

P
R
O
M
O
C
I
Ó
N

1
9
9
8

UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO

ESCUELA DE POST GRADO

MAESTRÍA EN INFORMÁTICA



**"EXPERIMENTACIÓN DE INGENIERÍA GENÉTICA
EN EL FUNCIONAMIENTO DE ALGORITMOS GENÉTICOS
RESULTADOS Y EVALUACIÓN EN OPTIMIZACIÓN
FUNCIONAL"**

T E S I S

**PARA OPTAR EL GRADO DE:
MAGISTER SCIENTIAE
EN INFORMÁTICA**

PRESENTADA POR:

Bach. ERNESTO NAYER TUMI FIGUEROA

01

PUNO - PERÚ DEL 2001

UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO

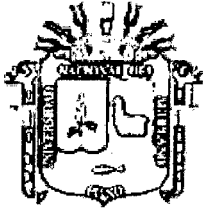
BIBLIOTECA CENTRAL

Fecha Ingreso:

01 OCT. 2012

N°

00147



UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POS GRADO
MAESTRIA EN INFORMATICA

TESIS DE MAESTRIA

“EXPERIMENTACIÓN DE INGENIERIA GENETICA EN
EL FUNCIONAMIENTO DE ALGORITMOS GENETICOS
RESULTADOS Y EVALUACIÓN EN OPTIMIZACION
FUNCIONAL”

Presentado por:
Ernesto Nayer Tumi Figueroa
ntumi@correoweb.com

Director: M.Sc. Víctor Bustinza Choque
Asesor: M. Sc. Octavio Morillos Valderrama

Área de Investigación
Inteligencia Computacional

Puno, C.U. Junio del 2001


UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POST GRADO
MAESTRIA EN INFORMATICA

TESIS

“EXPERIMENTACIÓN DE INGENIERIA GENETICA EN EL
FUNCIONAMIENTO DE ALGORITMOS GENETICOS
RESULTADOS Y EVALUACIÓN EN OPTIMIZACION
FUNCIONAL”

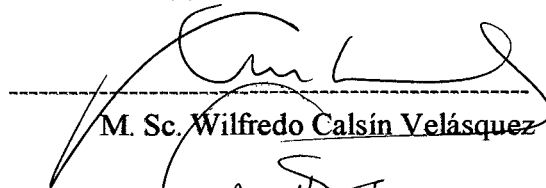
JURADOS

PRESIDENTE




Ph.D(c) Lucio Avila Rojas

PRIMER MIEBRO



M. Sc. Wilfredo Calsín Velásquez

SEGUNDO MIEMBRO

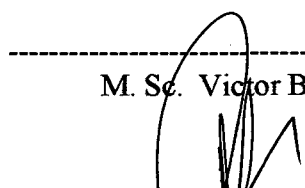


M. Sc. Luis Huarachi Coila

DIRECTOR

M. Sc. Victor Bustinza Choque

ASESOR



M. Sc. Octavio Morillos Valderrama

CONTENIDOS

RESUMEN	1
II. METODO DE LA INVESTIGACION	3
2.1 PLANTEAMIENTO Y DEFINICIÓN DEL PROBLEMA	3
2.2 OBJETIVO GENERAL	3
2.3 OBJETIVOS ESPECIFICOS	3
2.4 HIPOTESIS	4
2.5 JUSTIFICACION E IMPORTANCIA	4
2.6 REVISION DE LITERATURA	5
2.6.1 PRESENTACION DE LOS ALGORITMOS GENETICOS	5
2.6.2 AVANCES DE INVESTIGACIÓN Y RECONOCIIENTO DE HECHOS	12
2.6.3 SISTEMA DE CONCEPTOS	16
2.5.4 LEYES ACERCA DEL PROBLEMA Y EL METODO	25
2.6.5 MARCO OPERATIVO	34
III. RESULTADOS Y DISCUSIÓN	48
3.1 IDENTIFICACION DE LAS FUNCIONES DE EVALUACIÓN	48
3.2 DESCRIPCION DEL ALGORITMO GENETICO CON OPERADOR DE CLONACION Y MUTACIÓN	51
3.3 IMPLEMENTACION DEL ALGORITMO GENETICO	54
3.4 RESULTADOS EXPERIMENTALES DEL ALGORITMO GENETICO CANONICO EN OPTIMIZACION FUNCIONAL	58

3.5 RESULTADOS CON LA INCORPORACIÓN DE LOS OPERADORES DE CLONACION Y MUTACIÓN Y LA INCORPORACIÓN DE LOS PRINCIPIOS DE INGENIERIA GENETICA EN EL ALGORITMO GENETICO CANONICO.	62
3.6 COMPARACION DE APLICACIÓN DEL PROCESO DE INGENIERIA GENETICA EN OTRAS POBLACIONES PARA SU ANÁLISIS	65
3.7 RESULTADOS DE APLICACIÓN DEL OPERADOR DE MUTACIÓN MÚLTIPLE	70
3.8 RESULTADOS DE UN CASO DE UNA POBLACIÓN TÍPICA CON APLICACIÓN DE INGENIERIA GENETICA	74
3.9 RESULTADOS DE APLICACIÓN DE LOS PRINCIPIOS DE INGENIERIA GENETICA AL CASO DE UNA FUNCIÓN DE DOS VARIABLES INDEPENDIENTES.	78
IV. CONCLUSIONES	81
V. RECOMENDACIONES	82
VI. BIBLIOGRAFÍA CITADA	83
ANEXO A CODIGO FUENTE	
ANEXO B SALIDA DEL PROCESO EVOLUTIVO	

A mi familia

RECONOCIMIENTO

Mi eterno reconocimiento a mi Universidad Nacional del Altiplano, fue la que me dio los fundamentos del conocimiento y fueron en sus aulas en que programe mis primeras líneas de código en el legendario Sinclair - Comodore. Bajo la penumbra fría de un aula que también era laboratorio y oficina del centro de cómputo.

Mi agradecimiento a mis profesores del Post Grado

Mi reconocimiento a mis estudiantes a los que debo mi voluntad por continuar esta brega de lograr navegar en el enorme y profundo mar del conocimiento, por los que espero seguir alcanzando mayores logros y metas.

RESUMEN

En el presente trabajo de investigación mostramos a los Algoritmos Genéticos como la confirmación matemática del inevitable proceso evolutivo y el papel que tiene que ver relacionado con la Computación Evolutiva. El proceso evolutivo es algo dinámico. La teoría de la selección de las especies sostiene que aquellos individuos de una población que posean los caracteres más ventajosos dejarán proporcionalmente más descendencia en la siguiente generación; y si tales caracteres se deben a diferencias genéticas, que pueden transmitirse a los descendientes, tenderá a cambiar la composición genética de la población, aumentando el número de individuos con dichas características. De esta forma, la población completa de seres vivos se adapta a las circunstancias variables de su entorno. El resultado final es que los seres vivos tienden a perfeccionarse en relación con las circunstancias que los envuelven. Por lo que la población cambia (evoluciona) hacia la figura del más fuerte.

