliza para interpretar la información proporcionada por la prueba, en términos de la presencia de algunos antígenos es interesante. Sin embargo, debido a la subjetividad de la misma, no ha podido ser automatizado.

#### **III.4.2. INTERPRETACIÓN DE LA PRUEBA**

Las prueba Western Blot queda definida en una tira de nitrocelulosa manchada con diferentes tonalidades de gris, que pueden ir del blanco al negro.

Estas tiras se interpretan revisando si existe la presencia de algún antígeno o patrón de antígenos en la misma. Esto es, si en la tira se observa una concentración obscura en algún peso relacionado a una proteína en particular, que puede ser interpretada como una “banda”, esto significa que hay presencia de anticuerpos contra ese antígeno en particular y es un fuerte indicio de que el paciente puede estar infectado.

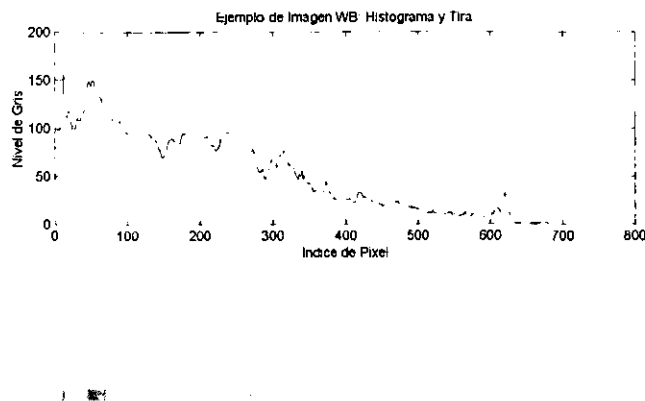
En el caso particular del VIH se ha encontrado que la prueba WB es muy sensible ya que se han detectado las bandas relacionadas con los antígenos cuya presencia es la confirmación de la enfermedad. El patrón de esta enfermedad en WB se puede detectar a simple vista (Larralde *et al.*, 1998). Sin embargo, para el caso de la neurocisticercosis no se ha logrado establecer un patrón que determine un diagnóstico preciso (Larralde *et al.*, 1989).

### III.4.3. DIGITALIZACIÓN DE LAS IMÁGENES

Las imágenes WB representan la versión digitalizada de la prueba WB. El análisis de las imágenes WB permite obtener más información acerca de la respuesta inmunológica del paciente (Bribiesca *et al.*, 2000).

La inspección visual de la prueba WB permite que se pierdan muchos detalles de la imagen, así como mucha información útil en el seguimiento fino de la enfermedad y para la investigación. El análisis digital de las imágenes permite una exploración más detallada de las características de las tiras.

Por otro lado, las tiras de nitrocelulosa que se obtienen como resultado de la prueba WB presentan bandas que con el tiempo y al contacto continuo con la luz, van perdiendo intensidad. Este problema queda resuelto con la digitalización, ya que la información de la imagen queda almacenada y no sufre cambio alguno (Larralde *et al.*, 1998).



**Figura** Error! Unknown switch argument.: **Imagen**

La digitalización de las imágenes se puede obtener de varias formas, existe el equipo AMBIS OIS, el cual permite obtener la digitalización y realizar algunos tipos de análisis.

También se puede tomar una fotografía de un conjunto de tiras Western Blot procurando que la colocación sea homogénea y posteriormente con algún software de tratamiento de imágenes (Photoshop, Gimp, etc.) separar las tiras. Una vez obtenidas las imágenes WB se puede aplicar sobre ellas diferentes técnicas de procesamiento digital de imágenes y de reconocimiento de patrones.

En principio, las imágenes WB muestran en un fondo blanco diferentes tonalidades de gris. Cuando se presenta una sucesión de niveles oscuros de gris, se dice que está presente una banda. Generalmente una imagen WB presenta varias bandas. La figura 10 presenta una imagen WB junto con su histograma que indica sus niveles de gris.

#### **III.4.4. TRATAMIENTO PREVIO Y ANÁLISIS**

Ya se han realizado algunos intentos para extraer más y mejor información de las imágenes digitales WB utilizando análisis numérico (Larralde *et al.*, 1998), redes neuronales (Lamiell *et al.*, 1993) y técnicas de reconocimiento de patrones (Bribiesca *et al.*, 2000).

En el caso del tratamiento con análisis numérico (Larralde *et al.*, 1998) se utilizó el digitalizador de imágenes AMBIS OIS para la obtención de las imágenes WB de pacientes infectados con VIH, se realizó un proceso de eliminación de fondo para observar con mayor claridad la ubicación de las bandas, se aplicó un ajuste de tamaño en las imágenes WB para forzarlas a quedar todas de un sólo tamaño, se establecieron varios criterios de diagnóstico, se realizó una revisión de imagen y un análisis cualitativo y cuantitativo de los resultados obtenidos.



Esta investigación concluyó que el diagnóstico automático del WB en la infección por VIH es posible. Asimismo demostró lo útil que puede ser la digitalización para organizar y analizar las imágenes WB sin perder las posibles correlaciones entre los antígenos.

La investigación en la que se utilizaron redes neuronales (Lamiell *et al.*, 1993) reportó muy buenos resultados en el análisis de imágenes WB de diferentes tipos de un virus que afecta a primates y eventualmente puede transmitirse a humanos denominado herpesvirus. Se logró que el sistema clasificara entre cuatro tipos diferentes de suero. La efectividad se reflejó en un porcentaje de diagnóstico positivo automatizado del 87% y un falso positivo de 2%, el cual superó el diagnóstico obtenido por técnicos humanos del 65% y 9% respectivamente

En el trabajo de reconocimiento de patrones (Bribiesca *et al.*, 2000) se utilizaron operadores de la técnica conocida como morfología matemática (Serra, 1982) para la detección de bandas en imágenes WB de infectados de neurocisticercosis. En el mencionado trabajo se concluyó que la técnica permite la extracción y análisis de bandas con facilidad y que elimina fuentes de variación de las imágenes WB (como “ruido” de fondo, falsa identificación de bandas e imprecisión en la cuantificación de cada banda).

### III.5. DEFINICIÓN DEL PROBLEMA

#### **III.5.1. ¿PORQUÉ ES DIFÍCIL DE DIAGNOSTICAR LA NEUROCISTICERCOSIS?**

Diversos factores provocan que la neurocisticercosis sea difícil de diagnosticar:

- Muchos pacientes infectados no presentan síntomas, incluso cuando la infección es severa.
- La sintomatología de la enfermedad es muy compleja ya que puede presentar muchas combinaciones y confundirse con otros padecimientos.

- A pesar de los avances en imagenología e inmunología, a la fecha no existe una prueba que sea totalmente confiable para el diagnóstico.
- No siempre hay disponibilidad de pruebas de tecnología sofisticada en las zonas consideradas endémicas.
- En casi cualquier estudio epidemiológico se presentan diagnósticos erróneos debido a la carencia de pruebas efectivas.
- Incluso en pacientes hospitalizados, sometidos a pruebas sofisticadas, puede haber confusión en el diagnóstico debido a la falta de especificidad de las imágenes neurológicas y al número de falsos positivos y falsos negativos que pueden ocurrir en las pruebas inmunológicas.

Actualmente existe un criterio de diagnóstico que auxilia a los médicos y trabajadores de la salud para evaluar los datos clínicos, radiológicos, inmunológicos y epidemiológicos del paciente para poder establecer un grado de certeza en el diagnóstico y así tomar las medidas pertinentes. El criterio utiliza cuatro grados de certeza: absoluta, mayor, menor y epidemiológica con base en los antecedentes del paciente y los resultados de sus estudios (Del Brutto *et al.*, 1996).

Concentrándonos en las pruebas inmunológicas, a pesar de sus deficiencias, han demostrado ser importantes auxiliares en el diagnóstico, sobretodo cuando se carece de pruebas radiológicas; tal es el caso de la prueba Western Blot, que ha generado una controversia en cuanto a su efectividad como herramienta de diagnóstico del padecimiento.

### **III.5.2. WESTERN BLOT COMO AUXILIAR EN EL DIAGNÓSTICO**

Se ha sugerido que la prueba Western Blot es determinante en el diagnóstico de la neurocisticercosis debido a que alcanza un 98% de sensibilidad y un 100% de especificidad (Tsang *et al.*, 1989). Sin embargo otros estudios revelan, que cuando la lesión es simple (un solo cisticerco), la sensibilidad disminuye drásticamente a un 28% (Wilson *et al.*, 1991).

En México se ha realizado el intento de encontrar una relación entre la prueba WB y la neurocisticercosis pero aun no se han logrado resultados satisfactorios para el diagnóstico fehaciente (Larralde *et al.*, 1989). Por otro lado, se ha sugerido que en pruebas no sesgadas el desempeño de los estudios serológicos puede ser considerablemente menor que en pruebas controladas (Ramos-Kuri *et al.*, 1992).

Los estudios demuestran que el análisis inmunológico de LCR presenta mejores resultados que el análisis de suero. Encontrar un método que mejore sustancialmente el diagnóstico utilizando solamente suero representa muchas ventajas, tales como: facilitar los estudios epidemiológicos, ser un estudio más económico, proveer seguridad al paciente (la extracción de LCR es un procedimiento peligroso y doloroso) y proveer al especialista con más elementos para el diagnóstico como paso previo a la solicitud de estudios radiológicos o bien como sustituto en caso de no existir la posibilidad de obtenerlos.

Uno de los problemas a resolver es determinar si existe una respuesta estándar del sistema inmunológico humano contra la presencia de cisticercos en el sistema nervioso central. De existir esa respuesta del sistema inmunológico, el problema sería detectar las enzimas que reaccionan contra las proteínas que componen al cisticerco. Concretamente en la prueba WB, esto significaría determinar la existencia de alguna banda (o patrón de bandas) en las imágenes WB que pudiera caracterizar a esta infección.

El problema de la detección de bandas en las imágenes WB ya ha quedado resuelto, en su mayor parte, con técnicas de morfología matemática (Bribiesca *et al.*, 2000). En esta tesis se propone aprovechar los resultados obtenidos en los trabajos anteriores, tanto en el área biomédica como en el área computacional, para intentar definir un patrón o un conjunto de patrones que puedan auxiliar a un especialista en el diagnóstico de la neurocisticercosis.

### III.5.3. PROPUESTAS DE SOLUCIÓN

En esta tesis se presentan tres propuestas de solución para encontrar el patrón que auxilie el diagnóstico de la neurocisticercosis utilizando técnicas de Computación Evolutiva. Además en la sección de recomendaciones se incluyen otras alternativas de solución obtenidas a partir de los antecedentes y resultados obtenidos. Las ideas básicas de las tres propuestas son:

1. Utilizar Algoritmos Genéticos para determinar una imagen WB modelo que se asemeje a los casos de los pacientes que presentaron neurocisticercosis confirmada. En este caso el modelo quedará definido como aquella tira que presente el mayor parecido con respecto de las tiras de pacientes enfermos. Como primera función objetivo se propone encontrar aquella tira que posea el mínimo error (absoluto) con respecto de las tiras de los pacientes confirmados. Una vez determinado esto, la tira modelo entra en un proceso de verificación contra todas las muestras que se tienen, buscando minimizar el error tanto de falso positivo como de falso negativo y buscando superar el porcentaje de sensibilidad obtenido por ELISA basado en suero (50%-85%).
2. Determinar por medio de Algoritmos Genéticos una imagen WB compuesta de dos partes. Una parte con las características de las imágenes de neurocisticercosis y otra parte con las características de las imágenes de otras enfermedades. Una vez obtenida esta imagen se tendrá un proceso de verificación con las muestras disponibles, para evaluar el método.
3. Utilizar Programación Genética para determinar un modelo eficiente de patrón para auxiliar el diagnóstico de la neurocisticercosis usando las imágenes Western Blot. En este caso el modelo será obtenido a través de regresión simbólica usando operadores matemáticos (+, -, \*, /) y de igualdad (<, >, ≤, ≥) para encontrar un programa que pueda describir el comportamiento de las imágenes en base al conjunto que pertenecen.

### **III.5.4. LIMITACIONES**

Las pruebas serológicas tienen limitaciones en el sentido de que pueden ser afectadas por diversos factores, tales como: la calidad de los reactivos, el tiempo de realización del proceso y la experiencia del laboratorista. También el proceso de digitalización puede presentar problemas, como la posición de las tiras WB al momento de ser fotografiadas o el proceso mismo de digitalización con densitómetro.

Para este trabajo se cuenta con un total de 201 imágenes WB, de las cuales 49 pertenecen a individuos con neurocisticercosis positiva y el resto a individuos con otros padecimientos neurológicos bien clasificados.

Este trabajo se ha realizado con material proporcionado por el Departamento de Inmunología del Instituto de Investigaciones Biomédicas (IIB) de la Universidad Nacional Autónoma de México.

Es importante señalar que los resultados obtenidos por esta investigación no necesariamente proporcionarán un patrón que pueda ser utilizado en cualquier región del mundo, debido a que la respuesta inmune puede variar dependiendo de si la zona es endémica o no. Sin embargo, el método podrá ser reproducido en un experimento similar, y obtener sus propios resultados, que aunque puedan ser diferentes, proporcionarán conclusiones para la región en que se realicen.

### **III.6. RESUMEN**

La neurocisticercosis humana es una enfermedad parasitaria del sistema nervioso central que representa un grave problema de salud pública en varios países en vías de desarrollo. Aún no existe un sólo método de diagnóstico que sea determinante por sí mismo, por lo

tanto, los médicos especialistas deben auxiliarse de su propia experiencia clínica y de varios métodos auxiliares para dar un diagnóstico adecuado.

Los métodos inmunológicos de diagnóstico son los que presentan mejores perspectivas para la realización de estudios epidemiológicos porque son más económicos y porque pueden ser realizados con muestras de suero de los pacientes. Desafortunadamente, aún no han proporcionado resultados satisfactorios.

La prueba inmunológica Western Blot ha despertado especial interés en la comunidad científica debido a su efectividad en el diagnóstico del SIDA, por lo que se ha intentado con ella el análisis de otras enfermedades incluyendo la neurocisticercosis.

Se propone el uso de la versión digital de la prueba Western Blot y de técnicas de Computación Evolutiva para encontrar un método que pueda ser la prueba concluyente de diagnóstico, o bien un poderoso auxiliar del especialista como un paso previo a otras pruebas. En los capítulos siguientes se presenta el desarrollo a detalle y los resultados obtenidos de las propuestas de solución.

### III.7. REFERENCIAS

Aluja, A.; Escobar, A.; Escobedo, F.; Flisser, A ; Laclette, J. P.; Larralde, C.; Madrazo, I.; Velázquez, V. y Willms, K. 1987. *Cisticercosis : una recopilación actualizada de los conocimientos básicos para el manejo y control de Taenia solium*. Biblioteca de la Salud. Instituto Nacional de Salud Pública y Fondo de Cultura Económica. DF, México.

Bribiesca, E.; Sánchez, H.; Govezensky, T.; Larralde, C. 2000. "Morfología matemática aplicada a imágenes Western Blot: el caso de la neurocisticercosis humana". MICAI, Acapulco, Mex. Pp. 147-156.

- Del Brutto, Oscar H. 1998. "Neurocisticercosis".  
(Documento Web).  
<http://www.scn.es/cursos/tropical/Cisticercosis.htm>  
25 de noviembre de 1999.
- Del Brutto, O.H.; Wadia, N.M.; Dumas, M.; Cruz, M.; Tsang, V.C.W. y Schantz, P.M.  
1996. "Proposal of diagnostic criteria for human cysticercosis and neurocysticercosis". *Journal of Neurological Sciences*. 142: 1-6.
- Flisser, Ana y Malagón, Filiberto. 1989. *Cisticercosis humana y porcina: su conocimiento e investigación en México*. Conacyt, Limusa-Noriega. DF, México.
- Koh, P. 1996. "Question: What is the Western Blot test".  
(Documento Web)  
<http://crystal.biol.csufresno.edu:8080/projects/155.html>  
27 de Marzo de 2000.
- Lamiell J.M, Ward J.A y Hilliard J.K. 1993. "Detection of Type-Specific herpesvirus Antibodies by Neural Network Classification of Western Blot Densitometer Scans". *IEEE International Conference on Neural Networks*. Vol. 3: 1731-1736.
- Larralde, C. ; Montoya, R.M. ; Sciotto E. ; Díaz M.L. ; Govezensky T. ; Coltorti, E. 1989. "Deciphering western blots of tapeworms antigens (Taenia Solium, Echinococcus Granulosis y Taenia Crassiceps) reacting with sera from neurocisticercosis and hydatid disease patients". *American Journal of Tropical Medicine Hygiene*. 1989. Vol 40 ; 3 ; 282-290.
- Larralde C., Paz E., Viveros M., Padilla A., Soler C. y Govezensky T. 1998. "Análisis numérico de imágenes digitalizadas de Western Blot: el caso del VIH". *Gaceta Médica de México*. 134: 385-398.

Plancarte A., Espinoza B. y Flisser A. 1989. "Inmunodiagnóstico de la cisticercosis humana por el ensayo inmunoenzimático". Publicado en Flisser y Malagón 1989 (ver referencia), pp 159-163.