Por otro lado, actualmente la revolución de la Ingeniería Genética llega a la utilidad práctica en forma de nuevas terapias, prácticas preventivas, diagnósticos, medicamentos y estrategias de trasplantes de órganos; los nuevos alimentos se asoman al mercado, asimismo se presentan los resultados de incorporar estos principios en el desarrollo de la informática especialmente en el área de la Inteligencia Computacional en todo esto esta presente el denominado "CLON" aquel organismo que es idéntico, en genotipo y fenotipo a otro sea o no su progenitor. Aunque a los ojos de la opinión pública pueda resultar un fenómeno extraño, los organismos clónicos abundan en la naturaleza, especialmente en el mundo microbiano, donde se dan formas de reproducción que persiguen la generación de una descendencia con un contenido genético idéntico al de sus progenitores. Asimismo también presentamos al operador de "MUTACIÓN" que implica la alteración aleatoria del material genético dando nuevas características tanto en su composición y estructura, a través de los genes, los que después de ser manipulados generan descendientes con características notorias que los distinguen de los demás normales individuos en los que en el trabajo hemos mencionado posibles soluciones.

Se ha experimentado y observado el desempeño de los Algoritmos Genéticos, a los que se les ha incorporado la tecnología de la Ingeniería Genética utilizando para ello exclusivamente los operadores de Mutación y Clonación. Se ha experimentado la aplicación de la Mutación Simple y la Mutación Múltiple como operadores principales en la aplicación de los principios de ingeniería genética en los algoritmos genéticos primeramente se ha estudiado todo lo relacionado con los Algoritmos Genéticos, luego se aplicó en dos funciones de evaluación las que han sido seleccionadas del estudio realizado por Zbigniew Michalewicz que describe en su texto " Genetic Algorithms + Data Structures = Evolution Programs ", luego de probar el funcionamiento del algoritmo genético simple canónico en estas funciones , hemos utilizado las mismas funciones de evaluación para ver el funcionamiento del AG con la incorporación de los principios de la Ingeniería Genética a través de los operadores de clonación y mutación encontrándose una influencia significativa en su funcionamiento y la forma como trabajan, así también se encontró que el operador de mutación múltiple es el mejor y con su aplicación se ha mejorado y superado la búsqueda genética al contrastarse con los resultados encontrados por Michalewicz, quien utilizó el algoritmo genético.

II. METODO DE LA INVESTIGACIÓN

2.1 PLANEAMIENTO Y DEFINICIÓN DEL PROBLEMA

Actualmente en la Universidad Nacional del Altiplano los temas de investigación en informática y computación están dando comienzo a su desarrollo a través de la maestría en informática y las carreras profesionales afines, los Algoritmos Genéticos son un área de bastante interés que empieza a aparecer con grandes expectativas, debido a su carácter de ser robusto e independiente de su aplicación y de ser genérico en la búsqueda de soluciones inspirado en la teoría evolucionista, se tiene que adaptar de manera particular a cada problema en particular por ello que actualmente se presenta muchas variantes y las investigaciones se ahondan en encontrar las mejores formas de funcionamiento, es por ello que con criterio de ampliar el espectro del funcionamiento de los Algoritmos Genéticos se ha experimentado y evaluado su rendimiento utilizando un Algoritmo Genético al que se le ha incorporando los principios de la Ingeniería Genética, para lograr esto se ha utilizado el operador CLONACIÓN junto con el de MUTACIÓN. Por lo que pretendemos responder la pregunta

¿ Que resultados se obtendrán en una implementación de los principios de la Ingeniería Genética en el funcionamiento de los Algoritmos Genéticos utilizando para ello los operadores de clonación y mutación ?

2.2 OBJETIVO GENERAL

- Experimentar los operadores de Clonación y Mutación bajo el principio de la Ingeniería Genética en el Funcionamiento de los Algoritmos Genéticos.

2.3 OBJETIVOS ESPECIFICOS

- Determinar las funciones de evaluación de Optimización Funcional

- Analizar y describir el funcionamiento los operadores de Clonación y Mutación
- Construir un programa que implemente el AG y experimentar con las funciones de optimización funcional
- Construir un programa que implemente el AG con la incorporación de Ingeniería Genética y experimentar con las funciones de optimización funcional
- Identificar las ventajas de desventajas de incorporación de la Ing. Genética en el funcionamiento de Algoritmos Genéticos.

2.4 HIPOTESIS

“ La utilización de los principios de la Ingeniería Genética Influye en el funcionamiento del Algoritmo Genético”

2.5 JUSTIFICACION E IMPORTANCIA

Es en los actuales momentos en que la humanidad esta sorprendida frente al vertiginoso avance las tecnologías emergentes como la Computación Evolutiva, Ingeniería Genética y otros por todo ello nuestras vidas serán influenciadas por los resultados que estas presenten, los resultados se verán en la medicina la agricultura y por supuesto también el desarrollo de la Informática y la Computación precisamente es esta nuestra preocupación de que manera la Ingeniería Genética y la Computación Evolutiva pueden cooperar en la búsqueda de las mejores soluciones a los problemas de optimización, por ello el presente trabajo hemos experimentado en optimización funcional.

Dentro de la Maestría en Informática de la UNA-Puno una de la áreas de investigación que es la inteligencia computacional con el presente trabajo estaríamos iniciando una

pagina de su desarrollo y con ello contribuiríamos a encaminar una parte del campo de la Inteligencia Artificial, por otro lado en mucha de nuestra carreras profesionales de nuestra universidad se llevan curso de optimización, métodos numéricos, programación lineal, investigación operativa en los que se esta desarrollando los contenidos de la manera tradicional siguiendo los procedimientos computacionales que son en su mayoría determinísticos, esto es decir utilizando formulas que son aplicadas bajo determinadas condiciones, con el presente trabajo damos una alternativa a esta forma de resolver estos problemas, presentamos los métodos heurísticos de búsqueda de soluciones en problemas de optimización funcional, con ello pretendo ampliar un poco mas el espectro de metodologías en este campo de la Inteligencia Computacional

2.6 REVISION DE LITERATURA

2.6.1 UN POCO DE HISTORIA Y PRESENTACIÓN DE LOS ALGORITMOS GENETICOS

Los Algoritmos Genéticos basan su funcionamiento en el principio de la Selección de las Especies, lo que hoy conocemos por *Darwinismo*. El Darwinismo tiene su origen en los trabajos de Alfred Russel Wallace sobre la tendencia de las variedades de seres vivos a apartarse del tipo originario, y por supuesto, también en los trabajos de Charles Darwin. Se cree que los escritos de Wallace fueron leídos por Darwin creando un fuerte impacto e interés en él, y se habla de la falta de decoro que supuso la lectura de los trabajos de Wallace en último lugar durante la primera sesión de la Sociedad Linneana, el 1 de julio de 1858, después de la lectura de varios resúmenes de Darwin, aún habiéndose acordado lo contrario.

En cualquier caso, Darwin se puso a trabajar de inmediato en su famoso libro "El origen de las Especies", que apareció en 1859. Anteriormente a esto, otras importantes contribuciones fueron la teoría de Jean-Baptiste de Lamarck y los estudios de Geología de Charles Lyell (basados en James Hutton), donde se ofrece una explicación de cómo la selección natural puede imprimir variaciones dañinas en una especie, y en los cuales se cree que se basó el trabajo posterior de Edward Blyth sobre la existencia de una estructura genética característica en toda forma viviente.

El darwinismo está también muy influido por la obra del economista inglés Thomas Robert Malthus, autor de "Ensayo sobre el principio de la población" donde expuso sus doctrinas sobre el crecimiento de la población. Dada la elevada tasa (geométrica) de reproducción de todos los seres orgánicos, su número tiende a crecer a ritmo exponencial, y dado que los alimentos, espacio físico, etc. no lo hacen en la misma proporción (crecen de forma aritmética o lineal), y mientras esto ocurra, nacerán muchos más individuos de los que es posible que sobrevivan, y en consecuencia, como sea que generalmente se recurre a la lucha por la existencia, ya sea con individuos de la misma especie o de especies distintas, o simplemente con el entorno, intentando modificar sus características adversas, se desprende que un individuo, si actúa de un modo provechoso para él, tendrá una mayor probabilidad de sobrevivir, y será seleccionado naturalmente. En resumen, que sobreviven los más fuertes.

DARWINISMO

La teoría de la selección de las especies sostiene que aquellos individuos de una población que posean los caracteres más ventajosos dejarán proporcionalmente más descendencia en la siguiente generación; y si tales caracteres se deben a diferencias genéticas, que pueden transmitirse a los descendientes, tenderá a cambiar la composición genética de la población, aumentando el número de individuos con dichas características. De esta forma, la población completa de seres vivos se adapta a las circunstancias variables de su entorno.

El resultado final es que los seres vivos tienden a perfeccionarse en relación con las circunstancias que los envuelven. En conclusión: la población cambia (evoluciona) hacia la figura del más fuerte.

Nadie pone en duda el hecho de que las especies evolucionan (cambian). Sin embargo, existe una discusión acerca de cómo lo hacen, porqué lo hacen, y hacia dónde se dirige la evolución.

La teoría de la selección de las especies argumenta que la evolución de los seres vivos en la naturaleza, (de la cual nosotros somos fruto), se produce gracias a la

competencia y la lucha entre los individuos. Este es el "cómo", aparentemente el mismo argumento ilustra el porqué. Los seres vivos evolucionan porque no pueden dejar de hacerlo: los seres vivos pueden tener o no esta actitud de competencia, tal vez puedan decidir si participar o no en estas luchas contra otros, pero en cualquier caso los que no luchan morirán o se reproducirán en menor grado, por lo que un comportamiento pacífico tenderá a desaparecer. En cuanto a la vida, la explicación más sencilla es que se creó al azar y que su objetivo es obtener individuos cada vez más perfectos y poderosos.

Darwin incluso intenta probar que la evolución es la divina providencia de la naturaleza, e intenta justificar lo que a los ojos del hombre parece éticamente incomprensible; en palabras de Darwin "[...] cuando meditamos sobre la lucha, podemos consolarnos con la creencia de que la guerra de la naturaleza no es continua, que no se siente miedo, que la muerte es generalmente rápida y que los vigorosos, los sanos y los felices son los que sobreviven y se multiplican".

Todo esto lo entiende perfectamente el programador de Algoritmos Genéticos. Los Algoritmos Genéticos funcionan muy bien en problemas simples, y parece lógico que la evolución se haya producido gracias a un mecanismo similar. En un Algoritmo Genético se eligen las mejores soluciones, para que al reproducirse, generen otras nuevas que combinen los aspectos positivos de cada progenitor. Sin embargo, en muchos problemas ocurre que en la combinación de soluciones no sólo no se mantienen las cualidades positivas de los progenitores, sino que además se generan con frecuencia soluciones no válidas, cuya aproximación al objetivo buscado es nula. A esto se le ha llamado el "problema de la decepción", "epistasia" o "ausencia de bloques constructores". Aunque el gen mutado se transmita, es posible que se necesitaran de otras mutaciones para que el resultado sea beneficioso.

CONCLUSIONES DEL DARVINISMO

- o El principal mecanismo de los seres vivos es una, incluso un cambio geométrica.
- o Los individuos descendientes de una pareja son todos distintos.

- En la naturaleza hay escasez de recursos, es decir, de alimentos, espacio para reproducirse.. Por tanto se va a establecer una lucha entre todos los individuos de esa especie por esos recursos, con lo cual los mejor dotados van a sobrevivir, y los menos dotados no. Esto lo llamó SELECCIÓN NATURAL.
- Estos individuos que sobreviven pasan sus características a sus descendientes.
- Los caracteres adquiridos se heredan

FALLOS DE DARWIN

1º Los caracteres adquiridos no se heredan.

2º Creía que evolucionaban los individuos, no las especies.

NEODARWINISMO

Pierde el nombre de darwinismo porque pretende llevar al día los postulados de Darwin. Al perder ese nombre llegaron a una conclusión, los que evolucionaban no son los individuos aislados, sino poblaciones enteras. En segundo lugar, los cambios que se producen en las poblaciones y la aparición de nuevos caracteres se deben a dos cosas; mutaciones y recombinaciones genéticas.

MUTACIONES, son cambios que se producen de forma espontánea y su frecuencia es de 1×10^{-5} . A priori, se puede considerar que una mutación es perjudicial.

RECOMBINACIONES GENÉTICAS, son combinaciones entre los genes del padre y de la madre en el momento de formarse los gametos.

Las mutaciones junto con las recombinaciones genéticas hace que los individuos sean todos distintos.

El Genotipo (no se ve) es un conjunto de genes que tiene un individuo, que es el que se transmite a sus descendientes. El genotipo y la influencia del ambiente es el fenotipo (se ve).

MUTACIONES

Una mutación es un cambio heredable en el material genético de una célula.

En la naturaleza las mutaciones se originan al azar y, aunque las causas siguen siendo inciertas, se conocen bastantes agentes externos, mutágenos, que pueden producir mutaciones como: las radiaciones ambientales y sustancias químicas. Una mutación, puede provocar alteraciones en el organismo en el que se presente; pero desaparece en el momento en que muere el individuo en que se originó.

Se distinguen varios tipos de mutaciones en función de los cambios que sufre el material genético.

1. **Mutaciones cromosómicas** . Este tipo de mutaciones provoca cambios en la estructura de los cromosomas.

Delección. Implica la pérdida de un trozo de cromosoma; los efectos que se producen en el fenotipo están en función de los genes que se pierden.

Duplicación. En este caso existe un trozo de cromosoma repetido.

2. **Mutaciones genómicas**. Este tipo de mutaciones afectan a la dotación cromosómica de un individuo, es decir, los individuos que las presentan tienen en sus células un número distinto de cromosomas al que es propio de su especie. No son mutaciones propiamente dichas, porque no hay cambio de material genético, sino una aberración, la cual suele ser el resultado de una **separación anormal** de los cromosomas durante la **meiosis**, con lo que podemos encontrarnos individuos triploides (3n), tetraploides (4n), etc.

Estos poliploides así formados son genéticamente muy interesantes en las plantas cultivadas, y hoy en día la mayoría de variedades gigantes de fresas, tomates, trigo, ... que existen en el mercado, tienen este origen.

Mutaciones génicas. Son las verdaderas mutaciones, porque se produce un cambio en la estructura del ADN.

El primer evolucionista fue Lamarck que en 1809 en su obra filosofía zoológica estable el transformismo según el cual los seres vivos forman un todo continuo. Lamarck se inclinó por un cambio gradual, según el cual unas especies se transformaron en otras sin necesidad de recurrir a nuevas creaciones. El mecanismo que propuso fue que los organismos desarrollan a lo largo de su vida estructuras que se adaptan mejor al medio ambiente, y que estas modificaciones estructurales son hereditarias. El Lamarckismo se resume en la frase: "El uso crea el órgano y el desuso lo atrofia." Su teoría no fue aceptada por sus contemporáneos al contraponerse a la teoría creacionista defendida por Cuvier, la cual gozaba del prestigio de este naturalista.