Ramos-Kuri, M.; Montoya, R.M.; Padilla, A; Govezensky, T.; Díaz M.L.; Sciutto, E.; Sotelo, J. y Larralde, C. 1992. "Immunodiagnosis of Neurocysticercosis: Disappointing Performance of Serology (Enzyme-Linked Immunosorbent Assay) in an unbiased sample of neurological patients". *Archives of Neurology*. 49:633-636.

Robles, Clemente. 1997. *Tratamiento de la Neurocisticercosis*. Litoarte. DF, México.

Serra, J. 1982. *Image Analysis and Mathematical Morphology*, Volume 1. Academic Press Limited. London.

Sotelo, Julio. 1989. "Elisa en el diagnóstico de la neurocisticercosis". Publicado en Flisser y Malagón, 1989 (ver referencia), pp 165-167.

Tsang, V.C.W., Brand, J. y Boyer, E. 1989. "Enzyme linked immunoelectrotransferency blot assay and glycoprotein antigens for diagnosis human cysticercosis (*Taenia Solium*)". *The Journal of Infectious Diseases*. 159:50-59.

University of Arizona. "The Biology Project, Immunology, Western Blot Activity". 15 de enero de 1998. (Documento Web).

<http://www.biology.arizona.edu/immunology/activities/western-blot/west1.html>

25 de noviembre de 1999.

Wilson M., et al. 1991. "Clinical Evaluation of the Cysticercosis Enzyme-Linked Immunoelectrotransfer Blot in Patients with Neurocysticercosis". *The Journal of Infectious Diseases*. 164:1007:1009.



## IV. ALGORITMOS GENÉTICOS APLICADOS A IMÁGENES WESTERN BLOT: EL CASO DE LA NEUROCISTICERCOSIS

### IV.1. INTRODUCCIÓN

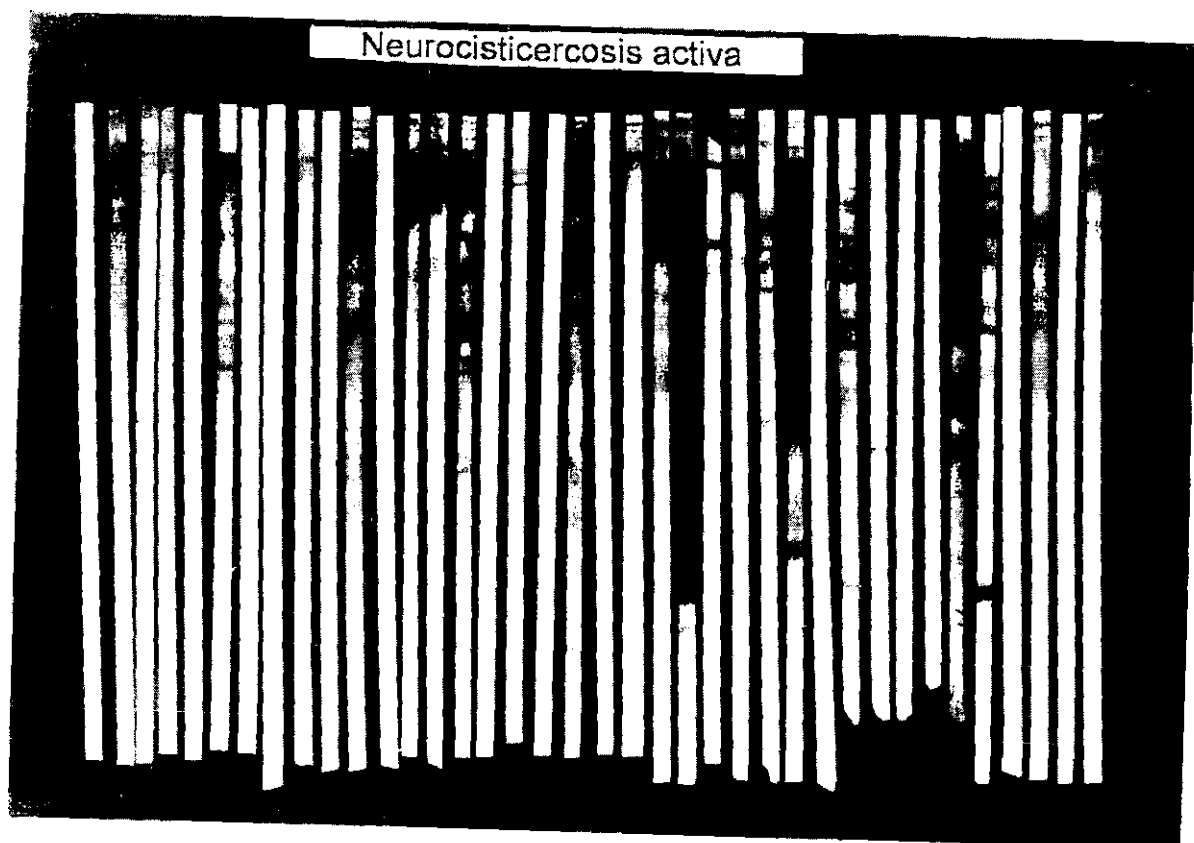
En la búsqueda de un método auxiliar para el diagnóstico de la neurocisticercosis, se propone en este capítulo la utilización de Algoritmos Genéticos para definir una imagen Western Blot modelo para los casos de pacientes infectados.

Como se describió en el capítulo anterior, en las pruebas Western Blot se observa la reacción del sistema inmunológico de los pacientes expresada por anticuerpos contra componentes moleculares del parásito (Larralde *et al.*, 1989). En el caso ideal todas las pruebas de los pacientes con neurocisticercosis deberían presentar un cierto comportamiento que permitiera distinguirlos a simple vista de las pruebas de los pacientes que no están infectados. Sin embargo, en la realidad no ocurre así. Algunas de las pruebas del conjunto de pacientes infectados presentan características comunes, pero algunas otras son muy diferentes. El problema radica en detectar contra qué componentes moleculares del parásito reacciona “*normalmente*” el sistema inmunológico humano. Esto significa determinar en términos de la prueba Western Blot, cuáles son las bandas que pueden indicar la presencia del parásito.

Para generar una imagen modelo que exprese el comportamiento del sistema inmunológico ante la presencia del parásito, se sugiere la utilización de imágenes Western Blot y Algoritmos Genéticos. En esta primera aproximación se intenta determinar si es suficiente una imagen modelo para auxiliar el diagnóstico de la neurocisticercosis.

## IV.2. DESCRIPCIÓN

En esta aproximación se utilizan 201 imágenes Western Blot suministradas por el Instituto de Investigaciones Biomédicas, de las cuales, 49 pertenecen a individuos con neurocisticercosis y el resto a pacientes con otras enfermedades. Del primer grupo, 38 pertenecen a pacientes con neurocisticercosis activa (quistes viables) y 11 de pacientes con neurocisticercosis inactiva (forma cicatrizal, granulomas o calcificaciones). Esta clasificación está en función de la duración de la infestación y de factores inmunológicos del huésped (Sotelo, 1996).



**Figura Error! Unknown switch argument.: Pruebas Western Blot**

El conjunto de otras enfermedades incluye a pacientes con epilepsia, alteración psiquiátrica, parálisis facial, neoplasia cerebral, alteración vascular cerebral, neuroinfección, traumatismo craneoencefálico, mal de Parkinson y otras enfermedades neurológicas no especificadas.

La figura 11 muestra pruebas Western Blot con neurocisticercosis activa, mientras que la figura 12 muestra pruebas WB para casos de epilepsia.

Es importante distinguir las imágenes del conjunto de neurocisticercosis del conjunto de imágenes de otras enfermedades porque en la práctica es posible confundir el diagnóstico de acuerdo a los síntomas presentados. Si se logra definir un modelo que permita distinguir entre neurocisticercosis y estas otras enfermedades, se estará brindando al especialista una poderosa herramienta para auxiliar el diagnóstico.

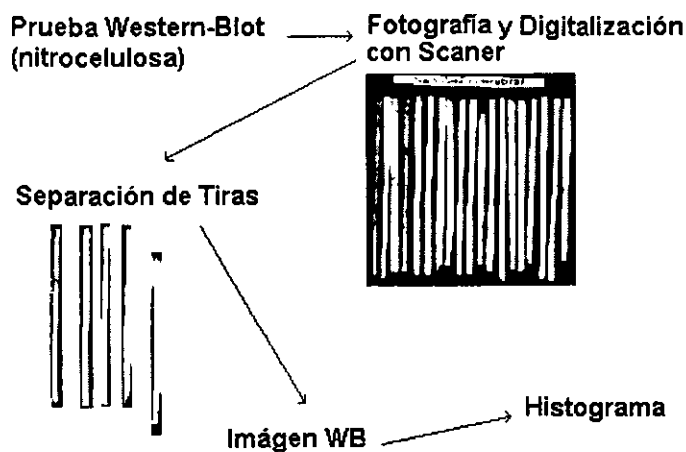


La imagen digital de las fotografías se obtuvo utilizando un *scanner* convencional. La separación de las tiras se realizó utilizando el software comercial Corel Draw que genera imágenes digitales en formato PCX.

Para obtener los valores numéricos en forma de histograma a partir de las imágenes se utilizó un programa de Análisis de Imágenes Western Blot (Sánchez, 1998) con el que se hace la lectura de las imágenes, el proceso de normalización en intensidad y de desmanchado, y una conversión a otra representación.

Los histogramas quedaron representados en archivos planos (ASCII) con un valor numérico entero inicial  $n$  que representa el tamaño del histograma en pixels y  $n$  valores enteros numéricos entre 0 y 255 que representan a los histogramas en sí. La figura 13 describe esquemáticamente el proceso.

Asimismo, el apéndice A contiene el programa 1, con el que se pueden obtener los histogramas y las imágenes WB, así como los valores de ancho y largo de las imágenes. El apéndice no incluye el proceso de normalización ni de desmanchado.



**Figura Error! Unknown switch argument.:**

El programa 2 del apéndice A hace la conversión de los archivos ASCII a una matriz en la que se representan los 201 histogramas con valores enteros entre 0 y 255 y a un vector que representa los tamaños de los 201 histogramas. Los tamaños de los histogramas varían entre 620 y 768 pixels y tienen una media de 695 pixels.

Una vez con los histogramas necesarios para iniciar el proceso se realizó el diseño del experimento. Para los experimentos realizados se utilizó el Toolbox de Algoritmos Genéticos (Chipperfield *et al.*, 1994) desarrollado en la Universidad de Sheffield, Inglaterra.

Con el objeto de poder realizar varias pruebas en forma sistemática, se creó un programa que hiciera la lectura de parámetros de los Algoritmos Genéticos desde un archivo externo, ejecutara cada algoritmo y generara archivos externos de resultados. Con este propósito se hicieron modificaciones al código fuente original del programa del Toolbox para algoritmos genéticos que utiliza poblaciones. Este programa se denomina “*metawbga*”.

#### IV.3. APROXIMACIÓN GENÉTICA (FORMULACIÓN)

De acuerdo al objetivo principal planteado en esta tesis, se busca una imagen WB que sirva como modelo para auxiliar el diagnóstico de la neurocisticercosis por lo que se debe encontrar, en el caso ideal, una imagen que sea muy parecida a las imágenes WB de los pacientes con neurocisticercosis y muy diferente a las imágenes WB de los pacientes con otras enfermedades. Con este propósito se utiliza el conjunto de histogramas de pacientes con neurocisticercosis tratando de encontrar un histograma que sea lo más parecido a dicho conjunto utilizando Algoritmos Genéticos.

Después de encontrar el histograma modelo se realiza un proceso de evaluación en el que se calculan las diferencias entre el histograma modelo y los 201 histogramas, para determinar la efectividad y la calidad de los resultados. En el caso ideal, la diferencia debe ser menor en el conjunto de neurocisticercosis.

Si se considera a cada valor de los histogramas (pixel) como una variable, se tiene un problema con 768 variables. Por lo anterior, y tomando en cuenta que el tiempo de procesamiento está en función del número de variables del problema, se realizaron pruebas a diferentes intervalos de muestreo de los histogramas para determinar la sensibilidad del muestreo en los resultados.

Se realizaron diez pruebas para cada tipo de recombinación (discreta, intermedia y lineal). Las diez pruebas corresponden a diferentes intervalos de muestreo entre 1 y 10, es decir para cada tipo de recombinación se obtienen histogramas de diferentes tamaños (77, 86, 96, 110, 128, 154, 192, 256, 384 y 767) dependiendo del número de muestras. De esta forma se trata de llevar a cabo un equilibrio entre la calidad de los resultados y el tiempo de cómputo utilizado.

Tomando en cuenta lo anterior se definió la siguiente función objetivo:

$$f(x) = \sum_{i=1}^M \sum_{j=1}^N X_i - Y_{ij} ,$$

donde:

- X es una imagen WB (histograma) con valores reales entre 0 y 1,
- Y es una imagen WB (histograma) perteneciente al conjunto de neurocisticercosis,
- M es el número de genes (pixels),
- N es el número de imágenes WB pertenecientes al conjunto de neurocisticercosis,
- $X_i$  representa el pixel  $i$  del histograma X,
- $Y_{ij}$  representa al pixel  $i$  del histograma  $j$  del conjunto de neurocisticercosis.

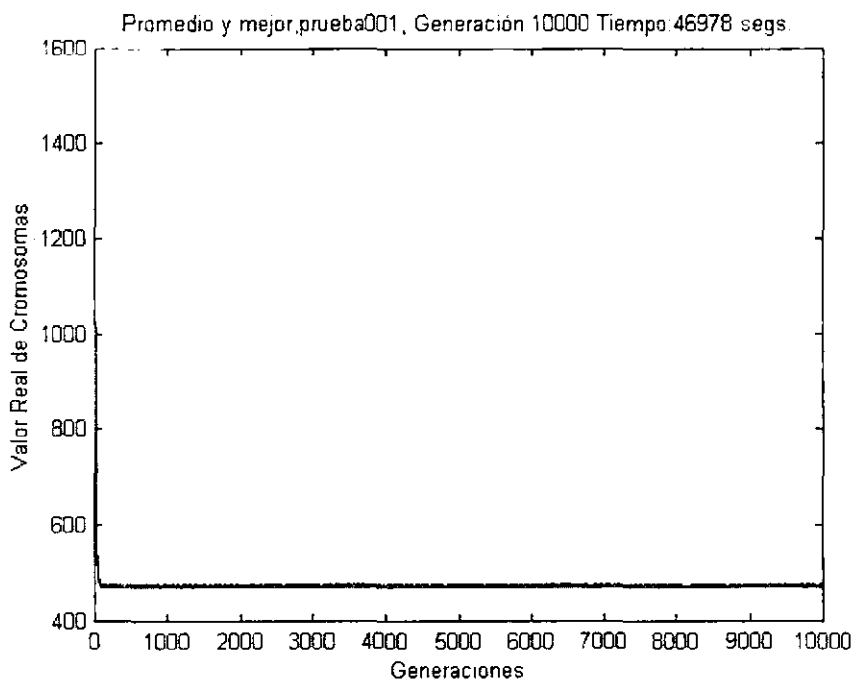
Para la evaluación de la función se utilizó una aptitud normalizada (definida en la sección II.3.3) para obtener valores entre 0 y 1 con  $r_{\max} = \text{número de muestras} * \text{número de individuos del conjunto de neurocisticercosis}$ . Esto representa que una imagen podrá obtener el valor de 1 solamente en el caso de que sea igual a todas las imágenes del conjunto de neurocisticercosis.

Para evitar que el tiempo de cómputo se hiciera excesivo, sólo en las primeras tres pruebas se utilizaron 10000 generaciones como máximo. Con base en los resultados, pruebas posteriores utilizaron sólo 2000 generaciones como máximo.

Del mismo modo se efectuaron pruebas del comportamiento del algoritmo de acuerdo a los parámetros utilizados, por lo que en base a ese estudio se utilizaron cuatro sub-poblaciones, una probabilidad de cruzamiento de 1.0, una probabilidad de mutación de 0.45, una tasa de migración de 0.2 cada 20 generaciones, un método de selección universal estocástico (*stochastic universal selection*) y 250 individuos por sub-población.

#### IV.4. PRUEBAS Y RESULTADOS

Para llevar a cabo estas pruebas se utilizó una computadora personal con procesador Pentium III, a 550 MHZ, con 128 MB de memoria RAM y 16 GB de memoria secundaria.



**Figura Error! Unknown switch argument.: Prueba 1: Valores mejor**

En la figura 14 se muestra el proceso evolutivo del Algoritmo Genético de la primera prueba. Como se puede observar, no hay una mejoría en la evolución, por lo que se redujo el máximo número de generaciones de 10000 a 2000 a partir de la cuarta prueba. Las pruebas dos y tres presentan un comportamiento similar a la prueba uno.