En 1859 Charles Darwin, nieto de Erasmus, publica Sobre el Origen de la Especies por medio de la Selección Natural. Darwin estudió la distribución de las especies en la naturaleza durante un viaje científico alrededor del mundo así como la generación de híbridos artificiales en sus experimentos botánicos. Según Darwin en las especies hay una tendencia enorme a la variación sobre las que actúa la selección natural y la selección sexual. De la lectura de Malthus derivó la idea de una sobreabundancia de prole en competición de unos recursos limitados, y esta lucha le sugirió un guión para el cambio de composición de una población. Aquellos organismos que poseyeran caracteres que los ajustaran mejor a su ambiente, tendrían una tendencia a reproducirse de una forma más prolífica que los otros, y sus caracteres se transmitirán a las generaciones futuras en mayor proporción. Así, las poblaciones estarían continuamente mejorando, y sus adaptaciones a los ambientes a los que están sometidas, y las poblaciones con adaptaciones no adecuadas acabarían extinguiéndose.

ASPECTOS GENETICOS UTILIZADOS EN EL ANTIGUO PERU

La reducción del riesgo climático en los ecosistemas, permitió el uso adecuado de esos nichos biológicos y el aprovechamiento eficiente de sus especies vegetales y animales. A su vez, la domesticación y el mejoramiento genético de especies y variedades servían y variedades servían como se ha visto, para enfrentar el impacto ambiental.

Como lo señalara el Ing. Oscar Blanco "... la diversidad ambiental de los Andes ha originado, como es lógico, una profusa variabilidad genética en la flora y fauna, como resultado de un permanente proceso evolutivo. La especialización de los ecotipos y la sub-siguiente radiación adaptativa han sido favorecidas, por un lado, por la intensa radiación solar que caracteriza a las altas montañas y por el otro, por el efecto aislante de barreras físicas como los nevados...en resumen, los Andes son un emporio de variabilidad genética" (Blanco, 1987).

Como él destaca más adelante, en los Andes se domesticaron más de 150 especies vegetales y cinco especies animales. Entre los vegetales se encuentran 45 especies de granos, tubérculos y leguminosas (como maíz, papa, frijoles, quinua, kiwicha, oca, camote, pallar, olluco, tarwi o lupinus, etc, y 45 especies de frutas (como tomate, calabaza, pepino, tomate de árbol, chirimoya, lúcuma, etc).

Si bien la mayoría de las especies se han domesticado en las zonas altoandinas, también las otras regiones naturales, como la costa y la selva han producido sus propias especies y variedades, en armonía con las culturas correspondientes.

Junta a la domesticación de especies silvestres, las culturas andinas utilizaron técnicas de mejoramiento y remodelación genética. "...en cada una de las especies vegetales cultivadas se han buscado variedades de mayor productividad, de ciclo vegetativo más corto, resistentes a factores climáticos adversos (como la helada, sequía o granizo), a diversidad de insectos y a otros parámetros animales como las enfermedades resistentes a la salinidad con características de sabor, textura, color y otras cualidades organo-lépticas plasticidad adaptativa y en general todo tipo de intensificación de los determinantes genéticos de la domesticidad" (Blanco, 1987).

Asimismo señala "para conseguir avances de magnitud como los que nos muestran los materiales mejorados andinos se requería de un vasto acervo genético que sólo se puede conseguir mediante colecciones variedades conocidas hoy en día como "bancos de germoplasma", cuya función es, además de concentrar una amplia base genética, la de preservar el material de la erosión genética... la selección, método de mejoramiento genético por excelencia, fue profusamente practicada en la obtención de variedades como maíz, quinua, tarwi, frijoles, pallares, frutales y tubérculos" (Blanco, 1987).

El uso y conservación de las semillas incluía técnicas de limpieza y refrescamiento (en trueque con las mantenidas en los relictos o genocentros de las partes altas) a fin de asegurar el vigor de las plantas. "Hubo un vasto conocimiento de los fenómenos de reposo fisiológico o latencia en ciertas semillas o tubérculos que determinaban el momento oportuno de siembra" (Blanco, 1987).

En relación a las especies animales, la domesticación y el mejoramiento genético más importante fue la de los camélidos sudamericanos, en especial la llama y la alpaca. Estos animales, como lo ha mostrado la ciencia moderna, son los que mejor se adaptan a las frágiles ecologías de las praderas altoandinas.

Probablemente incidió en su uso el interés de los incas en controlar el transporte y la alimentación de sus tropas o la producción de lana fina para la textilería usada en los intercambios comerciales. De igual manera, la domesticación del cuy o conejillo de indias, fue importante para la alimentación familiar.

2.6.2 AVANCES DE INVESTIGACIÓN Y RECONOCIMIENTO DE HECHOS

DIPLOIDES Y DOMINANCIA

- En la Naturaleza, los genotipos son diploides y contienen uno o más pares de cromosomas (a los que se les llama homólogos), cada uno de los cuales contiene información (redundante) para las mismas funciones. Por ejemplo, asumamos que tenemos el siguiente cromosoma diploide:

AbCDefGhIj

aBCdeFgHi j

Si asumimos que en este caso los genes representados por letras mayúsculas son los dominantes y los representados mediante letras minúsculas los recesivos, entonces el fenotipo correspondiente al cromosoma anterior sería

ABCDeFGHIj

Aunque no se sabe con certeza la razón de la redundancia en los cromosomas, las teorías más aceptadas en Biología dicen que los diploides ayudan a conservar alelos y combinaciones de alelos que fueron útiles en el pasado, y que mediante la dominancia se puede proteger a esos alelos del daño que pueda causar la selección en un ambiente hostil. En el caso del algoritmo genético, los diploides pueden usarse para mantener soluciones múltiples (al mismo problema), las cuales pueden conservarse a pesar de que se exprese sólo una de ellas. De esta manera las viejas lecciones no se pierden para siempre, y la dominancia permite que se recuerden las viejas lecciones de vez en cuando. Hillis (1990, 1992)

INVERSIÓN

Holland (1975) propuso formas de adaptar la codificación de su algoritmo genético original, pues advirtió que el uso de cruza de un punto no trabajaría correctamente en algunos casos.

El operador de inversión es un operador de reordenamiento inspirado en una operación que existe en genética. A diferencia de los AGs simples, en genética la función de un gene es frecuentemente independiente de su posición en el cromosoma (aunque frecuentemente los genes en un área local trabajan juntos en una red regulatoria), de manera que invertir parte del cromosoma retendrá mucha (o toda) la "semántica" del cromosoma original.

Para usar inversión en los AGs, tenemos que encontrar la manera de hacer que la interpretación de un alelo sea la misma sin importar la posición que guarde en una cadena. Holland propuso que a cada alelo se le diera un índice que indicara su posición "real" que se usaría al evaluar su aptitud. Por ejemplo, la cadena 00010101 se codificaría como:

{(1, 0) (2, 0) (3, 0) (4, 1) (5, 0) (6, 1) (7, 0) (8, 1)}

en donde el primer elemento de cada uno de estos pares proporciona la posición "real" del alelo dado. La inversión funciona tomando dos puntos (aleatoriamente) a lo largo

de la cadena, e invirtiendo la posición de los bits entre ellos. Por ejemplo, si usamos la cadena anterior, podríamos escoger los puntos 3 y 6 para realizar la inversión; el resultado sería:

{ (1, 0) (2, 0) (6, 1) (5, 0) (4, 1) (3, 0) (7, 0) (8, 1) }

Esta nueva cadena tiene la misma aptitud que la anterior porque los índices siguen siendo los mismos. Sin embargo, se han cambiado los enlaces alélicos. La idea de este operador es producir ordenamientos en los cuales los esquemas benéficos puedan sobrevivir con mayor facilidad. Por ejemplo, supongamos que en el ordenamiento original el esquema 00**01** es muy importante. Tras usar este operador, el esquema nuevo será 0010****. Si este nuevo esquema tiene una aptitud más alta, presumiblemente la cruce de un punto lo preservará y esta permutación tenderá a diseminarse con el paso de las generaciones.

CRUZA CON REORDENACIÓN

Algunos investigadores han utilizado exitosamente operadores que combinan la cruce con la inversión, a fin de mantener permutaciones que, de otra manera, se perderían con la cruce convencional. La aplicación más común de este tipo de operadores han sido los problemas como el del viajero, en el que una representación cromosómica obvia sería usar números de 1 a N (donde N representa el número de ciudades a visitarse), de tal forma que un individuo podría ser:

1 2 3 4 5 6 7 8 9

indicando que las ciudades se visitarán en orden ascendente.

Un operador que se ha usado frecuentemente en la literatura es el denominado PMX (*Partially Matched Crossover*). Para entender cómo funciona veamos el siguiente ejemplo en el que tenemos las 2 cadenas siguientes como padres:

$$\begin{array}{l}
 A = 9 \ 8 \ 4 \\
 B = 8 \ 7 \ 1
 \end{array}
 \left| \begin{array}{l}
 5 \ 6 \ 7 \\
 2 \ 3 \ 10
 \end{array} \right.
 \left| \begin{array}{l}
 1 \ 3 \ 2 \ 10 \\
 9 \ 5 \ 4 \ 6
 \end{array} \right.$$

Se consideran entonces 2 puntos de cruce que se eligen aleatoriamente. Posteriormente, se intercambian estos 2 segmentos en los hijos que se generan. El resto de cada hijo se produce haciendo mapeos entre los padres.

Retomemos en ejemplo anterior en que los padres eran:

$$\begin{array}{l}
 A = 9 \ 8 \ 4 \\
 B = 8 \ 7 \ 1
 \end{array}
 \left| \begin{array}{l}
 5 \ 6 \ 7 \\
 2 \ 3 \ 10
 \end{array} \right.
 \left| \begin{array}{l}
 1 \ 3 \ 2 \ 10 \\
 9 \ 5 \ 4 \ 6
 \end{array} \right.$$

Los hijos que se producen son entonces:

$$\begin{array}{l}
 A' = 9 \ 8 \ 4 \\
 B' = 8 \ 10 \ 1
 \end{array}
 \left| \begin{array}{l}
 2 \ 3 \ 10 \\
 5 \ 6 \ 7
 \end{array} \right.
 \left| \begin{array}{l}
 1 \ 6 \ 5 \ 7 \\
 9 \ 2 \ 4 \ 3
 \end{array} \right.$$

El bloque entre las 2 líneas horizontales simplemente se intercambia (sin modificaciones) de los padres. Para el resto de cada cadena procedemos a chequear los mapeos entre los 2 cromosomas. Por ejemplo, en este caso, el 5 corresponde con el 2, el 3 con el 6 y el 10 con el 7. De tal manera que en la cadena A' tendrá 6 en vez del 3, 5 en vez del 2 y 7 en vez del 10. Para la cadena B', el 10 reemplazará al 7, el 2 al 5 y el 3 al 6.

Este operador preserva las permutaciones codificadas en un cromosoma, y hace que siempre se tengan soluciones válidas. Aunque se han diseñado otros operadores similares (por ejemplo el OX —*order crossover*— y el CX —*cycle crossover*—), el PMX sigue siendo considerado un operador clásico, además que experimentalmente se ha demostrado que es superior a los otros 2 en el problema del viajero.

2.6.3 SISTEMA DE CONCEPTOS

TERMINOS UTILIZADO EN INGENIERIA GENETICA

Alelos: cada uno de los dos genes presentes en el mismo lugar (locus) del par de cromosomas homólogos. En general, uno de los diferentes estados alternativos del mismo gen.

Clones: grupo de células o de organismos de idéntica constitución genética entre sí y con el antepasado común del que proceden por división binaria o por reproducción asexual.

Código Genético: código cifrado por la disposición de nucleótidos en la cadena polinucleótida de un cromosoma que rige la expresión de la información genética en proteínas, es decir, la sucesión de aminoácidos en la cadena polipeptídica. La información sobre todas las características determinadas genéticamente en los seres vivos genética está almacenada en el ADN y cifrada mediante las 4 bases nitrogenadas. Cada sucesión adyacente de tres bases (codón) rige la inserción de un aminoácido específico. En el ARN la timina es sustituida por uracilo. La información se transmite de una generación a otra mediante la producción de réplicas exactas del código.

Cromosoma: corpúsculo intracelular alargado que consta de ADN, asociado con proteínas, y constituido por una serie lineal de unidades funcionales conocidas como genes. La especie humana tiene 46 cromosomas (23 pares). Su número varía desde el mínimo de un cromosoma en las obreras de la hormiga *Myrmecia pilosula* hasta los 1.260 cromosomas (630 pares) del helecho *Ophioglossum recitulatum*

Evolución biológica: cambios primero molecular, después celular, y por último de organismos, a lo largo de la historia como resultado de mutaciones en el ADN, de su reproducción y de procesos de selección. Los caracteres adquiridos no se heredan.

La especie humana comparte el 98'4% del ADN con el de dos especies de chimpancé, el común y el pigmeo. La evolución depende sobretodo de mutaciones en los genes

reguladores de los genes estructurales, que hacen que se activen o desactiven, mas que de mutaciones en los mismos genes estructurales.

Fenotipo: conjunto de todas los caracteres manifiesto expresados por un organismo, sean o no hereditarias.

Gen: unidad física y funcional del material hereditario que determina un caracter del individuo y que se transmite de generación en generación. Su base material la constituye una porción de cromosoma (locus) que codifica la información mediante secuencias de ADN.

Genética: ciencia que trata de la reproducción, herencia, variación y del conjunto de fenómenos y problemas relativos a la descendencia.

Genoma: conjunto de todos los genes de un organismo, de todo el patrimonio genético almacenado en el conjunto de su ADN o de sus cromosomas.

Genotipo: constitución genética, de uno o mas genes, de un organismo en relación a un rasgo hereditario específico o a un conjunto de ellos.

Germoplasma: la variabilidad genética total, representada por células germinales, disponibles para una población particular de organismos.

Ingeniería genética: conjunto de técnicas utilizadas para introducir un gen extraño (heterólogo) en un organismo con el fin de modificar su material genético y los productos de expresión.

Manipulación genética: formación de nuevas combinaciones de material hereditario por inserción de moléculas de ácido nucleico, generadas fuera de la célula, en el interior de cualquier virus, plásmido bacteriano u otro sistema vector fuera de la célula. De esta forma se permite su incorporación a un organismo huésped en el que no aparecen de forma natural pero en el que dichas moléculas son capaces de reproducirse de forma continuada. Al referirse al proceso en sí, puede hablarse de manipulación genética, ingeniería genética o tecnología de ADN recombinante. También admite la denominación de clonación molecular o clonación de genes, dado que la formación de

material heredable puede propagarse o crecer mediante el cultivo de una línea de organismos genéticamente idénticos.

Mutación: cambio del material genético. Puede afectar a cambios en un par de bases del ADN, en un gen específico o en la estructura cromosómica. La mutación en la línea germinal o relativa a las células sexuales, puede conducir a patologías genéticas o a cambios substanciales de la evolución biológica. En relación a las células somáticas la mutación constituye el origen de algunos cánceres y de ciertos aspectos del envejecimiento.