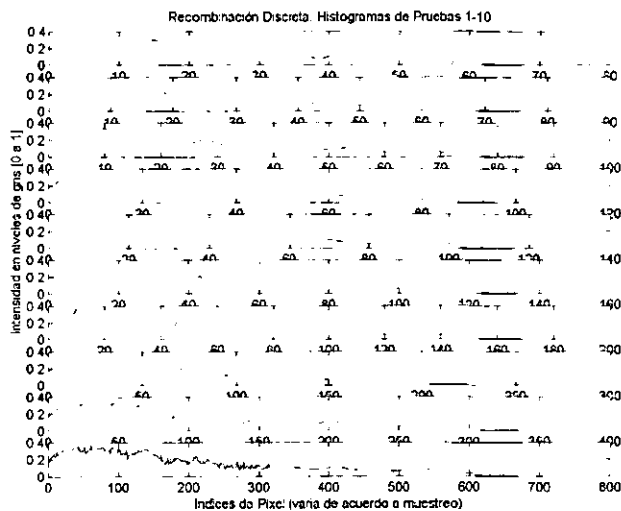
*Tabla 4.1. Resultados de pruebas 1 a 30*

Prueba	Número var.	Tiempo (segs.)	Máxima Generación	Mejor Individuo F(X)	Mejor Aptitud
001	77	46978	10000	4.701890e+002	0.8754
002	86	37285	10000	5.217048e+002	0.8762
003	96	39402	10000	5.867087e+002	0.8753
004	110	8166	2000	6.710828e+002	0.8755
005	128	8499	2000	7.813191e+002	0.8754
006	154	9595	2000	9.376822e+002	0.8757
007	192	10746	2000	1.170669e+003	0.8756
008	256	12327	2000	1.559471e+003	0.8757
009	384	16672	2000	2.339136e+003	0.8757
010	767	35782	2000	4.680902e+003	0.8755
011	77	7427	2000	4.701905e+002	0.8754
012	86	7634	2000	5.217097e+002	0.8762
013	96	7947	2000	5.867203e+002	0.8753
014	110	8212	2000	6.711750e+002	0.8755
015	128	8590	2000	7.814897e+002	0.8754
016	154	9445	2000	9.384617e+002	0.8756
017	192	10655	2000	1.172083e+003	0.8754
018	256	12735	2000	1.564407e+003	0.8753
019	384	16927	2000	2.365216e+003	0.8743
020	767	35826	2000	4.907558e+003	0.8694
021	77	7455	2000	4.709284e+002	0.8752
022	86	7673	2000	5.227073e+002	0.8760
023	96	7914	2000	5.880499e+002	0.8750
024	110	8141	2000	6.730208e+002	0.8751
025	128	8526	2000	7.848580e+002	0.8749
026	154	9357	2000	9.424553e+002	0.8751
027	192	10666	2000	1.180992e+003	0.8745
028	256	12611	2000	1.584103e+003	0.8737
029	384	16785	2000	2.416582e+003	0.8716
030	767	35333	2000	5.169275e+003	0.8625

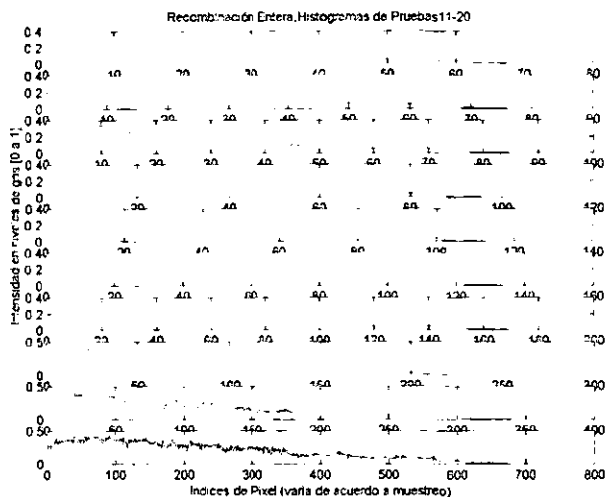
Las pruebas 1 a 10 corresponden a la recombinación discreta, de la 11 a la 20 a la recombinación intermedia y de la 21 a la 30 a la recombinación lineal. En la tabla 4.1 se puede observar que el tiempo de procesamiento se incrementa de acuerdo al número de variables y al máximo número de generaciones en forma proporcional. También se puede observar que los valores de la función objetivo aplicada al mejor individuo se incrementan de acuerdo al número de variables, mientras que el valor de aptitud permanece prácticamente constante.



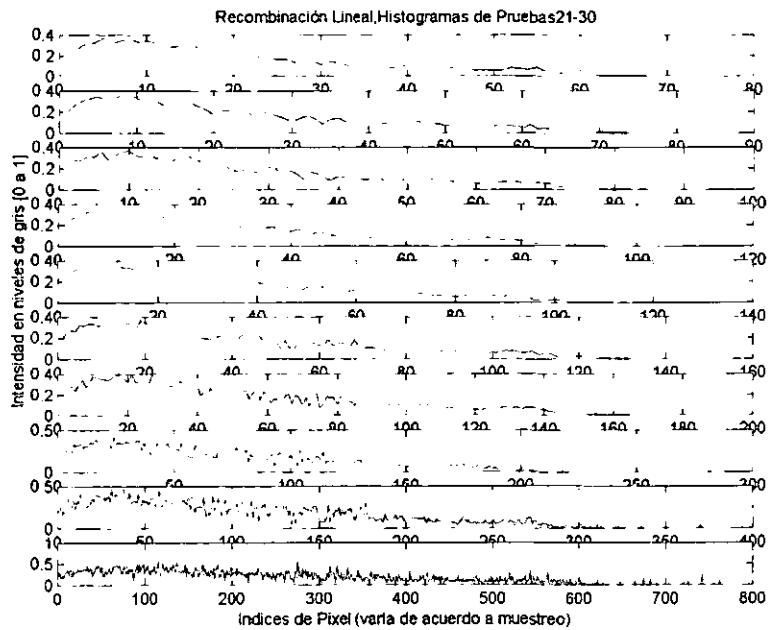
En las figuras 15, 16 y 17 se pueden observar los histogramas obtenidos en las diferentes pruebas para recombinación discreta, intermedia y lineal respectivamente. En las figuras 18, 19 y 20 se observa lo propio para las imágenes Western Blot obtenidas.



**Figura Error! Unknown switch argument.: Histogramas de**



**Figura Error! Unknown switch argument.: Histogramas con pruebas**



**Figura Error! Unknown switch argument.: Histogramas de**

Recombinación Discreta, Tiras de Pruebas 1-10

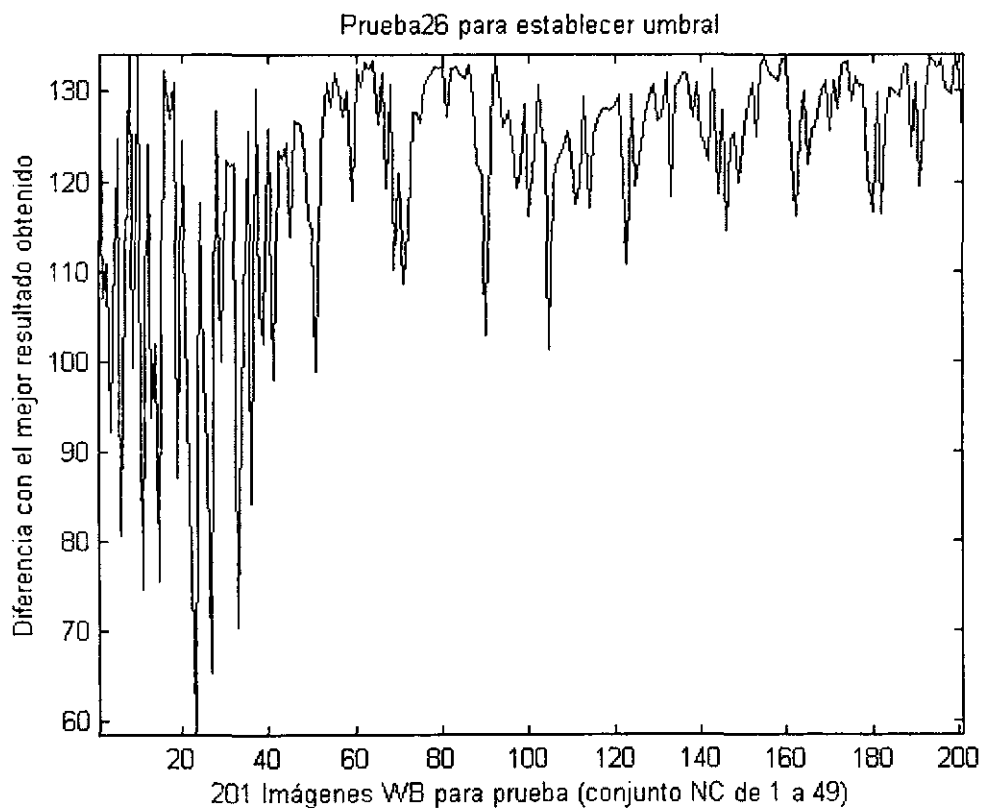
**Figura Error! Unknown switch argument.: Imágenes WB de**

**Figura Error! Unknown switch argument.: Histogramas de pruebas**

**Figura Error! Unknown switch argument.: Histogramas de**

Las figuras 15 a 20 proporcionan información acerca de las bandas que aparecen más frecuentemente en las imágenes Western Blot pertenecientes al conjunto de pacientes con neurocisticercosis, y pueden apoyar a los especialistas a determinar los componentes moleculares del cisticerco.

Una vez generadas las imágenes modelo con mejor desempeño que se muestran en las figuras 18-20, se llevó a cabo la evaluación de las mismas comparándolas con las imágenes disponibles. Dicha evaluación consiste en el cálculo de la suma de la diferencia absoluta de la imagen modelo con cada una de las 201 imágenes que se tienen.

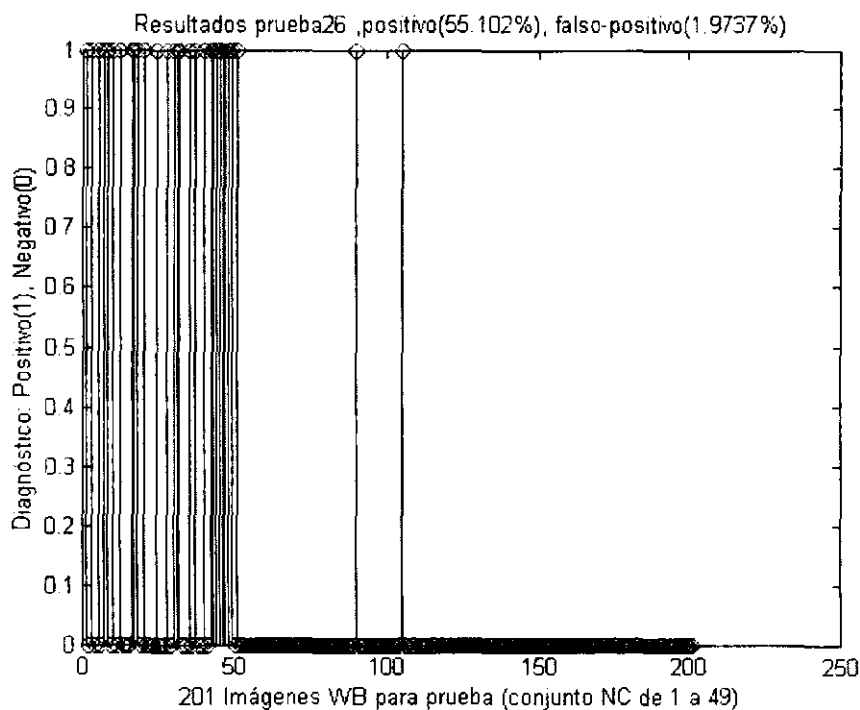


**Figura Error! Unknown switch argument.:  
Evaluación para obtener umbral**

Esta evaluación reveló un comportamiento semejante para todas las pruebas, el cual se muestra en la figura 21. Es claro que los primeros 49 histogramas presentan una menor diferencia que el resto de las pruebas.

Sin embargo, es necesario definir un criterio de decisión para apoyar el diagnóstico. Es decir, se requiere encontrar un valor *umbral*, tal que, si la diferencia es mayor que el valor de umbral, el diagnóstico es negativo; y si la diferencia es menor o igual al valor de umbral el diagnóstico es positivo.

Para la obtención de este valor umbral, se probaron 100 valores numéricos diferentes distribuidos uniformemente entre la menor y la mayor diferencia de la evaluación en cada una de las pruebas realizadas. El mejor resultado obtenido en todas las pruebas se muestra en la figura 22.



**Figura** Error! Unknown switch argument.:

En la figura 22 puede observarse que el porcentaje de falsos positivos es muy bajo (1.97%), no así el número de falsos negativos (44.9%). En términos de investigación biomédica, con este método estamos obteniendo una sensibilidad del 55.1% y una especificidad del 98.03%. Esta información revela que hay cierta similitud entre varias imágenes Western Blot pertenecientes al conjunto de neurocisticercosis, pero esta similitud no es general en todos los casos, lo que sugiere explorar alternativas en la definición de la aproximación genética.

## IV.5. CONCLUSIONES

El diagnóstico de la neurocisticercosis sigue siendo un reto para los médicos especialistas. Generalmente se basa en su experiencia y en algunos métodos auxiliares. Un método que ha causado especial interés en la comunidad científica es la prueba Western Blot y su versión digital, las imágenes Western Blot.

En este capítulo se ha presentado un método auxiliar de diagnóstico basado en imágenes Western Blot y Algoritmos Genéticos. La idea es encontrar una imagen Western Blot que pueda ser utilizada como patrón que permita determinar si un paciente está infectado por el parásito.

Los resultados obtenidos en esta primera aproximación, muestran que con este método se puede generar una imagen Western Blot modelo que alcanza una sensibilidad del 55.1% y una especificidad del 98.03% en el diagnóstico.

Es importante resaltar que en estos experimentos no se observó una diferencia significativa en la calidad de los resultados, a pesar de haber utilizado diferentes intervalos de muestreo. Esto significa que se puede reducir el número de variables hasta establecer un equilibrio entre la calidad de los resultados visuales (definición clara de bandas específicas) y el tiempo de procesamiento.

Aunque los resultados obtenidos en este capítulo no mejoran los de otras pruebas (ELISA por ejemplo), muestran que es posible encontrar un patrón auxiliar de diagnóstico, a pesar de que no todos los pacientes infectados presentan la misma reacción inmunológica al parásito. Es evidente que se requiere una redefinición del problema que tome en cuenta elementos adicionales. El capítulo siguiente desarrolla esta idea.

#### IV.6. REFERENCIAS

Chipperfield A., Fleming P., Pohlheim H., Fonseca C. 1994. *Genetic Algorithm Toolbox for use with Matlab. User's Guide*. Department of Automatic Control and Systems Engineering, University of Sheffield. UK.

Larralde, C. ; Montoya, R.M. ; Sciotto E. ; Díaz M.L. ; Govezensky T. ; Coltorti, E. 1989. "Deciphering western blots of tapeworms antigens (Taenia Solium, Echinococcus Granulosis y Taenia Crassiceps) reacting with sera from neurocisticercosis and hydatid disease patients". *American Journal of Tropical Medicine Hygiene*. 1989. Vol 40 ; 3 ; 282-290.

Sánchez, H. 1998. Desarrollo de un Sistema de Análisis de Imágenes Western Blot. Programa de computadora. Instituto de Investigación en Matemáticas Aplicadas y en Sistemas e Instituto de Investigaciones Biomédicas. UNAM. México, DF, México.

Sotelo, J. 1996. "Clínica y diagnóstico de la cisticercosis humana". *Encontro do cone Sul sobre Teniase e Cisticercose*. Curitiba. Organizacion pan-americana de saude.

## V. ALGORITMOS GENÉTICOS APLICADOS A IMÁGENES WESTERN BLOT: DECISIÓN ENTRE NEUROCISTICERCOSIS Y OTRAS ENFERMEDADES

### V.1. INTRODUCCIÓN

En el capítulo IV se desarrolló un procedimiento basado en Algoritmos Genéticos para encontrar imágenes patrón Western Blot que puedan auxiliar en el diagnóstico de la neurocisticercosis. El procedimiento se basa en generar histogramas que incluyan todas las características de las imágenes pertenecientes al conjunto de individuos infectados.

Aunque las imágenes obtenidas en esta primera aproximación proporcionan cierta información valiosa acerca de las zonas donde se concentran “bandas”, dicha información no ha sido suficiente para dar un diagnóstico fehaciente de los pacientes que están infectados y de los que no.

Por este motivo este capítulo propone una alternativa basada en encontrar pares de imágenes Western Blot utilizando Algoritmos Genéticos. De cada par encontrado, una imagen debe poseer características del conjunto de Neurocisticercosis (NC) y otra del conjunto de pacientes que padecen otras enfermedades (OE). De este modo, para cada par se realiza una comparación de semejanza con las imágenes Western Blot ya clasificadas.

En el caso ideal, las imágenes pertenecientes al conjunto de neurocisticercosis deben ser más parecidas a la imagen NC que a la imagen OE y las imágenes pertenecientes al conjunto de otras enfermedades deben ser más parecidas a la imagen OE que a la imagen NC.

Es claro que los pares obtenidos no van a reflejar por si mismos un patrón de diagnóstico, pero si deben dar elementos para tomar una decisión con base en los datos que se tienen en las imágenes.



## V.2. DESCRIPCIÓN

Para este experimento se utilizaron las mismas 201 imágenes y sus correspondientes histogramas. A diferencia de la primera aproximación, se buscan pares de histogramas. Con objeto de representar un par de histogramas de tamaño  $n$  se utiliza un solo histograma de tamaño  $2n$ , donde  $1$  hasta  $n$  representa al primer histograma (NC) y  $n+1$  hasta  $2n$  representa al segundo histograma (OE).

El hecho de duplicar el tamaño de los histogramas implica que el número de variables se duplica y por tanto el tiempo de procesamiento.

En esta prueba se utiliza el mismo programa de lectura de parámetros y ejecución de los algoritmos genéticos del capítulo anterior (programa 3 del apéndice A), con las adecuaciones necesarias en los parámetros para procesar un mayor número de variables y la obtención de dos histogramas representados en uno sólo.

## V.3. APROXIMACIÓN GENÉTICA (FORMULACIÓN)

En este caso se está buscando una imagen WB compuesta de dos partes. La primera parte con características de neurocisticercosis y la segunda con características de las otras enfermedades que se tienen en el conjunto de muestras.

La idea general es tomar una decisión acerca de si una imagen Western Blot es más parecida al modelo de neurocisticercosis o si es más parecida al modelo de las otras enfermedades. Lo ideal sería que todas las imágenes del conjunto de neurocisticercosis se parezcan más al modelo de neurocisticercosis y las demás al otro modelo.

En el Algoritmo Genético un individuo obtendrá mejor evaluación si los primeros 49 elementos de prueba se parecen más a su primera parte (neurocisticercosis) y si los

elementos restantes se parecen más a la segunda parte (otras enfermedades). Por tanto se define la siguiente función objetivo.

$$f(X) = \omega_1 \sum_{i=1}^M \left[ \sum_{j=1}^{Ns} |X_j - Y_{ij}| \leq \sum_{j=1}^{Ns} |X_{j+Ns} - Y_{ij}| \right] + \omega_2 \sum_{i=M+1}^N \left[ \sum_{j=1}^{Ns} |X_j - Y_{ij}| > \sum_{j=1}^{Ns} |X_{j+Ns} - Y_{ij}| \right]$$

Donde:

X es una imagen WB de dos partes de tamaño  $Ns$  cada una.

Y es una imagen WB perteneciente al conjunto de imágenes muestra,

M es el número de imágenes WB pertenecientes al conjunto de neurocisticercosis,

N es el número total de imágenes WB muestra,

$Ns$  es el número de pixeles entre 77 y 767 dependiendo del rango de muestreo,

$X_i$  representa el pixel  $i$  del histograma X;

$Y_{ij}$  representa al pixel  $i$  del histograma  $j$  del conjunto de imágenes muestra.

$\omega_1$  y  $\omega_2$  representan la proporción en que afectan a la función objetivo tanto el conjunto de neurocisticercosis como el de otras enfermedades. Para este experimento se utilizó la misma proporción, es decir:  $\omega_1 = \omega_2$ .

Al igual que en el capítulo IV, se realizaron 30 pruebas en total, diez pruebas correspondientes a diferentes intervalos de muestreo para cada tipo de recombinación (discreta, intermedia y lineal). Del mismo modo estas imágenes contienen entre 77 y 767 pixels.

Para la evaluación se utilizó una aptitud normalizada con valores entre 0 y 1 con  $r_{\max} =$  número de muestras \* número de imágenes de prueba (201 para esta aplicación).

En las primeras diez pruebas se utilizó 2500 como máximo número de generaciones, mientras que en las demás pruebas se utilizaron 500 generaciones con objeto de reducir el tiempo de procesamiento. Lo anterior basados en el comportamiento resultante en las primeras 10 pruebas que mostraron variaciones poco significativas posteriores a la generación 2500.

Se utilizaron cuatro sub-poblaciones, una probabilidad de cruzamiento de 1.0, una probabilidad de mutación de 0.45, una tasa de migración de 0.2 cada 20 generaciones, el método de selección universal estocástico y 250 individuos por sub-población.

#### V.4. PRUEBAS Y RESULTADOS

En estas pruebas se utilizó una computadora personal con procesador Pentium III, a 550 MHZ, con 128 MB de memoria RAM y 16 GB de memoria secundaria.

*Tabla 5.1. Resultados de pruebas 31 a 60*

Prueba	Número Var.	Tiempo (segs.)	Max. Gen.	Mejor Individuo	Mejor F(x)	Mejor Aptitud	Sensibilidad (%)	Especificidad (%)
031	154	77890	2500	7.462687e-002	0.9254		98	82
032	172	93382	2500	6.467662e-002	0.9353		98	65
033	192	79312	2500	7.960199e-002	0.9204		100	78
034	220	82228	2500	6.965174e-002	0.9303		100	69
035	256	84992	2500	7.462687e-002	0.9254		98	71
036	308	89951	2500	6.467662e-002	0.9353		96	68
037	384	100458	2500	7.462687e-002	0.9254		98	87
038	512	114167	2500	6.965174e-002	0.9303		96	69
039	768	142336	2500	5.970149e-002	0.9403		100	74
040	1534	367970	2500	6.965174e-002	0.9303		100	70
041	154	15547	500	8.457711e-002	0.9154		100	89
042	172	15708	500	8.457711e-002	0.9154		100	83
043	192	16318	500	7.960199e-002	0.9204		100	86
044	220	16784	500	8.457711e-002	0.9154		98	88
045	256	17852	500	7.960199e-002	0.9204		96	87
046	308	18911	500	7.960199e-002	0.9204		98	87
047	384	20965	500	6.467662e-002	0.9353		100	76
048	512	23731	500	6.965174e-002	0.9303		100	74
049	768	28852	500	7.960199e-002	0.9204		100	79
050	1534	73399	500	8.955224e-002	0.9104		100	80
051	154	15464	500	8.457711e-002	0.9154		86	87
052	172	15879	500	9.452736e-002	0.9055		92	74
053	192	16127	500	9.452736e-002	0.9055		98	84

054	220	16992	500	9.452736e-002	0.9055	88	80
055	256	18034	500	1.094527e-001	0.8905	98	89
056	308	19274	500	9.452736e-002	0.9055	96	86
057	384	20761	500	7.462687e-002	0.9254	98	87
058	512	24160	500	8.955224e-002	0.9104	94	78
059	768	28805	500	1.044776e-001	0.8955	96	87
060	1534	77322	500	1.044776e-001	0.8955	90	74

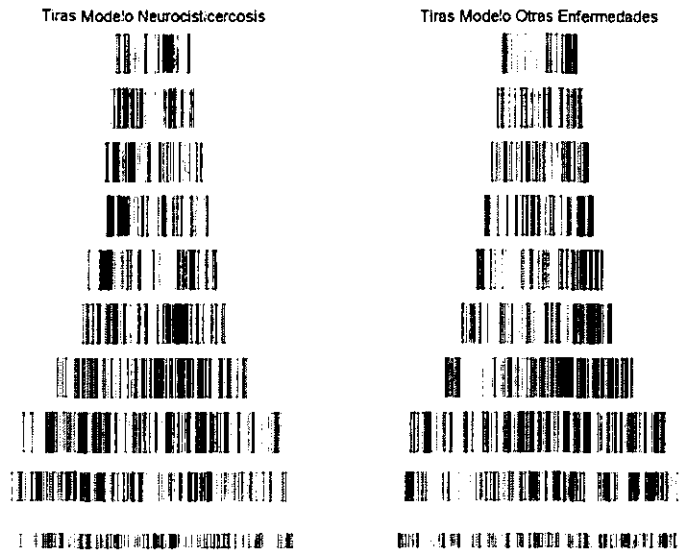
Para todas las pruebas realizadas se utilizó un solo archivo de entrada de parámetros denominado *paramet.txt*, cuyos detalles pueden ser analizados en el apéndice A.

En la tabla 5.1. se puede observar que el tiempo de procesamiento se incrementa linealmente de acuerdo al número de variables consideradas. Sin embargo, este hecho no afecta la calidad de los resultados en términos de la sensibilidad y la especificidad de la prueba.

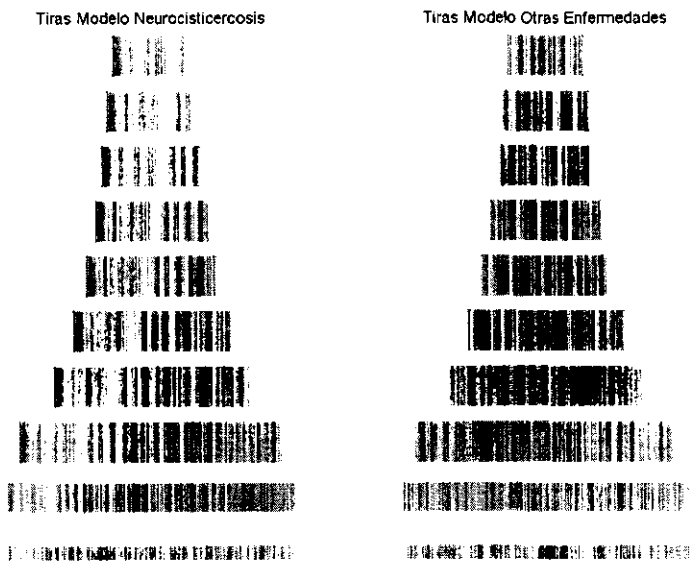
La sensibilidad o capacidad de la prueba para detectar la infección, alcanza valores que van de 86 a 100% y la especificidad, o capacidad de la prueba para no confundir la infección con otros padecimientos, alcanza valores de 65 a 89%.

Estos resultados nos muestran que este método para definir una prueba es efectivo incluso para diferentes intervalos de muestreo, lo que la hace más confiable para auxiliar el diagnóstico. Es interesante notar que los resultados en términos de sensibilidad y especificidad son muy similares para los tres tipos de recombinación, con una ligera ventaja en las pruebas correspondientes a recombinación intermedia.

En las figuras 23 a 25 se puede observar que las imágenes obtenidas no siguen un patrón definido que auxilie al especialista a través de una simple inspección visual, lo que implica que esta prueba necesariamente debe hacerse de forma automatizada.



**Figura Error! Unknown switch argument.: Imágenes WB para decisión**



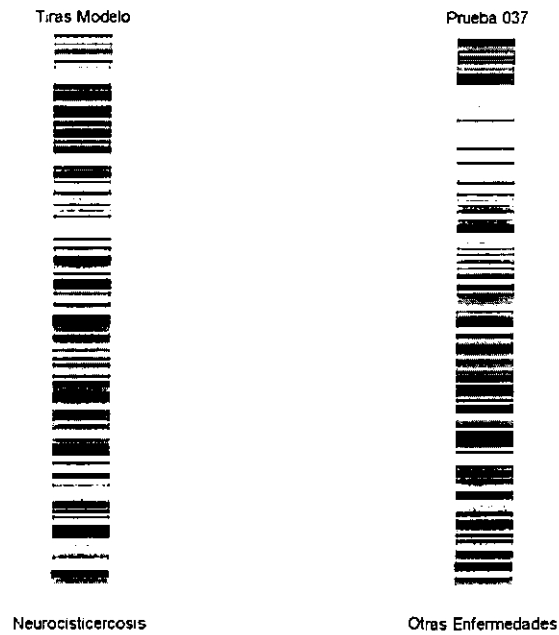
**Figura Error! Unknown switch argument.: Imágenes WB para**



**Figura Error! Unknown switch argument.: Imágenes WB para**

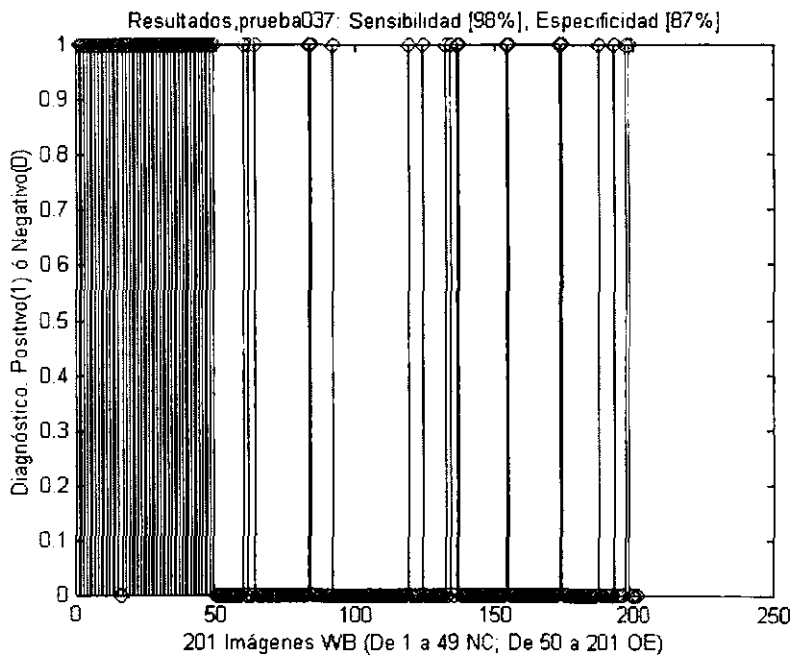
La prueba automática se lleva a cabo obteniendo el histograma del caso en estudio mediante el método descrito en la sección IV.2. y en la figura 13 del capítulo IV. Una vez obtenido el histograma, se toma un número de muestras (equidistantes) igual a alguno de los números de muestras utilizados para obtener las imágenes modelo. Se realiza la comparación y se determina a que modelo se parece más el histograma en cuestión, es decir, se toma la decisión si pertenece a un paciente infectado o no.

Por ejemplo, la figura 26 muestra las imágenes modelo obtenidas en la prueba 037 de la tabla 5.1. Para nosotros como humanos, no es fácil discernir a cual imagen se parece más aquella que deseamos clasificar. Sin embargo utilizando un programa con diferencias absolutas dicha clasificación es sencilla.



**Figura Error! Unknown switch argument.: Imágenes**

Considerando estas dos imágenes modelo para clasificar los 201 histogramas iniciales, se obtiene un 98% de sensibilidad y un 87% de especificidad. Los resultados se muestran en la figura 27.



**Figura Error! Unknown switch argument.: Resultados**

Los resultados obtenidos en estos experimentos demuestran que este método puede ser utilizado como auxiliar en el diagnóstico de la neurocisticercosis con el apoyo de un conjunto de imágenes de muestra como las 201 que se utilizaron en este experimento.

Si tomamos en cuenta que algunos de los resultados obtenidos con este método superan los obtenidos en otras pruebas realizadas sólo con suero (a ELISA por ejemplo), estamos hablando de que este método puede ser considerado como un auxiliar muy eficaz en zonas en las que no se tiene acceso a pruebas más sofisticadas o bien como un paso previo a dichos estudios.

No obstante se debe aclarar que las imágenes modelo se han obtenido en base a elementos conocidos en pruebas controladas. Sería deseable llevar a cabo un estudio “en ciego” (en el que los resultados de diagnóstico no se conozcan) que involucre un conjunto de imágenes diferentes a las utilizadas en el experimento. Lo anterior nos permitiría evaluar más precisamente el desempeño del método.

Los resultados en general muestran un alto porcentaje de sensibilidad lo que indica que es un método muy eficiente para determinar la presencia del parásito, sin dejar de lado la probabilidad relativamente alta (11% en el mejor de los casos) de diagnosticar *falsos positivos*. Sin embargo, es preferible el diagnóstico de falsos positivos que de falsos negativos, ya que en el primer caso se puede solicitar una segunda prueba mientras que en el segundo el diagnóstico erróneo podría perjudicar al paciente.

Es importante señalar que el método propuesto además tiene la característica de que puede ser reproducido bajo diferentes condiciones y por tanto generar imágenes modelo específicas para zonas determinadas (ej. zonas endémicas).



## VI. PROGRAMACIÓN GENÉTICA APLICADA A IMÁGENES WESTERN BLOT

### VI.1. INTRODUCCIÓN

Los capítulos IV y V han presentado dos aproximaciones para generar imágenes modelo que puedan ser utilizadas para auxiliar el diagnóstico eficiente de la neurocisticercosis. Este capítulo presenta una variación basada en el método de Programación Genética.

En la literatura se ha sugerido que la Programación Genética puede ser utilizada para el procesamiento digital de imágenes (Poli, 1996). Trabajos recientes han demostrado la efectividad de esta técnica para resolver algunos problemas relacionados con esta área, como son: detección de bordes (Ross *et al.*, 2000), segmentación (Chiu *et al.*, 2000), orientación (Roberts y Howard, 2000), detección de objetivos (Benson, 2000) y clasificación (Rauss *et al.*, 2000).

Con este trabajo preliminar se busca mostrar la posibilidad de la utilización de esta técnica para apoyar el diagnóstico. Se considera que el potencial ofrecido por esta forma de representación de programas amerita explorarla como alternativa de solución.

La utilización de Programación Genética e imágenes Western Blot para apoyar al diagnóstico de la neurocisticercosis tiene como objetivo fundamental encontrar un programa de computadora que pueda emitir un diagnóstico auxiliar. La idea general es que el programa buscado se acerque lo más posible al diagnóstico acertado con base en las imágenes Western Blot disponibles de pacientes.