TERMINOS BIOLÓGICOS

Ambiente : Aquello que rodea a un organismo. Puede ser 'físico' (abiótico) o biótico. En ambos casos, el organismo ocupa un nicho que ejerce una influencia sobre su aptitud dentro del ambiente total. Un ambiente biótico puede presentar funciones de aptitud dependientes de la frecuencia dentro de una población; esto es, la aptitud del comportamiento de un organismo puede depender de cuántos más estén comportándose igual. A través de varias generaciones, los ambientes bióticos pueden fomentar la co-evolución, en la cual la aptitud se determina mediante la selección parcial de otras especies.

Aptitud : Vagamente se refiere a la capacidad de adaptación. Frecuentemente se mide como el éxito reproductivo relativo. También es proporcional al tiempo que resta para la extinción. Los individuos aptos son los que se adaptan a su ambiente actual y cuyos descendientes se adaptarán también a los ambientes futuros.

Darwinismo : Teoría de la evolución, propuesta por Charles Darwin, y que dice que la evolución se efectúa a través de variaciones aleatorias de características hereditarias, en conjunción con un proceso de selección natural (supervivencia del más apto). Muchos años después del trabajo de Darwin se descubrió un mecanismo físico para la evolución, en términos de genes y cromosomas.

Descendiente : Un individuo generado por cualquier proceso de reproducción.

Diploide : Se refiere a una célula que contiene 2 copias de cada cromosoma. Las copias son homólogas (i.e., contienen los mismos genes en la misma secuencia). En muchas especies que se reproducen sexualmente, los genes en uno de los conjuntos de cromosomas se heredan del gameto del padre (esperma), mientras que los genes del otro conjunto de cromosomas son del gameto de la madre (óvulo).

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).

Gene : Una unidad de herencia. Se puede definir de diferentes maneras según el objetivo. Los biólogos moleculares usan este término de manera un tanto más abstracta: "Cualquier información hereditaria para la cual hay un sesgo favorable o desfavorable de selección igual a varias o muchas veces su velocidad de cambio endógeno".

Haploide : Se refiere a la célula que contiene un solo cromosoma o conjunto de cromosomas, cada uno de los cuales consiste de una sola secuencia de genes. Un ejemplo es un gameto.

Individuo : Un solo miembro de una población.

Lamarckismo : Teoría de la evolución que precedió a la de Darwin. Lamarck creía que la evolución se había llevado a cabo a través de la herencia de ciertas características adquiridas. Esto es, las habilidades o características físicas que un individuo adquiere durante su vida pueden pasarse a sus descendientes. Aunque la herencia Lamarckiana no se lleva a cabo en la Naturaleza, esta idea ha sido aplicada exitosamente en algunas áreas de la computación evolutiva.

Migración : La transferencia de (los genes de) un individuo de una sub-población a otra.

Mutación : Proceso mediante el cual un solo nucleótido (elementos básicos de ADN) es modificado del padre al hijo. La mutación natural se produce normalmente por errores de copiado del ADN.

Nicho : En los ecosistemas naturales, hay muchas formas diferentes en la que los animales pueden sobrevivir (en los árboles, de la cacería, en la tierra, etc.) y cada estrategia de supervivencia es llamada un "nicho ecológico". Las especies que ocupan diferentes nichos (e.g., una que se alimenta de plantas y otra que se alimenta de insectos) pueden coexistir entre ellas sin competir, de una manera estable. Pero si dos especies que ocupan el mismo nicho se llevan a la misma zona, habrá competencia, y eventualmente la especie más débil se extinguirá (localmente). Por lo tanto, la diversidad de las especies depende de que ocupen una diversidad de nichos (o de que estén separadas geográficamente).

Ontogénesis : Se refiere al período de vida de un solo organismo desde su nacimiento hasta su muerte.

Población : Un grupo de individuos que pueden interactuar juntos, por ejemplo, para reproducirse.

Progenitor (Padre) : Un individuo que toma parte en la reproducción para generar uno o más individuos a los que se les conoce como descendientes o hijos.

Reproducción : La creación de un nuevo individuo a partir de 2 progenitores (reproducción sexual). La reproducción asexual es la creación de un nuevo individuo a partir de un solo padre.

Selección : Proceso mediante el cual algunos individuos en una población son seleccionados para reproducirse, típicamente en base a su aptitud (los de mayor aptitud son seleccionados).

La Selección Dura (*Hard Selection*) se da cuando sólo los mejores individuos se mantienen para generar progenia futura.

Esto contrasta con la "Selección Blanda" (*Soft Selection*), en la cual se usa un mecanismo probabilístico para mantener como padres a individuos que tengan aptitudes relativamente bajas.

TERMINOLOGÍA USADA EN 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 hipersuperficie que se obtiene al aplicar la función de aptitud a cada punto del espacio de búsqueda.

Bloque Constructor : Un grupo pequeño y compacto de genes que han co-evolucionado de tal manera que su introducción en cualquier cromosoma tiene una alta probabilidad de incrementar la aptitud de dicho cromosoma.

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 o un arreglo de enteros.

Cruza : Un operador de reproducción que forma un nuevo cromosoma combinando partes de cada uno de sus cromosomas padres. Estudiaremos más adelante las propuestas principales de cruza que existen.

Darwinismo : Teoría que inspiró a todas las ramas de la computación evolutiva.

Decepción (Deception) : La condición donde la combinación de buenos bloques constructores nos lleva a una reducción en la aptitud, en vez de un incremento. Este fenómeno fue propuesto originalmente por Goldberg para explicar el porqué el algoritmo genético no funciona satisfactoriamente en algunas tareas.

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.

Epístasis : Es la interacción entre los diferentes genes de un cromosoma. Se refiere a la medida en que la contribución de aptitud de un gen depende de los valores de los otros genes. Los problemas con poca epístasis o sin ella son triviales (una técnica escalando la colina es suficiente), pero los que tienen mucha epístasis son muy difíciles, aún para un algoritmo genético. Una epístasis elevada significa que no se pueden formar los bloques constructores, y por tanto habrá decepción.

Especiación : Técnicas análogas al aislamiento geográfico se usan 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 cantidad de migración es permitida.

Esquema : Un patrón de valores de genes en un cromosoma que puede incluir estados 'no importa' (don't cares). Por tanto, en un cromosoma binario, cada esquema puede especificarse mediante una cadena de la misma longitud del cromosoma, en la que cada carácter 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 de definición** es la distancia entre las 2 posiciones más alejadas de 'no importa's. Por lo tanto, #1##0# tiene un orden 2 y una longitud de definición de 3.

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, 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 máximos locales.

Gene : Una subsección 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.

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.

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

Individuo : Un solo miembro de la población. Cada individuo contiene un cromosoma (o, de manera más general, un genoma) que representa una solución posible a la tarea en cuestión (i.e., es un solo punto en el espacio de búsqueda). Usualmente se almacena también otro tipo de información en cada individuo (e.g., su aptitud).

Inversión : Un operador de reordenamiento que trabaja mediante la selección de puntos de corte en un cromosoma, invirtiendo el orden de todos los genes comprendidos entre ellos.

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.

Nicho : Frecuentemente queremos mantener la diversidad de una población, y en algunos problemas se sabe que la función de aptitud es multimodal, es decir, que tiene varios picos máximos a la misma altura. En este caso podemos considerar cada pico de la función de aptitud como análogo a un nicho. Aplicando técnicas como la compartición de aptitud podemos impedir que la población converja a un solo pico, y formar en vez de eso sub-poblaciones estables a cada pico. Esto es análogo a tener diferentes especies ocupando diferentes nichos.

Operador de Reproducción : Un mecanismo que influencia la forma en que se pasa la información genética de padres a hijos durante la reproducción. Los operadores de reproducción caen en 3 amplias categorías: cruza, mutación y reordenamiento.