Con el programa se debe obtener un diagnóstico positivo cuando se le proporcione como entrada una imagen Western Blot correspondiente a un paciente infectado y se debe obtener un diagnóstico negativo en cualquier otro caso. La principal ventaja de encontrar una prueba auxiliar en forma de programa es que el proceso se puede automatizar directamente

y que mientras mayor sea el número de muestras para la generación del programa, los resultados deberán ser más aproximados a un diagnóstico acertado.

## VI.2. DESCRIPCIÓN

Al igual que en las aproximaciones anteriores se cuenta con 201 imágenes, de las cuales, las primeras 49 corresponden a casos confirmados de neurocisticercosis y las restantes a otras enfermedades. Para este experimento se utiliza nuevamente la representación en forma de histogramas.

Considerando el valor “uno” como diagnóstico positivo y el valor “cero” como negativo, el programa buscado debe proporcionar los siguientes resultados:

Dada:	Imagen 1	Diagnóstico=1
Dada:	Imagen 2	Diagnóstico=1
.		
.		
.		
Dada:	Imagen 49	Diagnóstico=1
Dada:	Imagen 50	Diagnóstico=0
Dada:	Imagen 51	Diagnóstico=0
.		
.		
.		
Dada:	Imagen 201	Diagnóstico=0

A mayor número de aciertos en el diagnóstico, el programa tendrá un mejor desempeño. Se busca asimismo generar un programa que reciba como entrada un histograma (vector de números reales) y que proporcione como salida un diagnóstico (cero ó uno). Dadas estas características el programa se puede modelar como un problema de regresión simbólica (ver sección II.4.5).

### VI.3. PROGRAMACIÓN GENÉTICA (FORMULACIÓN)

Como se mencionó en el capítulo II, la Programación Genética requiere del uso de funciones y terminales:

$$F=\{+, -, *, /, >, <, \leq, \geq\}$$

$$T=\{X, R\}$$

Donde para esta aplicación,

$X$  es el conjunto de imágenes Western Blot representada como una matriz que contiene 201 vectores, cada vector representa un histograma (valores reales de 0 a 1)  
 $R$  es el conjunto de los números reales entre 0 y 1. Todas las funciones propuestas utilizan dos operandos, por lo que la aridad es siempre dos.

El resultado esperado es por tanto un vector de 201 valores. El vector debe contener “unos” de la posición 1 a la 49 y “ceros” de la posición 50 a la 201. Este vector óptimo o deseado representa el diagnóstico acertado de los 201 casos.

Estamos buscando entonces un programa cuyo resultado minimice el número de errores, es decir:

$$G(P(X)) = \min(V_{P(X)} - V_{opt}) \dots\dots\dots (7)$$

Donde:

- $X$  es la matriz que representa los 201 histogramas,
- $P(X)$  es el programa a evaluar,
- $V_{P(X)}$  es el vector resultado obtenido con el programa  $P(X)$ ,
- $V_{opt}$  es el vector de *unos* y *ceros* deseado,
- $G(P(X))$  es la función que desea minimizar el programa.

Los programas a evaluar en cada generación del algoritmo de Programación Genética se representan en forma de vector, y cada vector tiene valores numéricos que representan un programa en notación prefija.

La siguiente tabla muestra la representación de cada valor numérico utilizado:

<i>Valor</i>	0-0.99	1	2	3	4	5	6	7	8	11	12
<i>Representa</i>	El mismo valor	+	-	*	/	>	<	≥	≤	X	R

Cuando se generan los posibles programas aleatorios en forma de vector, se utilizan valores de 1 a 8 y de 11 a 12, y donde se encuentra un valor correspondiente a *R* se convierte a un número aleatorio entre 0 y 0.999.

Un ejemplo de programa generado sería el vector:

3	1	12	11	2	12	12
---	---	----	----	---	----	----

Después de la posible conversión quedaría:

3	1	0.428	X	2	0.2169	0.1122
---	---	-------	---	---	--------	--------

Y eso representaría al siguiente programa en notación prefija:

$(* (+ 0.428 X) (- (0.2169) (0.1122)))$

En la función de evaluación se realiza la conversión de cada vector a evaluar hacia un formato comprensible por el intérprete de Matlab. Esto lo lleva a cabo la *función01* descrita en el apéndice B.

Una vez hecha la conversión, se aplica el programa a los valores de la matriz  $X$  y se obtiene una matriz  $Y$ . La suma de las *diferencias* absolutas de  $X-Y$  se almacena entonces en un vector de 201 valores reales.

Como se dijo anteriormente, se busca que la matriz  $Y$  sea muy parecida a la matriz  $X$  en los primeros 49 renglones, y muy diferente en los demás. Para determinar la evaluación óptima del programa  $P(X)$  se genera un vector de *umbrales* normalmente distribuidos entre la menor y la mayor *diferencia*. Para cada uno de los *umbrales* se toma una decisión con cada una de las *diferencias*. Si la *diferencia* es menor o igual al *umbral* el resultado es “uno” (diagnóstico positivo), si es mayor el resultado es “cero” (diagnóstico negativo).

Todo el proceso descrito genera un *vector resultante* para cada programa. A cada *vector resultante* se evalúa en cada generación del algoritmo, y se busca minimizar el número de errores. Para este propósito se utiliza una aptitud normalizada, para tener valores entre 0 y 1 con  $r_{\max}$  = número de imágenes de prueba.

Las operaciones de cruzamiento y mutación se realizan en forma estándar conforme a la descripción detallada en la sección II.4.

Para evitar realizar un programa para cada prueba, se generó un solo programa para la realización de todos los experimentos con la entrada de parámetros desde un archivo externo (Programa 1 del Apéndice B).

Se realizaron 27 pruebas en total con probabilidades de cruzamiento de 0.95, 0.85 y 0.75 (9 pruebas cada uno). Intervalos de muestreo de los histogramas de 3, 6 y 9 (9 pruebas cada uno) debido a que en los experimentos realizados anteriormente se demostró que su influencia no es determinante en los resultados. Asimismo, se utilizaron 32, 64 y 128 como máximo número de nodos (9 pruebas cada uno), una probabilidad de mutación de 0.005, selección universal estocástica, una sola población de 150 individuos y un máximo número de generaciones de 100. La función objetivo utilizada (funcion1) se incluye en el apéndice B.

## VI.4. PRUEBAS Y RESULTADOS

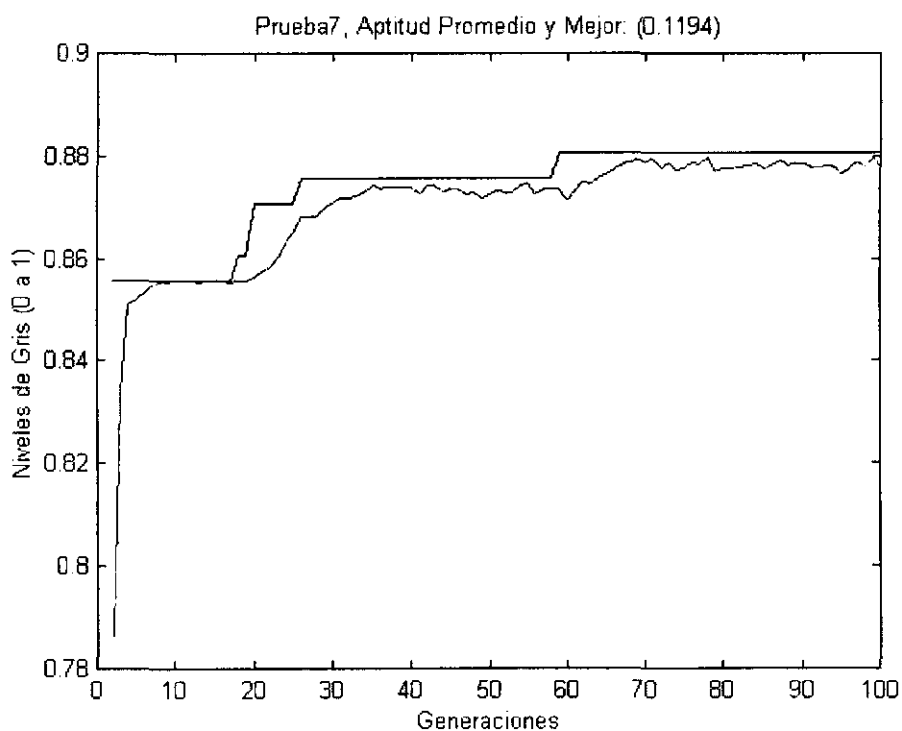
En estas pruebas se utilizó una computadora personal con procesador Pentium III, a 550 MHZ, con 128 MB de memoria RAM y 16 GB de memoria secundaria.

Tabla 6.1. Resultados de Pruebas 1 A 27

Prueba	Número Var.	Max. Nodos	Max. Gen.	Tiempo (segs.)	Mejor Aptitud	Sensibilidad (%)	Especificidad (%)
001	86	32	100	2683	0.8756	40.82	96.71
002	128	32	100	3874	0.8806	38.78	96.71
003	256	32	100	1009	0.8607	48.98	97.37
004	86	64	100	5379	0.8756	42.86	97.37
005	128	64	100	6635	0.8657	48.98	98.03
006	256	64	100	9471	0.8657	48.98	98.03
007	86	128	100	10023	0.8806	42.86	98.03
008	128	128	100	10916	0.8557	46.94	96.05
009	256	128	100	16691	0.8756	42.86	97.37
010	86	32	100	2744	0.8806	38.78	96.71
011	128	32	100	4307	0.8756	40.82	96.71
012	256	32	100	4339	0.8756	42.86	97.37
013	86	64	100	4971	0.8706	46.94	98.03
014	128	64	100	6486	0.8756	40.82	96.71
015	256	64	100	8809	0.8756	42.86	97.37
016	86	128	100	10089	0.8706	38.78	95.39
017	128	128	100	12364	0.8756	42.86	97.37
018	256	128	100	19233	0.8706	40.82	96.05
019	86	32	100	2824	0.8706	46.94	98.03
020	128	32	100	2729	0.8806	42.86	98.03
021	256	32	100	4606	0.8856	38.78	97.37
022	86	64	100	5413	0.8756	42.86	97.37
023	128	64	100	6220	0.8806	40.82	97.37
024	256	64	100	3396	0.8756	42.86	97.37
025	86	128	100	10192	0.8756	42.86	97.37
026	128	128	100	12628	0.8706	38.78	95.39
027	256	128	100	18319	0.8557	46.94	96.05

Se puede observar en la tabla 6.1. que el tiempo de procesamiento se incrementa de manera proporcional al número de muestras utilizados y al máximo número de nodos, sin embargo los resultados no presentan variaciones sustanciales, presentándose los mejores resultados para las pruebas 5 y 6 donde se alcanza un 48.08% de sensibilidad y un 98.03% de especificidad. En general, los resultados van del 38.78 al 48.08% de sensibilidad y del 95.39 al 98.03% de especificidad.

La figura 28 muestra el resultado obtenido en la prueba 7. Se observa en ella la evolución del mejor individuo y del promedio en términos de aptitud.



**Figura Error! Unknown switch argument.: Evolución de**

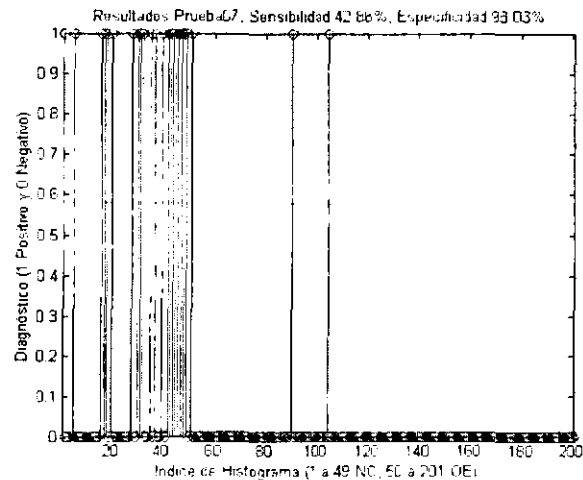
El programa resultante después de 100 generaciones es el siguiente:

```
feval('times', feval('plus', feval('minus', feval('lt', x, feval('ge', 0.35907, feval('plus', feval('plus', feval('gt', feval('le', x, feval('times', 0.77711, feval('times', x, x))), x), feval('spdivide', 0.0053562, x))), feval('times', feval('gt', x, feval('times', 0.63461, 0.3088))), feval('minus', feval('gt', x, feval('lt', feval('lt', feval('le', 0.0034952, 0.56844), 0.38839), feval('spdivide', feval('lt', feval('spdivide', feval('spdivide', x, x), x), feval('minus', x, feval('times', feval('gt', x, x), 0.99589))), feval('plus', feval('spdivide', x, x), feval('spdivide', 0.0034952, x))))), feval('times', 0.36485, feval('minus', 0.33188, feval('gt', x, feval('lt', x, 0.77576))))))))) feval('plus', feval('gt', 0.15579, feval('lt', feval('lt', feval('le', 0.69268, feval('minus', feval('times', x, feval('ge', 0.9216, x))), feval('spdivide', 0.3506, 0.72733))), x), feval('spdivide', feval('lt', feval('spdivide', x, x), feval('minus', x, 0.40134)), feval('plus', feval('le', x, feval('times', 0.77711, feval('times', x, x))), feval('spdivide', 0.0034952, x))))), feval('gt', x, feval('spdivide', feval('times', 0.5725, feval('lt', 0.63461, 0.82178))), feval('plus', feval('le', feval('lt', 0.36485, 0.3088), x), feval('le', 0.0034952, 0.85083))))), feval('gt', x, x), 0.52512)
```

Que en notación prefija quedaría:

```
(* (+ (- (< x (>= 0.35907 (+ (+ (> (<= x (* 0.77711 (* x x))) x) (/ 0.0053562 x)) (* (> x (* 0.63461 0.3088)) (- (> x (< (<= 0.0034952 0.56844) 0.38839) (/ (< (/ (/ x x) x) (- x (* (> x x) 0.99589))) (+ (/ x x) (/ 0.0034952 x)))))) (* 0.36485 (- 0.33188 (> x (< x 0.77576)))))) (+ (> 0.15579 (< (< (< 0.69268 (- (* :: '> 0.9216 x)) (/ 0.3506 0.72733))) x) (/ (< (/ x x) (- x 0.40134)) (+ (<= x (* 0.77711 (* x x)) (/ 0.0034952 x)))))) (> x (/ (* 0.5725 (< 0.63461 0.82178)) (+ (<= (< 0.36485 0.3088) x) (<= 0.0034952 0.85083)))))) (> x x) 0.52512)
```





**Figura Error! Unknown switch argument.:**

Después de aplicar el criterio de decisión a la prueba 7, se observa una sensibilidad del 42.86% y una especificidad del 98.03%.

Es claro que los resultados obtenidos no mejoran los de las aproximaciones descritas en los capítulos IV y V, pero ejemplifican la utilización de la Programación Genética en este contexto. Es importante señalar que el tiempo de procesamiento tomado por esta aproximación, se reduce hasta en un 60% con respecto de las aproximaciones con Algoritmos Genéticos.

Trabajo futuro debe explorar nuevas formas de formular este problema, así como incorporar mejores operadores que eviten la convergencia prematura observada en la figura 28. Asimismo se puede ampliar el conjunto de funciones (ej. exp, log, etc.), o bien hacer un nuevo planteamiento de la función objetivo para utilizar operadores de otro dominio (filtrado, por ejemplo).

Otra alternativa interesante sería el tratar de definir un criterio para aceptar la presencia de bandas en las imágenes WB, de esa forma podría buscarse un programa para la definición de un patrón de diagnóstico del tipo `(if ((banda3 AND banda4) OR banda2)) diagnóstico=1).`

## VI.5. CONCLUSIONES

En este capítulo se presentan los resultados de experimentos para encontrar programas auxiliares de diagnóstico de la neurocisticercosis basados en imágenes Western Blot. La búsqueda de los programas se realiza con la técnica de Computación Evolutiva conocida como Programación Genética.

En los experimentos realizados, se encontraron programas cuyos resultados van del 39 al 48% de sensibilidad y del 95 al 98% de especificidad. El mejor de ellos obtuvo un 48% de sensibilidad y un 98% de especificidad. Es claro que aunque no son comparables a los resultados obtenidos en los capítulos IV y V, si se puede observar una reducción considerable en el tiempo de procesamiento del algoritmo.

Estos resultados preliminares sugieren la exploración de nuevas alternativas, por ejemplo: que si se amplía el conjunto de funciones se puede obtener una mayor calidad en las soluciones. También sería aconsejable la utilización de otros operadores genéticos, como edición y permutación, así como la exploración de programas un dominio diferente al de la regresión simbólica, por ejemplo, filtrado de señales.

## VI.6. REFERENCIAS

- Benson, Karl. 2000. "Evolving Finite State Machines with Embedded Genetic Programming for Automatic Target Detection". *Proceedings of the Congress on Evolutionary computation*. IEEE. Vol.2. Págs. 1543-1549. La Jolla, California, EUA.
- Chiu, P.; Girgensohn, A.; Polak, W.; Rieffel, E.; Wilcox, L. y Bennett, F.H. 2000. "A Genetic Segmentation Algorithm for Data Streams and Video". *Proceedings of the Genetic and Evolutionary Computation Conference*. Darrel Whitley, *et al* editores. Págs. 666-673. Jul. 10-12. Las Vegas, Nevada, EUA.