Operador Genético : Un operador de búsqueda que actúa sobre una estructura de codificación que es análoga al genotipo de un organismo (e.g., un cromosoma).

Población Panmítica : Una población mezclada en la cual cualquier individuo puede reproducirse con otro con una probabilidad que depende sólo 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 mayor parte de los algoritmos evolutivos convencionales tienen poblaciones panmíticas.

Reordenamiento : Un operador de reordenamiento es un operador de reproducción que cambia el orden de los genes en un cromosoma, con la esperanza de juntar los genes que se encuentran relacionados, facilitando así la producción de bloques constructores.

Sub-Población : Una población puede sub-dividirse 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 sub-divisiones pueden influenciar marcadamente la dinámica evolutiva de una población. Las subpoblaciones 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 Requisito : En algoritmos genéticos cuando una población no tiene una "variedad requisito", la cruce no será útil como operador de búsqueda, porque tendrá propensión a simplemente regenerar a los padres.

2.6.4 LEYES ACERCA DEL PROBLEMA Y EL METODO

LA COMPUTACIÓN EVOLUTIVA DENTRO DE LA INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial (IA) es un área de investigación muy polifacética, pues abarca una gran variedad de diferentes tareas tales como teoría de juegos, demostración de teoremas, predicción, resolución general de problemas, percepción (principalmente visión y reconocimiento de voz), programación automática, procesamiento en lenguaje natural, y aprendizaje de máquina en general. Desde los orígenes de la IA los investigadores coincidieron en afirmar que una entidad "inteligente" debiera ser capaz de mejorar su comportamiento en base a la propia experiencia que acumulara al efectuar la misma (o muy similar) acción un número repetido de ocasiones. Asimismo,

dicha entidad debiera ser capaz de detectar cuando ha cometido un error, y usarlos para mejorar su desempeño futuro. La siguiente caracterización de aprendizaje proporcionada por Michalski abarca muchas ideas importantes:

Aprender es construir o modificar representaciones de lo que se experimenta.

Esta definición se concentra en una representación interna que el sistema de aprendizaje (ya sea un humano o un programa de computadora) construye y modifica a partir de su ambiente. La investigación de la IA clásica se concentra en el uso de representaciones simbólicas que se basan en un número finito de primitivas de representación y reglas para la manipulación de símbolos. Juntas, estas primitivas y estas reglas constituyen un sistema formal, y por tanto proporcionan un modelo universal de computación.

El período simbólico de la IA se remonta al período comprendido entre 1962 y 1975 aproximadamente, seguido por un período muy intenso de sistemas basados en el conocimiento (sistemas expertos) de 1976 a 1988, el cual enfatizó la incorporación de grandes cantidades de conocimiento dentro de los sistemas de aprendizaje. Sin embargo, las representaciones simbólicas (e.g., lógica de predicados, redes semánticas o "frames") fueron una característica central de este período. Actualmente, la IA está empezando a extenderse a otras direcciones, a la vez que trata de integrar los diferentes métodos existentes en sistemas de gran escala que puedan combinar sus ventajas al máximo posible (por ejemplo, las arquitecturas de pizarrón).

Sin embargo, hubo un período sub-simbólico antes del simbólico en la IA, el cual data de 1950 a 1965. Este período no usaba representaciones simbólicas del conocimiento, sino más bien subsimbólicas (numéricas). Los algoritmos evolutivos son un ejemplo de este período, y en la cronología presentada anteriormente pueden verse los intentos previos al período simbólico de la IA.

Además de la computación evolutiva, hubo otro paradigma sub-simbólico que precedió también al período simbólico: las redes neuronales. Esta técnica se basa en un modelo

simplificado del sistema nervioso central de los animales en el cual el conocimiento se distribuye sobre una serie de pesos que tienen las conexiones de los nodos (llamados unidades) de un grafo. Estas unidades intentan emular las neuronas, y las conexiones a las dendritas del cerebro y el axón. Ambas conforman las conexiones a otras células nerviosas, y los pesos pueden verse como una analogía de la fortaleza de la conexión sináptica. Durante el proceso de aprendizaje se actualizan los pesos de manera incremental.

Dentro del aprendizaje de máquina, los algoritmos evolutivos se consideran como una técnica de aprendizaje no supervisado que presentan 2 interesantes propiedades:

- No requieren de que se les presente ningún ejemplo o contra-ejemplo, o siquiera conocimiento. El algoritmo es capaz de generar por sí mismo el conocimiento que requiere.
- La creación de nuevos ejemplos (puntos de búsqueda) mediante el algoritmo se realiza de manera probabilística ("adivinando") en base al conocimiento existente. Si el punto elegido es satisfactorio, se mantiene en la base de datos (la población); de lo contrario, se le desecha mediante el proceso de selección.

Los algoritmos evolutivos se consideran normalmente como técnicas de búsqueda heurística que usan operadores probabilísticos (en vez de determinísticos) y que funcionan como "cajas negras", pues no requieren conocimiento específico del dominio para actuar, ni son aplicables a sólo cierto tipo de representaciones.

Definición de Algoritmo Genético (Koza, 1992)

"El algoritmo genético es un algoritmo matemático altamente paralelo que transforma un conjunto (población) de objetos matemáticos individuales (típicamente cadenas de caracteres de longitud fija que se ajustan al modelo de las cadenas de cromosomas), cada uno de los cuales se asocia con una aptitud, en una población

nueva (es decir, la siguiente generación) usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y sobrevivencia del más apto y tras haberse presentado de forma natural una serie de operaciones genéticas (notablemente la recombinación sexual)."

COMPONENTES BÁSICOS DE UN ALGORITMO GENÉTICO

Para poder aplicar el Algoritmo Genético se requiere de los 5 componentes básicos siguientes:

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

OPTIMIZACION FUNCIONAL

La Optimización puede considerarse como la búsqueda de la mejor solución (solución óptima) de un problema. El término mejor aquí depende del contexto en el que se trabaje, podría significar solución que minimiza los costes, o maximiza los beneficios, o que hace que la distancia recorrida sea mínima, etc. Esta primera reflexión sobre lo que se entiende por Optimización refleja claramente las importantísimas e indudables aplicaciones de esta área de las matemáticas a un amplio espectro de problemas; aplicaciones que surgen en la práctica totalidad de las Ciencias.

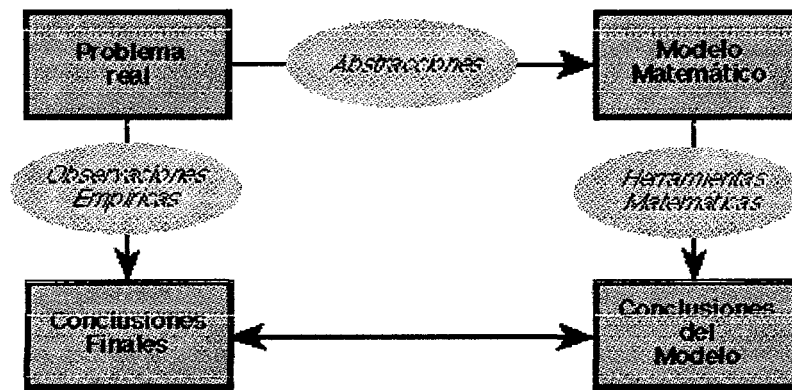
El abordar un problema real de Optimización supone básicamente dos etapas:

- Determinar el modelo matemático que rige el problema.

- Resolver dicho problema usando una serie de técnicas matemáticas.