## VII. CONCLUSIONES

### VII.1. CONCLUSIONES GENERALES

La neurocisticercosis humana es una enfermedad parasitaria del sistema nervioso central que representa un serio problema de salud pública en varios países del mundo. El diagnóstico acertado de esta enfermedad actualmente se basa en la experiencia del médico especialista y en el apoyo de algunas pruebas de laboratorio.

La prueba Western Blot (WB) ha causado especial interés para el análisis del comportamiento del sistema inmunológico ante la presencia del parásito. En esta tesis se utiliza la versión digital de pruebas WB (imágenes WB) y técnicas de Computación Evolutiva para encontrar métodos auxiliares de diagnóstico.

El objetivo es proporcionar un método para auxiliar a los médicos especialistas en el diagnóstico acertado de este padecimiento en estudios epidemiológicos o clínicamente como un paso previo a pruebas más costosas (como la tomografía axial computarizada) o más peligrosas y dolorosas para el paciente (como la extracción de líquido cefalorraquídeo).

La Computación Evolutiva se compone de técnicas para búsqueda y optimización basadas en los mecanismos de la evolución y de la genética natural. Parten de una población inicial generada aleatoriamente que representa a un conjunto de soluciones posibles al problema considerado. En estas técnicas se busca la reproducción de los individuos más aptos de la población para mejorar los resultados. Algoritmos Genéticos y Programación Genética son dos de las técnicas de Computación Evolutiva que gozan de mayor popularidad.

El trabajo presentado en esta tesis desarrolla dos aproximaciones de solución basadas en Algoritmos Genéticos y otra basada en Programación Genética. Dichas propuestas buscan

incrementar tanto la *sensibilidad* como la *especificidad* de los métodos hasta ahora existentes.

En una primera aproximación se llevó a cabo la búsqueda de una imagen Western Blot modelo para auxiliar el diagnóstico de la neurocisticercosis. Los resultados obtenidos brindan información acerca de las *bandas* que se presentan más frecuentemente en las imágenes Western Blot de pacientes infectados y esta información puede ser de utilidad para afinar la prueba relacionada a esta enfermedad. Al evaluar esta primera propuesta de solución con las imágenes Western Blot disponibles de neurocisticercosis y de otras enfermedades se determinó que alcanza una sensibilidad del 55% y una especificidad del 98%, lo cual no es suficiente para dar un diagnóstico fehaciente. Este resultado muestra que realmente existe una respuesta inmunológica de los pacientes infectados, pero que esta respuesta no es homogénea, lo que sugiere tratar de extraer más información de las imágenes WB tanto de los pacientes que están infectados como de los que no.

En una segunda aproximación se realizó la búsqueda de pares de imágenes Western Blot, de cada par, una imagen debería contener características del conjunto de neurocisticercosis y la otra del conjunto de otras enfermedades. La idea es, que al comparar digitalmente las imágenes disponibles con los pares de propuestas de solución, cada imagen tuviera un mayor parecido a la imagen correspondiente del par propuesto. Los resultados muestran que el método alcanza hasta un 100% de sensibilidad y un 89% de especificidad lo que indica que es efectivo para apoyar el diagnóstico de esta enfermedad. Estos resultados superan a los obtenidos con otros métodos de diagnóstico basados en suero (ej. ELISA) y se aproximan a los de extracción de líquido cefalorraquídeo y de tomografía computarizada, con la diferencia de que este método es más económico y más seguro que los mencionados. Adicionalmente presenta la ventaja de que es automático y es un excelente candidato para llevar a cabo estudios a gran escala. Sin embargo, antes de ser aplicado, se requiere de un esfuerzo adicional en el que se evalúen imágenes diferentes al conjunto utilizado para la obtención del modelo para poder determinar la eficiencia real del método.

Los resultados obtenidos por los Algoritmos Genéticos sugieren explorar alternativas más efectivas de solución. La Programación Genética al producir programas-solución, proporciona un marco atractivo para la generación de soluciones más eficientes tanto en la calidad de la respuesta, como en el tiempo de ejecución.

Los resultados preliminares generaron un programa que alcanza una 48% de sensibilidad y un 98% de especificidad. Es evidente que estas cifras están por debajo de las obtenidas en las aproximaciones de Algoritmos Genéticos, sin embargo se logró reducir sustancialmente el tiempo de procesamiento empleado en generar la imagen modelo. No obstante, se sugiere seguir explorando esta alternativa con un conjunto de funciones diferentes o bien con una nueva formulación del problema.

Cabe destacar que para cada uno de los métodos propuestos se han utilizado diferentes tamaños en el intervalo de muestreo de las imágenes disponibles, y en general, los resultados no se han visto afectados. Esto indica que se puede reducir el tiempo de procesamiento sin afectar la calidad de los resultados. Más aún, se puede aprovechar el paralelismo intrínseco característico de los algoritmos evolutivos para su implementación en multiprocesadores.

Los resultados generales obtenidos son muy alentadores ya que presentan un método automático, eficiente, económico y reproducible para auxiliar el diagnóstico de la neurocisticercosis. Sin perder de vista las limitantes ya expuestas, se puede considerar que este trabajo contribuye con una herramienta poderosa para apoyar la lucha contra esta enfermedad.

## VII.2. TRABAJO FUTURO

Durante el desarrollo de este proyecto surgieron muchas ideas para mejorar las aproximaciones descritas en esta tesis. Aquí se bosquejan algunas.

- El problema tratado en esta tesis puede ser visualizado como un intento de alcanzar una mayor sensibilidad y una mayor especificidad en una sola prueba. Estas dos características pueden contraponerse en muchos casos, lo que sugiere el uso de técnicas de optimización que contemplen varios objetivos. Sería interesante comparar los resultados del capítulo V con los resultados obtenidos de un experimento que utilice técnicas de optimización multi-objetivos.
- Ampliar los experimentos hechos con Programación Genética, posiblemente modelándolos con funciones de filtrado o un conjunto más amplio de funciones matemáticas (exp, log, etcétera).
- Definir criterios para la definición de “bandas” en las imágenes Western Blot (ej. considerando intensidad y tamaño en el histograma). Esto podría servir para encontrar patrones de diagnóstico que sean más útiles al especialista tales que puedan resumirse en una frase del tipo: “Nuestros resultados demuestran que la presencia de la banda X y de la banda Z, sin la presencia de la banda Y, son un fuerte indicio de que el paciente padece la infección”.
- Analizar el problema con técnicas clásicas de estimación y de detección, para tratar de modelar el “ruido” que aparece en las imágenes WB y así filtrar las imágenes antes de hacer otro procesamiento.
- Un problema observado en los experimentos realizados, es que requieren mucho tiempo de procesamiento cuando el número de muestras es elevado. Este problema puede ser solucionado si se aprovechan las características de paralelismo implícitas en los Algoritmos Genéticos y se implementan en una arquitectura paralela o distribuida.

# APÉNDICES

## APÉNDICE A: PROGRAMAS Y ARCHIVOS PARA ALGORITMOS GENÉTICOS

### PROGRAMA 1

```

% Apéndice A
% Programa #1
% Programa en Matlab para generar Matrices e Histogramas
% a partir de imagenes Western Blot

% INPUT
% Lee 201 archivos PCX que contienen imagenes WB
% OUTPUT
% Genera un archivo 'matrices.mat' con una matriz de 201-MaxAncho-MaxLargo
% Genera un archivo 'histogramas.mat' con una matriz de 201-MaxLargo
% Genera un archivo 'largo.mat' con un vector de 201 conteniendo los tamaños a lo largo.
% Genera un archivo 'ancho.mat' con un vector de 201 conteniendo los tamaños a lo ancho.

clear;
% Directorio donde se encuentran las imágenes
directorio='c:\imagenes wb pcx\';

for i=1:201,
    % Lectura de imágenes WB de archivos en formato PCX
    if i<10, eval(strcat('tira00',int2str(i),'=imread('',...
        directorio,'tira00',int2str(i),'.pcx');'));
    elseif i<100, eval(strcat('tira0',int2str(i),'=imread('',...
        directorio,'tira0',int2str(i),'.pcx');'));
    else eval(strcat('tira',int2str(i),'=imread('',directorio,...
        'tira',int2str(i),'.pcx');'));
    end

    % Generación de vectores de ancho y largo
    if i<10, eval(strcat('[largo(i) ancho(i)]=size(tira00',int2str(i),')'));
    elseif i<100, eval(strcat('[largo(i) ancho(i)]=size(tira0',int2str(i),')'));
    else eval(strcat('[largo(i) ancho(i)]=size(tira',int2str(i),')'));
    end

    % Generación de matriz de histogramas
    if i<10, eval(strcat('histogramas(',int2str(i),',1:largo(',int2str(i),...
        '))=mean(tira00',int2str(i),')'));

```

```

elseif i<100, eval(strcat('histogramas(',int2str(i),'1:largo(',int2str(i),...
    '))=mean(tira0',int2str(i),'');'));
else eval(strcat('histogramas(',int2str(i),'1:largo(',int2str(i),...
    '))=mean(tira',int2str(i),'');'));
end

```

Generación de matriz de imágenes

```

if i<10, eval(strcat('matrices(',int2str(i),'1:ancho(',int2str(i),...
    '),1:largo(',int2str(i),''))=tira00',int2str(i),'');'));
elseif i<100, eval(strcat('matrices(',int2str(i),'1:ancho(',int2str(i),...
    '),1:largo(',int2str(i),''))=tira0',int2str(i),'');'));
else eval(strcat('matrices(',int2str(i),'1:ancho(',int2str(i),...
    '),1:largo(',int2str(i),''))=tira',int2str(i),'');'));
end
end

```

```
clear tira* i directorio;
```

Almacenamiento de archivos generados

Normalmente en el directorio "work" de Matlab  
 a menos que se especifique otra cosa en las opciones.

```

save('histogramas','histogramas');
save('matrices','matrices');
save('ancho','ancho');
save('largo','largo');

```



## PROGRAMA 2

```
% Apéndice A
% Programa #2
% Genera histogramas a partir de archivos ASCII

% INPUT
% Lee 201 archivos ASCII que contienen el tamaño y los histogramas WB
% con valores de 0 a 255

% OUTPUT
% Genera un archivo 'histos.mat' que contiene:
% Una matriz 'histos' con los histogramas de tamaño 201 * MaxLargo
% Un vector 'largo' de tamaño 201 con los tamaños de los histogramas

clear;
directorio='c:\ivan\tesis\histogramas\';

for i=1:201,
    if i<10 fname=strcat(directorio,'tira00',int2str(i));
        elseif i<100 fname=strcat(directorio,'tira0',int2str(i));
            else fname=strcat(directorio,'tira',int2str(i));
        end
    load (fname, '-ASCII');
    if i<10
        eval(strcat('largo (i) =tira00',int2str(i),'(1);'));
        eval(strcat('histos(i,1:largo(i)-1) =tira00',int2str(i),'(2:largo(i));'));
    elseif i<100
        eval(strcat('largo (i) =tira0',int2str(i),'(1);'));
        eval(strcat('histos(i,1:largo(i)-1) =tira0',int2str(i),'(2:largo(i));'));
    else
        eval(strcat('largo (i) =tira',int2str(i),'(1);'));
        eval(strcat('histos(i,1:largo(i)-1) =tira',int2str(i),'(2:largo(i));'));
    end
end;

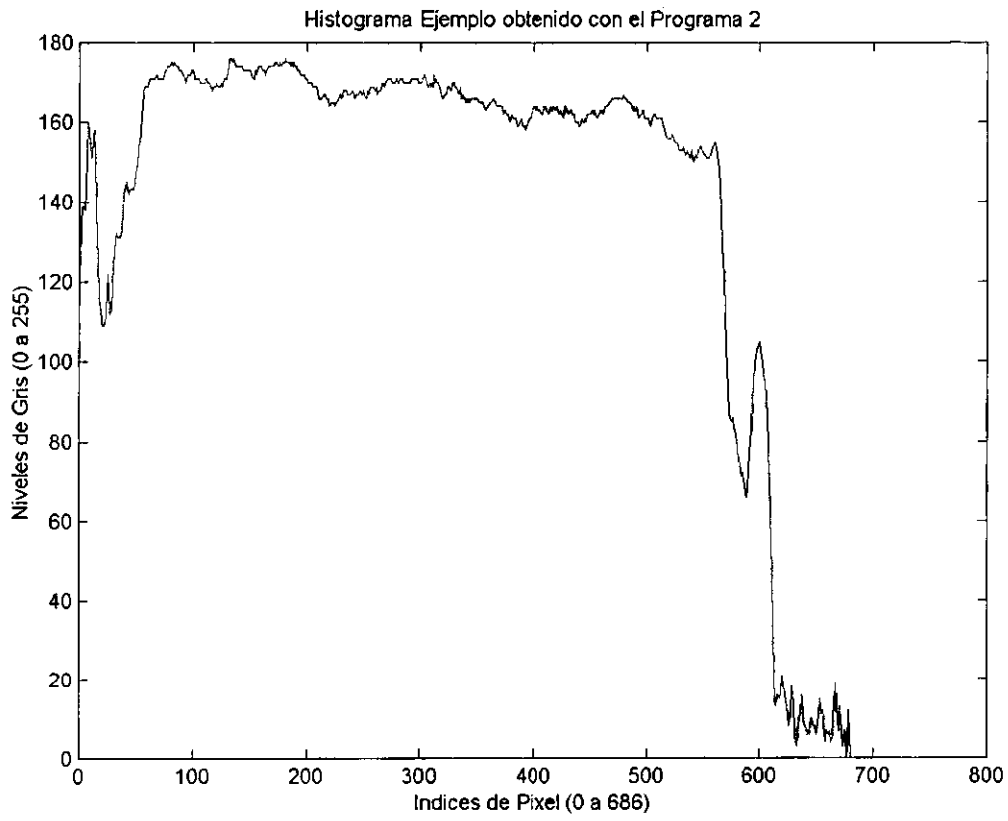
%Para guardar las variables
save('histos','largo','histos');
```

## EJEMPLO DE ARCHIVO DE HISTOGRAMA

Archivo de texto que representa a un histograma, el primer valor representa el *largo* del histograma y los siguientes *largo* valores representan niveles de gris de 0 a 255.

```
682 28 34 43 57 65 64 67 95 133 133 111 100 120 153 164 118 69 46 36 34 36 33 30 30 34 43
54 59 62 64 62 66 79 95 112 128 145 161 170 167 162 157 157 159 157 151 146 150 157 159 169
180 187 189 192 194 193 192 193 195 195 195 197 197 198 198 198 198 199 200 201 201 201 201
201 202 204 205 206 206 207 208 207 206 207 208 207 207 206 205 205 202 202 202 201 200 199
197 196 195 195 192 190 188 187 186 184 184 183 184 185 187 188 190 191 190 189 187 187 186
186 186 186 187 188 188 188 187 187 186 184 183 183 182 181 181 181 178 175 173 170 169 167
168 169 169 171 172 172 171 170 167 164 161 158 157 156 157 159 163 165 167 165 164 164 159
155 159 159 160 162 162 165 167 167 169 173 176 178 181 183 183 184 186 186 186 186 187 187
186 187 188 187 187 187 187 187 187 187 188 187 187 187 186 186 186 186 185 185 187 189 189
190 190 191 192 192 191 191 189 188 188 187 187 187 185 185 185 184 185 187 190 193 195 197
198 197 195 193 191 191 190 188 189 190 189 189 190 190 192 192 192 193 193 191 189 185 181
179 178 177 177 178 178 178 179 180 182 184 185 185 186 187 187 188 190 190 192 195 198 200
202 202 204 205 206 206 205 205 206 205 206 206 206 206 206 205 205 205 205 205 206 205 206
206 206 206 207 206 206 206 206 206 206 206 206 205 205 202 201 202 201 202 203 203 200 199
199 198 198 194 192 190 192 192 193 194 194 194 192 190 188 186 184 183 181 179 177 176 176
174 172 172 170 168 167 169 169 169 168 169 169 169 169 168 167 167 166 166 166 166 167 167
167 166 165 164 162 162 162 160 160 159 159 157 155 154 150 148 147 142 138 135 134 132 129
125 125 125 125 123 121 120 120 120 120 121 121 120 119 116 113 112 111 110 108 107 105 106
107 107 105 106 106 107 106 104 107 108 109 110 112 113 113 113 114 116 118 119 120 120 119
122 124 125 126 129 131 133 134 135 136 135 134 134 135 137 136 136 136 137 138 139 139 140
141 143 143 143 145 145 144 144 144 144 143 141 140 141 139 137 136 135 134 134 133 132 132
133 133 132 132 133 134 135 137 139 141 143 151 155 157 161 162 163 162 163 164 165 166 166
167 167 167 166 167 165 164 161 156 152 150 142 126 117 114 103 95 89 85 78 75 72 70 68 67
68 69 70 69 68 67 67 67 67 68 68 69 71 73 76 75 77 80 80 76 73 69 66 67 71 83 87 89 88 87
84 79 73 64 52 36 27 20 17 16 16 16 17 18 19 22 24 25 26 27 26 23 21 19 16 16 18 21 30 43
59 79 99 114 124 131 131 126 122 111 92 70 50 34 23 16 12 11 9 6 3 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0
```

Este sería el histograma obtenido por el programa 2 con los datos del archivo de ejemplo.



## PROGRAMA 3

```
* Apendice A
* Programa #3
% metawbga      Meta Algoritmo Genético Multipoblación
%              para tiras Western Blot
%
%
% INPUT
% Este programa lee varios parámetros desde
% un archivo ASCII 'parametros.txt'.
% Con esos parametros ejecuta varias corridas de
% un Algoritmo Genético Multipoblación.
%
% OUTPUT
% Para cada 'pruebaxxx' genera un directorio 'pruebaxxx'
% donde almacena un archivo 'Best.mat', el cual
% contiene las variables:
%
%      Best      Matric de 3*MAXGEN que contiene:
%      Best(:,1) Mejor evaluacion generación a generación
%      Best(:,2) Promedio de evaluación generación a generación
%      Best(:,3) Evaluaciones de la funcion cada generación
%
%      IndAll    Mejor Individuo Obtenido con el Algoritmo
%      gen       Número de generaciones
%      runtime   Tiempo de ejecución del algoritmo
%
% Se utilizan individuos en representación real.
%
% Parámetros del GA que se leen
% PPUEBA      Numero de la prueba actual
% INTERVALO   Intervalo de Muestreo en los histogramas WB (1 a 10)
% GGAP        Cuantos nuevos individuos se crean
% NOVER       Probabilidad de cruzamiento
% MUTR        Probabilidad de mutación
% MAXGEN      Máximo número de generaciones
% TERMEXACT   Valor de terminación si se encontró un mínimo
% INSR        Tasa deInserción rate, cuantos hijos se insertan
% SUBPOP      Número de subpoblaciones
% MIGR        Tasa de migración entre subpoblaciones
% MIGGEN      Número de generaciones entre migraciones
% NIND        Número de individuos por subpoblación
% SEL_F       Nombre de la función de selección
% XOY_F       Nombre de la función de cruzamiento
% MUT_F       Nombre de la función de mutación
% OBJ_F       Nombre de la función objetivo
```

```

% Código original para GA Toolbox
% Autor:      Andrew Chipperfield
% Institución: University of Sheffield
% Historia:   30-Mar-94      Archivo Creado

% Modificado: Marcos Iván Quintana
% Historia:   15-May-00      Creacion Meta AG
%            3-Ago-00      Documentación

clear;

%Lectura de los histogramas y de los tamaños
%Donde histos(i,:) es un vector de tamaño largo(i) pixels
%que representa una tira Western Blot con niveles de gris
%de 0 a 255

load('c:\publico\histos');

%Normalizar los valores de los histogramas de 0 a 1
histosg=ind2gray(histos,gray(255));
tamhisto=size(histosg);

%Iniciar el ciclo general
CONTINUAR=1;
PRUEBA_ACTUAL=0;
PRUEBA=0;

while CONTINUAR,

    %Leer Parámetros para el AG desde el archivo de entrada
    fid=fopen('c:\publico\paramet.txt');
        while PRUEBA_ACTUAL >= PRUEBA,
            line = fgetl(fid);
            eval(eval(line));
        end
    fclose(fid);
    PRUEBA_ACTUAL=PRUEBA;

    %Nombrar y Crear directorio de trabajo para el actual AG
    if PRUEBA < 10 nomdir=strcat('prueba00',int2str(PRUEBA));
    elseif PRUEBA < 100 nomdir=strcat('prueba0',int2str(PRUEBA));
    else nomdir=strcat('prueba',int2str(PRUEBA));
    end

    mkdir('c:\publico', nomdir);
    nomdir=strcat('c:\publico\', nomdir);

```

```

%Calcular el número de variables en base a INTERVALO
NVAR=ceil(tamhisto(2)/INTERVALO);

%Limpiar matrices de tamaño variable;
clear histosw FieldD Best IndAll Chrom ObjV FitnV SelCh ObjVoff;

%Ajustar la matriz de trabajo de histogramas a INTERVALO
histosw(1:201,:)=histosg(1:tamhisto(1),1:INTERVALO:tamhisto(2));

%Limites de las variables de decisión
RANGE = [0;1];

% Descriptor de campos
FieldD = rep(RANGE,[1,NVAR]);

% Inicializar matrices para almacenar los valores de aptitud
% mejores y promedio
Best = NaN * ones(MAXGEN,3);
Best(:,3) = zeros(size(Best,1),1);
% Matriz para almacenar los mejores individuos
IndAll = [];

% Crear población real
Chrom = crtrp(SUBPOP*NIND,FieldD);

% Reiniciar contador y tiempo
gen = 1;
t0 = clock;

% Calcular función objetivo a la población
ObjV = feval(OBJ_F,histosw,largo,Chrom);
% Contar el número de evaluaciones de la función objetivo
Best(gen+1,3) = Best(gen+1,3) + NIND;

% Inicia ciclo generacional
while gen <= MAXGEN,

% Guardar los valores de función objetivo mejor y promedio
% y también al mejor individuo
[Best(gen+1,1),ix] = min(ObjV);
Best(gen+1,2) = mean(ObjV);
IndAll = [IndAll; Chrom(ix,:)];

% Mostrar resultados para asegurar el avance y
% guardarlos en archivo
if (rem(gen,GSAVE) == 0)
runtime=etime(clock,t0);
save(strcat(nomdir,'\Best'),'Best','IndAll','gen','runtime');

```

```

        nomdir
        gen
    end

% Asignación de valor de aptitud
    FitnV = ranking(ObjV, [2 0], SUBPOP);

% Seleccionar individuos en la población
    SelCh = select(SEL_F, Chrom, FitnV, GGAP, SUBPOP);

% Recombinar individuos seleccionados
    SelCh=recombin(XOV_F, SelCh, XOVR, SUBPOP);

% Mutar hijos
    SelCh=mutate(MUT_F, SelCh, FieldD, [MUTR], SUBPOP);

% Calcular función objetivo en los hijos
    ObjVOff = feval(OBJ_F,histosw,largo,SelCh);
    Best(gen+1,3) = Best(gen+1,3) + size(SelCh,1);

% Insertar a los mejores hijos sustituyendo a los peores padres
    [Chrom, ObjV] = reins(Chrom, SelCh, SUBPOP, [1 INSR], ObjV, ObjVOff);

    gen=gen+1;

% Migrar individuos entre subpoblaciones
    if (rem(gen,MIGGEN) == 0)
        [Chrom, ObjV] = migrate(Chrom, SUBPOP, [MIGR, 1, 0], ObjV);
    end

% fin de ciclo generacional
end
% fin de ciclo de pruebas

end

% Fin de archivo

```

\*

## FUNCIONES OBJETIVO

### Función 01

```
% Apéndice A
% Funcion01
%
% INPUT
%   Histos   Matriz con los histogramas de cada imagen WB
%   Largos   Vector con los tamaños de cada histograma
%   Chrom    Matriz de valores reales de soluciones posibles
%
% OUTPUT
%   ObjVal   Vector de evaluación a minimizar

function ObjVal = funcion01(Histos, Largos, Chrom);

% Calcular parametros de población
%   Nind     Numero de Individuos de la Población
%   Nvar     Numero de variables
%   NThist   Número total de hisogramas recibidos
%   Max1     Largo máximo de histograma

[Nind, Nvar] = size(Chrom);
[NThist,Max1] = size(Histos);

% Número de individuos con NC
Nhist = 49;

% Ciclo para calcular ObjVal como la suma del error
% de cada individuo contra todos los histogramas de NC
for i=1:Nind,
    ObjVal(i,1)=0;
    for j=1:Nhist,
        ObjVal(i,1)=ObjVal(i,1)+sum(abs(Chrom(i,:)-Histos(j,:)));
    end
end;
end;
```



## Función 02

```
% Apéndice A
% Funcion02
%
% Función de Evaluación para pares de Imágenes WB
%
% Esta función retorna un valor de evaluación
% normalizado entre 0 y 1.
%
% La primera mitad de Chrom debe parecerse a los
% primeros Nhist histogramas y la segunda mitad
% de Chrom debe parecerse a los siguientes histogramas.
%
% Mientras mejor se cumpla este criterio, el valor de
% ObjVal(i,1) debe aproximarse más a 0.
%
% INPUT
%   Histos   Matriz con los histogramas de cada imagen WB
%   Largos   Vector con los tamaños de cada histograma
%   Chrom    Matriz de valores reales de soluciones posibles
%
% OUTPUT
%   ObjVal   Vector de evaluación a minimizar

function ObjVal = funcion06(Histos, Largos, Chrom);

% Calcular parametros de población
[Nind, Nvar] = size(Chrom);

% Muestrear histograma al tamaño de Chrom
%Histos=Histos(:,1:2:Nvar);
[NThist,Max1] = size(Histos);

%En que proporción deseamos impactar NV vs OE
%Proporción es PropMay:PropMen
PropMay=1;
PropMen=1;

% Numero de individuos con NC
Nhist = 49;

% Ciclo para calcular ObjVal como la suma del error medio cuadrático
% de cada individuo contra todos los histogramas de NC

for i=1:Nind,
    ObjVal(i,1)=0;
```

```

for j=1:NThist,
    DiferNC=sum(abs(Chrom(i,1:(Nvar/2)) -Histos(j,:)));
    DiferOE=sum(abs(Chrom(i,((Nvar/2)+1):Nvar)-Histos(j,:)));
    if j>Nhist,
        if DiferOE>DiferNC,
            ObjVal(i,1)=ObjVal(i,1)+PropMen;
        end;
    else
        if DiferNC>DiferOE,
            ObjVal(i,1)=ObjVal(i,1)+PropMay;
        end;
    end; %fin if anidado
end; %fin for j
ObjVal(i,1)=ObjVal(i,1)/((PropMen*(NThist-Nhist))+(PropMay*Nhist));
end; %fin for i

%fin de la función
end

```

## PARÁMETROS

Archivo de parámetros de las pruebas 1 a las 60 utilizadas en los capítulos 2 y 3.

```
'PRUEBA=001; INTERVALO =10; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 10000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=002; INTERVALO = 9; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 10000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=003; INTERVALO = 8; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 10000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=004; INTERVALO = 7; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=005; INTERVALO = 6; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=006; INTERVALO = 5; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=007; INTERVALO = 4; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=008; INTERVALO = 3; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=009; INTERVALO = 2; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=010; INTERVALO = 1; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'reccdis'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=011; INTERVALO =10; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'recint'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=012; INTERVALO = 9; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'recint'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
'PRUEBA=013; INTERVALO = 8; GGAP = .9; XOVR = 1; MUTR = 0.45; MAXGEN = 2000; TERMEXACT =
1e2; INSR = .9; SUBPOP = 4; MIGR = 0.2; MIGGEN = 20; NIND = 250; GSAVE =20; SEL_F = 'sus';
XOV_F = 'recint'; MUT_F = 'mutbga'; DOBLE=0; OBJ_F = 'funcion01';'
```







# APÉNDICE B: PROGRAMAS Y ARCHIVOS PARA PROGRAMACIÓN GENÉTICA

## PROGRAMA 1

```
% Apéndice B
% Programa 1
%
% Metawhqp      Meta Algoritmo para programación Genética
%               e imágenes Western Blot
%
% Código original para GA Toolbox
% Autor:       Andrew Chipperfield
% Institución: University of Sheffield
% Historia:    30-Mar-94      Archivo Creado
%
% Modificado:  Marcos Iván Quintana
% Historia:    15-May-00      Creación Meta AG
%             23-Ago-00      Adaptación para Meta GP
%             01-Sep-00      Documentación

clear;

%Lectura de los histogramas y de los tamaños
%Donde histos(i,:) es un vector de tamaño largo(i) pixels
%que representa una tira Western Blot con niveles de gris
%de 0 a 255

load('c:\publico\histos');

%Normalizar los valores de los histogramas de 0 a 1
histosg=ind2gray(histos,gray(255));
tamhisto=size(histosg);

%Funciones, Terminales
Funciones=[1,2,3,4,5,6,7,8]; % +,-,*,/,>,<,>=,<=
Terminales=[11,12];        % x R
Aridad=[2,2,2,2,2,2,2,2]; % Aridad Homogénea
Ad=[31]; %nada
Ada=[1]; %nada

%Iniciar el ciclo general
CONTINUAR=1;
PRUEBA_ACTUAL=0;
PRUEBA=0;
```

```

SUBPOP=1;
INSR=0.9;

while CONTINUAR,
    %Leer Parámetros del GP del archivo de entrada
    fid=fopen('c:\publico\paramet.txt');
    while PRUEBA_ACTUAL >= PRUEBA,
        line = fgetl(fid);
        eval(eval(line));
    end
    fclose(fid);
    PRUEBA_ACTUAL=PRUEBA;

    % Parámetros del GP que se han leído
    % PRUEBA      Número de la prueba actual
    % INTERVALO  Intervalo de Muestreo (1 a 10)
    % MAXNODE    Máximo número de nodos
    % GGAP       CGeneration gap, how many new individuals are created
    % GSAVE      Cada cuando salvar resultados
    % CRUZA      Crossover rate
    % MUTA       Mutation rate
    % MAXGEN     Maximum number of generations
    % NIND       Number of individuals per subpopulation
    % F_EVAL     Función de Evaluación

    %Nombrar y Crear directorio de trabajo para el actual AG
    if PRUEBA < 10 nomdir=strcat('prueba00',int2str(PRUEBA));
    elseif PRUEBA < 100 nomdir=strcat('prueba0',int2str(PRUEBA));
    else nomdir=strcat('prueba',int2str(PRUEBA));
    end

    mkdir('c:\publico', nomdir);
    nomdir=strcat('c:\publico\', nomdir);

    %Calcular el número de variables en base a INTERVALO
    NVAR=ceil(tamhisto(2)/INTERVALO);

    %Limpiar matrices de tamaño variable;
    clear histosw FieldD Best IndAll Chrom ObjV FitnV SelCh ObjVOff;

    %Ajustar la matriz de trabajo de histogramas a INTERVALO
    histosw(1:201,:)=histosg(1:tamhisto(1),1:INTERVALO:tamhisto(2));

    % Inicializar matrices para almacenar los valores de aptitud
    % mejores y promedio
    Best = NaN * ones(MAXGEN,3);

```



```

Best(:,3) = zeros(size(Best,1),1);

% Matriz para almacenar los mejores individuos
IndAll = [];

% Crear población de árboles en forma de vector y notación polaca
Chrom = crtrees(NIND, MAXNODE, Funciones, Aridad, Terminales, Ad, Ada);
Chrom = ((Chrom==12).*rand(size(Chrom))*0.9999)+(Chrom.*xor(Chrom==12,Chrom));

% Reiniciar contador y tiempo
gen = 1;
t0 = clock;

% Calcular función objetivo a la población
ObjV = feval(OBJ_F,Chrom,histosw);

% Contar el número de evaluaciones de la función objetivo
Best(gen+1,3) = Best(gen+1,3) + NIND;

% Inicia ciclo generacional
while gen <= MAXGEN,

    % Guardar los valores de función objetivo mejor y promedio
    % y también al mejor individuo
    [Best(gen+1,1),ix] = min(ObjV);
    Best(gen+1,2) = mean(ObjV);
    IndAll = [IndAll; Chrom(ix,:)];

    % Mostrar resultados para asegurar el avance y
    % guardarlos en archivo
    if (rem(gen,GSAVE) == 0)
        runtime=etime(clock,t0);
        save(strcat(nomdir,'\Bestgp'),'Best','IndAll','gen','runtime');
        nomdir
        gen
    end

    % Fitness assignment to whole population
    FitnV = ranking(ObjV,NaN,SUBPOP);

    % Select individuals from population
    SelCh = select('sus', Chrom, FitnV, GGAP, SUBPOP);

    % Recombinar individuos seleccionados
    SelCh = xvtrees(SelCh,CRUZA,Funciones,Aridad,Ad);

    % Mutar hijos
    SelCh = muttrees(SelCh,MUTA,Funciones,Terminales,Aridad,Ad,Ada,MAXNODE);

```

```

% Calculate objective function for offsprings
ObjVOff = feval(OBJ_F, SelCh, histosw);
Best(gen+1,3) = Best(gen+1,3) + size(SelCh,1);

% Insert best offspring in population replacing worst parents
[Chrom, ObjV] = reins(Chrom, SelCh, SUBPOP, [1 INSR], ObjV, ObjVOff);

gen=gen+1;

end
% fin de ciclo generacional

end
% fin de ciclo de pruebas

% Fin de archivo

```

## FUNCIÓN OBJETIVO

```

% Apéndice B
% funcion01 GP
%
% Funcion de Evaluación para Imagenes WB
% Utiliza estructuras de árbol para Programación Genética
%
% INPUT
% Chrom      Matriz que contiene estructuras
%            de Arbol representadas en vectores
% Histos     Matriz que contiene los histogramas
%            que representan la imágenes WB
%
% OUTPUT
% ObjVal     Vector con la evaluación de la función
%            a minimizar.
%
% ObjVal(i,1) puede tomar valores entre 0 y 1
% Se busca minimizar

function ObjVal = gpwbfun1(Chrom, histos);

% Calcular parametros de población
[Nind, Nvar] = size(Chrom);
[NThist, Max1] = size(histos);

%Calcular Número de Nodos
NNodos=sum(Chrom'~=0);

```

```

*Funciones y Terminales
Tam=[4 5 5 8 2 2 2 2 0 0 1];
FT(1,1:Tam(1))='plus'; % +
FT(2,1:Tam(2))='minus'; % -
FT(3,1:Tam(3))='times'; % *
FT(4,1:Tam(4))='spdivide'; %Division protegida
FT(5,1:Tam(5))='gt'; % >
FT(6,1:Tam(6))='lt'; % <
FT(7,1:Tam(7))='ge'; % >=
FT(8,1:Tam(8))='le'; % <=
FT(11,1:Tam(11))='x';

*Inicializar Pila e Individuos de GP
pila(5,1:40)=blanks(40);
IndiGP(Nind,1:40)=blanks(40);

*Este ciclo realiza la conversión de todos los árboles contenidos en Chrom
*A notación polaca comprendida por Matlab con la función "eval"
for i=1:Nind,
    cadena='';
    idx=0; B=0;
    for j=1:NNodos(i),
        k=NNodos(i)-j+1;
        if Chrom(i,k)<1
            idx=idx+1;
            [A B]=size(num2str(Chrom(i,k)));
            pila(idx,1:B)=num2str(Chrom(i,k));
            TamP(idx)=B;
            %Para el caso de que sea de sólo un individuo
            cadena=strcat('eval(''',num2str(Chrom(i,k)),''')');
            [A B]=size(cadena);
        elseif Chrom(i,k)<10, %Si es una función
            cadena=strcat('feval(''',...
                FT(Chrom(i,k),1:Tam(Chrom(i,k))),''',',...
                pila(idx,1:TamP(idx)),',',...
                pila(idx-1,1:TamP(idx-1)),')');
            [A B]=size(cadena);
            pila(idx-1,1:B)=cadena;
            TamP(idx-1)=B;
            idx=idx-1;
        else %Si es un terminal
            idx=idx+1;
            pila(idx,1:Tam(Chrom(i,k)))=FT(Chrom(i,k),1:Tam(Chrom(i,k)));
            TamP(idx)=Tam(Chrom(i,k));
            %Para el caso de que sea de sólo un individuo
            cadena=strcat('eval(''',...
                FT(Chrom(i,k),1:Tam(Chrom(i,k))),''')');
            [A B]=size(cadena);

```

```

        end
    end
    IndiGP(i,1:B)=cadena;
end

x=histos;
Nhist = 49;
NUM_UMB=200;

clear A B Chrom FT Tam Tamp cadena histos i idx j k largo pila;

%
%En este ciclo se realiza la evaluacion de cada individuo
%y la comparación con los umbrales para determinar la mejor evaluación
for i=1:Nind,
    y=eval(IndiGP(i,:));
    diferencias=sum(abs(y-x)');
    umbrales=linspace(min(diferencias),max(diferencias),NUM_UMB);
    [A B]=size(umbrales);
    if B~=NUM_UMB,
        errores=NThist;
    else
        for k=1:NUM_UMB,
            errores(k)=sum(diferencias(1:Nhist)<=umbrales(k))+...
                sum(diferencias(Nhist+1:NThist)>umbrales(k));
        end
    end
    ObjVal(i,1)=min(errores)/NThist;
end;

```

## FUNCIONES PARA CREACIÓN DE POBLACIÓN

```

% CRTREEP.M
% This function creates a tree-matrix of given size whose elements are random
% operators (from the Function_Set) and operands (from the Terminal_Set).
% Syntax:      [TreeP, NNodes] = crtrees(Nind, MaxNod, Function_Set,
%                                     Function-Arity, Terminal_Set, AIDF_Set,
%                                     ADF-Arity);
% This function produces a initial population of hierarchical tree expressions.
% An individual tree is represented in a vector form by means of its equivalent
% inverse polish notation. Each tree expression is composed of functions
% (internal nodes of the tree) from the pre-defined Function_Set, and operands
% (terminal or leaf nodes) taken from the Terminal_Set. This CRTREEP function
% also calls the routine TRE@EXP which produces valid single tree expressions.
% This function returns a matrix "TreeP" and a vector "NNodes" (optional) which
% contains the size (number of nodes) of each individual tree.
% Input Parameters:
%      Nind      - Number of individuals (rows).
%      MaxNod    - Maximum number of nodes considered for the tree

```

```

% expressions.
% Function-Set - Set of operations which represent internal nodes
% elements.
% Function-Arity - Set that contains the number of arguments taking-for
% each function of the Function-Set. The Function-Set
% and the Function-Arity have to be row vectors of the
% same size.
% Terminal-Set - Set of operands which represent the leaf nodes.
% ADF-Set - Set of Automatically Defined Functions (see Reference)
% ADF-Arity - Contains the number of arguments taking for each ADF.
% Output Parameters:
% TreeP - NInd x MaxNod matrix containing the initial population
% of tree expressions.
% NNodes - [optional] NInd vector containing the size (number of
% nodes) of each tree expression.
% DATE: 27.02.97.
% Author: Katya Rodriguez-Vazquez
% Evolutionary Computing Lab.
% Dept. of Automatic Control and Systems Engineering
% University of Sheffield

```

```

function [TreeP, NNodes] = crtrep(NInd, MaxNod, Function_Set, ...
    Function_Arity, Terminal_Set, ADF_Set, ...
    ADF_Arity);

TreeP = zeros(NInd, MaxNod);
NNodes = zeros(NInd, 1);

for i=1:NInd,
    [TreeP(i,:), NNodes(i)] = treexp(MaxNod, Function_Set, ...
        Function_Arity, Terminal_Set, ADF_Set, ADF_Arity);
end

```

```

% TREEXP.M
% This function creates a tree structure of given size whose elements are
% operators (from the Function-Set) and operands (from the Terminal Set)
% Syntax: [Tree, NN] = treexp(MaxNod, Function_Set, Function_Arity,
% Terminal_Set, ADF_Set, PT);
% This function produces a single tree expressions represented in a vector
% form by means of its equivalent polish notation. this expression is composed
% of functions (internal nodes of the tree) from the pre-defined Function-set,
% and operands (terminal or leaf nodes) taken from the Terminal Set. This
% function returns a vector "Tree" and a scalar NN which contains
% the size (number of nodes) of the single tree
% Input Parameters:
% MaxNod ~ Maximum number of nodes considered for the
% expressions.

```

```

%      Fuction-Set      - Set of operations which represent internal nodes
%                      elements.
%
%      Functin-Arity   - Set that contains the number of arguments taking for
%                      each function of the Function set. The function set
%                      and the Function_Arity have to be row vectors of the
%                      same size.
%
%      Terminal-Set    - Set of operands which represents the leaf nodes
%
%      ADF-Set         - Set of Automatically defined Functions
%
%      ADF-Arity       - Contains the number of arguments taking for each ADF
%
%      PT              - ???
%
%
% Output Parameters:
%
%      Tree            ~ NInd x MaxNod matrix containing the initial population of
%                      of tree expressions
%
%      NN              - [optionall NInd vector containing the size (number of
%                      nodes) of each tree expressions
%
% DATE: 27.02.97.
% Author: Katya Rodriguez-Vazquez
%         Evolutionary Computing Lab.
%         Dept. of Automatic Control and Systems
%         University of Sheffield

```

```

function [Tree, NN] = treexp(MaxNod, Function_Set, Function_Arity, ...
    Terminal_Set, ADF_Set, ADF_Arity, PT);

```

```

if nargin<7,
    PT = [];
end

```

```

NF = size(Function_Set, 2);
NT = size(Terminal_Set, 2);
NA = size(ADF_Set, 2);

```

```

dangling_limbs = 1;
i = 1;

```

```

while ((dangling_limbs > 0) & (i <= MaxNod))

```

```

    %Choose function or terminal
    if (rand > ((dangling_limbs*dangling_limbs) + 1)/(MaxNod - i)),
        index = ceil(NF * rand);
        if any(ADF_Set == Function_Set(index)),
            Tree(i) = Function_Set(index);
            if isempty(PT),
                Tree(i+1:i+Function_Arity(index)) = Terminal_Set ...
                    (ceil(NT*rand(1, Function_Arity(index))));
            else
                Tree(i+1:i+Function_Arity(index)) = Terminal_Set ...

```

```

                (sus(PT, Function_Arity(index)));
        end
        i = i + 1 + Function_Arity(index);
        dangling_limbs = dangling_limbs - 1;
    else
        Tree(i) = Function_Set(index);
        dangling_limbs = dangling_limbs + (Function_Arity ...
            (index)-1);
        i = i + 1;
    end
else
    if isempty(PT),
        Tree(i) = Terminal_Set(ceil(NT * rand));
    else
        Tree(i) = Terminal_Set(sus(PT, 1));
    end
    dangling_limbs = dangling_limbs - 1;
    i = i + 1;
end
end
end

Tree(i: MaxNod) = zeros(1, MaxNod-i+1);
NN = i-1;

```

## FUNCIONES PARA MODIFICAR ÁRBOLES

### Cruzamiento

```

% NVTREE.M
% This function performn the crossover operation between pairs of individual
% trees of the present population (OldPop).
% Syntax:      NewPop = xvtree(oldpop, Pcross, Function-Set, Function-Arity,
%                    ADF-Set);
% XVTREE produces a new population (offsprings) by recombining pairs of tree
% individuals of the present generation. This process is carried out by
% selecting a random node in each parent tree and exchanging their associated
% subexpression.
% The population of tree expressions is representad in a matrix where each row
% is a single tree individual.
% Input Parameters:
%   oldpop      - NInd x MaxNod matrix containig the present polulation
%                (before crossing over).
%   Pcross      - Probability of crossover occurring between two trees.
%                expressions.
%   Fuction-Set - Set of operations which represent internar nodes
%                elements.
%   Functin-Arity - Set that contains the number of arguments taking for

```

```

% each function of the Function-Set. The Function-Set
% and the Function-Arity have to be row vectors of the
% same size.
% ADF-Set - Set of Automatically Defined Functions (see Reference)
% Output Parameters:
% NewPop - NInd x MaxNod matrix containing the new individuals
% after crossing over, in the same format as OldPop.
% DATE: 27.02.97.
% Author: . Katya Rodriguez-Vasquez
% Evolutionary Computing Lab.
% Dept. of Automatic Control and Systems Engineering
% University of Sheffield

```

```

% Crossover operator for tree encoding.

```

```

function NewPop = xvtree(OldPop, Pcross, Function_Set, Function_Arity, ADF_Set);

```

```

[NInd, MaxNod] = size(OldPop);

```

```

Xops = floor(NInd/2);

```

```

odd = 1:2:NInd-1;

```

```

even = 2:2:NInd;

```

```

N_Nodes = sum(OldPop(1:NInd,:) & ones(NInd, MaxNod)');

```

```

NewPop = zeros(NInd, MaxNod);

```

```

for i=1:Xops,

```

```

    mum1 = MaxNod+1;

```

```

    dad1 = MaxNod+1;

```

```

    while ((mum1 >=(MaxNod+1)) | (dad1 >=(MaxNod+1))),

```

```

        mum_end1 = ceil(rand * (N_Nodes(odd(i))));

```

```

        [mum_start2, mum_end1] = valtree(OldPop(odd(i),:), ...

```

```

            mum_end1, Function_Set, Function_Arity, ADF_Set);

```

```

        mum_end2 = N_Nodes(odd(i));

```

```

        dad_start = ceil(rand * (N_Nodes(even(i))));

```

```

        [dad_end, dad_start] = valtree(OldPop(even(i),:), ...

```

```

            dad_start, Function_Set, Function_Arity, ADF_Set);

```

```

        mum1=mum_end1 + (dad_end - dad_start) + (mum_end2 - mum_start2);

```

```

        dad1=dad_start+(mum_start2 - mum_end1)+(N_Nodes(even(i)) ...

```

```

            - dad_end);

```

```

    end

```

```

    if (mum_end2 == mum_end1) | (dad_start == N_Nodes(even(i))),

```

```

        NewPop(odd(i),:) = OldPop(odd(i),:);

```

```

        NewPop(even(i),:) = OldPop(even(i),:);

```

```

    else

```

```

        NewPop(odd(i), 1:mum1) = [OldPop(odd(i), 1:mum_end1-1) ...

```

```

            OldPop(even(i), dad_start:dad_end) ...

```



```

        OldPop(odd(i), mum_start2+1:mum_end2)];
NewPop(even(i),1:dad1) = [OldPop(even(i),1:dad_start-1) ...
        OldPop(odd(i),mum_end1:mum_start2) ...
        OldPop(even(i), dad_end+1: N_Nodes(even(i)))]];
end
end

% If the number of individual is odd, the last individual cannot be mated
% but must be included in the new population

if rem(NInd,2),
    NewPop(NInd,:) = OldPop(NInd,:);
end

```

## Mutación

```

% MUTREE.M
% This function takes the current population and mutates each of its tree
% individuals with a low mutation rate.
% Syntax-      NewPop = =tree(OldPop, Pmut, Function-Set, Te=inal_Set,
%
%               Function-Arity, ADF_Set, ADF-Arity, Max=);
% MUTREE produces a new population (offsprings) by mutating each individual of
% the current tree population. This process is performed by randomly selecting
% a node, that can be internar or terminal, and replacing the associated subtree
% expression with a randomly generated subtree. TREEXP routine is called to
% generated this subtree.
% The population of tree exp'ressions is representad in a matrix where each row
% is a single tree individual.
% Input Parameters:
%
%      OldPop      -  NInd x MaxNod matrix containig the present polulation
%                    (before crossing over).
%
%      Pmut        -  mutation probability.
%
%      Fuction-Set -  Set of operations which represent internar nodes
%                    elements.
%
%      Function-Arity - Set that contains the number of arguments taking for
%                    each function of the Function-Set. The Function-Set
%                    and the Function-Arity have to be row vectors of the
%                    same size.
%
%      Terminal-Set - Set of operands which represent the leaf nodes.
%
%      ADF-Set     -  set of Automatically Defined Functions (see Reference)
%
%      ADF-Arity   -  Contains the number of arguments taking for each ADF.
%
%      MaxNM       -  Maxim= number of nodes allowed to generated a subtree
%                    expression.
%
% Output Parameters:
%
%      NewPop      -  Nlnd x MaxNod matrix containing the new individuals

```

```

%                               after crossing over, in the same format as OldPop.
% DATE: 27.02.97.
% Author: Katya Rodriguez-Vazquez
%           Evolutionary Computing Lab.
%           Dept. of Automatic Control and Systems Engineering
%           University of Sheffield
% Mutation Routine: MUTREE.M

function NewPop = mutree(OldPop, Pmut, Function_Set, Terminal_Set, ...
    Function_Arity, ADF_Set, ADF_Arity, MaxDM);

[NInd, MaxNod] = size(OldPop);
NewPop = zeros(NInd, MaxNod);
NF = size(Function_Set, 2);

NNodes = sum(OldPop(1:NInd, :) & ones(NInd, MaxNod)');

for i=1:NInd,
    if (rand < Pmut),
        nlength = MaxNod + 2;
        while ((nlength >= (MaxNod+1)) | (rem(nlength,2) == 0)),
            end1 = ceil(rand * NNodes(i));
            [start2, end1] = valtree(OldPop(i, :), end1, ...
                Function_Set, Function_Arity, ADF_Set);

            end2 = NNodes(i);

            [SubTree, mut_length] = treexp(MaxDM, Function_Set, ...
                Function_Arity, Terminal_Set, ADF_Set, ADF_Arity);
            nlength = (end1 - 1) + mut_length + (end2 - start2);

        end

        if (end1 == end2),
            NewPop(i,:) = OldPop(i,:);
        else
            NewPop(i,:) = [OldPop(i, 1:end1-1) SubTree(1: ...
                mut_length) OldPop(i, start2+1: end2) ...
                zeros(1, MaxNod-nlength)];
        end
    else
        NewPop(i,:) = OldPop(i,:);
    end
end
end

```



'PRUEBA=024; INTERVALO = 3; MAXNODE = 64; GGAP = 1; CRUZA = 0.75; MUTA = 0.005; MAXGEN = 100; NIND = 150; GSAVE =5; OBJ\_F = 'gpwbfun1';'  
'PRUEBA=025; INTERVALO = 9; MAXNODE = 128; GGAP = 1; CRUZA = 0.75; MUTA = 0.005; MAXGEN = 100; NIND = 150; GSAVE =5; OBJ\_F = 'gpwbfun1';'  
'PRUEBA=026; INTERVALO = 6; MAXNODE = 128; GGAP = 1; CRUZA = 0.75; MUTA = 0.005; MAXGEN = 100; NIND = 150; GSAVE =5; OBJ\_F = 'gpwbfun1';'  
'PRUEBA=027; INTERVALO = 3; MAXNODE = 128; GGAP = 1; CRUZA = 0.75; MUTA = 0.005; MAXGEN = 100; NIND = 150; GSAVE =5; OBJ\_F = 'gpwbfun1';'