Lógicamente, se buscan modelos sencillos que puedan resolverse con las herramientas analíticas y de cálculo disponibles. El precio pagado por la simplicidad puede ser la falta de fiabilidad de las conclusiones del modelo; por esa razón resulta recomendable su comparación con observaciones empíricas del problema real.

GRAFICO NRO 1



Los problemas que pueden ser resueltos mediante la Optimización Matemática son aquellos que pueden expresarse en la forma:

$$\begin{cases} \min & f(x) \\ & x \in D \end{cases} \quad \text{ó} \quad \begin{cases} \max & f(x) \\ & x \in D \end{cases}$$

Terminología:

- x es un vector de n componentes reales conocido como vector de variables de decisión.
- $f(x)$ es una función de n variables que se conoce como función objetivo.
- El conjunto D se conoce como región factible o conjunto de soluciones factibles.
- Los problemas de Optimización también se conocen como programas matemáticos.

El objetivo de la Optimización Matemática es, por tanto, encontrar máximos y mínimos de funciones de varias variables sujetas a una serie de restricciones.

Llegar a plantear un problema de optimización supone:

- Elegir las variables de decisión. Un excesivo número de variables puede aumentar la complejidad del problema, por lo que puede prescindirse de los factores con efectos mínimos sobre el modelo.
- Determinar la función objetivo. En algunas situaciones varias funciones pueden adaptarse al modelo, de la elección realizada puede depender la efectividad del sistema.
- Determinar las limitaciones o restricciones que han de imponerse a las variables para de esta forma obtener el espacio de soluciones factibles. Debe tenerse presente lo siguiente:
 - Omitir restricciones puede hacer que la solución del problema cambie totalmente.
 - Evitar imponer restricciones contradictorias, que hacen que el problema no tenga solución.
 - Controlar las restricciones redundantes, ya que no afectan a la solución pero pueden aumentar la complejidad del cálculo.
 - Deben usarse unidades de medida consistentes. Un error que suele ser común es utilizar diferentes unidades de medida en distintas restricciones.

Aunque esta materia sufrió un importante impulso durante los años cuarenta y cincuenta, sus orígenes se remontan bastante tiempo atrás. Los nombres de ilustres matemáticos de siglos pasados se encuentran, de una u otra manera, ligados a la Optimización. Muchos de los resultados conocidos en siglos pasados comenzaron a poder ser aplicados de forma efectiva en problemas reales. Como ejemplo cabe citar la utilización que las fuerzas aliadas hicieron de las técnicas de optimización durante la

Segunda Guerra Mundial para la planificación del abastecimiento a sus ejércitos; o la utilización por parte de los economistas de la antigua Unión Soviética.

EL PROBLEMA DE OPTIMIZACION FUNCIONAL

La Programación Matemática consiste en el cálculo de máximos y mínimos de funciones de varias variables sometidas a un conjunto de restricciones. Ambos problemas son en el fondo equivalentes ya que el punto en el que una función alcanza un máximo es el mismo en el que su opuesta alcanza un mínimo

Por tanto, el problema general de optimización puede considerarse como la búsqueda de mínimos de funciones sujetas a restricciones:

$$\begin{cases} \min f(x) \\ x \in D \end{cases}$$

A estos problemas también se les llaman **programas matemáticos** y a la Optimización Programación Matemática.

Terminología utilizada:

- x es el vector de variables de decisión.
- $f(x)$ es la función objetivo (también conocida como función de costes).
- D es el conjunto de soluciones factibles o región factible.

Dependiendo de la naturaleza de la función objetivo y de la región factible, existen diferentes áreas de la Optimización dedicadas al estudio de los diferentes problemas.

Problemas sin restricciones: no existe ninguna condición que deban cumplir las variables, cualquier punto es a priori una posible solución del problema, es decir el conjunto D es todo el espacio.

Problemas con restricciones de igualdad: las variables del problema deben verificar una serie de condiciones expresadas mediante ecuaciones (igualdades).

Problemas con restricciones de desigualdad: las variables del problema deben verificar una serie de condiciones expresadas mediante inecuaciones (desigualdades).

Problemas lineales: tanto la función objetivo como las funciones que determinan las restricciones que deben cumplir las variables son funciones lineales.

No obstante, existen ramas de la Optimización dedicada al estudio de otros tipos de problemas, por ejemplo la Programación Entera se encarga de los problemas en los que las variables solamente pueden tomar valores enteros.

Este tipo de problemas surgen con mucha frecuencia en la economía y en la gestión empresarial, véase los ejemplos como maximización de ingresos, minimización de costes.

SOLUCIONES ÓPTIMAS

Cuando se habla de solución óptima de un problema, o simplemente óptimo de dicho problema, es necesario distinguir entre los conceptos de:

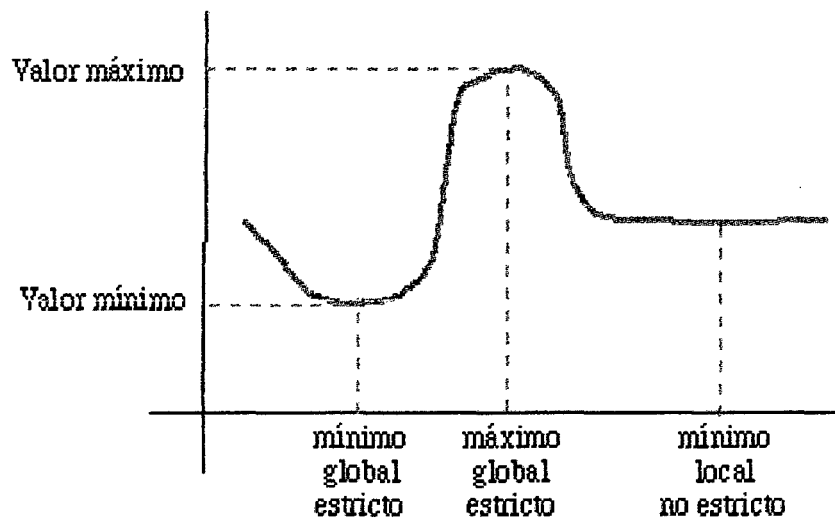
- **Punto óptimo:** punto en el cuál el problema alcanza el máximo o el mínimo.
- **Valor óptimo:** valor de la función objetivo en el punto óptimo.

Hecha esta observación, se puede hablar de dos tipos de mínimos y por supuesto de dos tipos de máximos:

- **Mínimos locales:** el valor de la función en ese punto es más pequeño que en cualquier otro punto en un cierto entorno suyo.
- **Mínimos globales:** el valor de la función en ese punto es más pequeño que en cualquier otro punto de la región factible.

Se dice que un óptimo es no estricto cuando existen otros puntos con valor de la función objetivo igual al valor óptimo. En caso contrario se dice que el óptimo es estricto. La siguiente figura pretende ilustrar los diferentes tipos de óptimos.

GRAFICO NRO 2



2.6.5 MARCO OPERATIVO

METODOS O PROCEDIMIENTOS

OPERACIÓN BÁSICA DEL ALGORITMO GENETICO

El funcionamiento de un AG como cualquier algoritmo evolutivo, comienza con una generación aleatoria de una población inicial con las potenciales soluciones. En una iteración cualquiera t el vector de cromosomas $P(t)=\{x1_t,x2_t,\dots,xn_t\}$ es evaluado de acuerdo a una función de ajuste o fitness para cada xi_t . Luego se forma una nueva población, $P(t+1)$ con los individuos que obtuvieron mejor ajuste. La nueva población es sometida a alteraciones mediante operadores denominados operadores genéticos como el de cruzamiento y la mutación para formar nuevas soluciones que volverán a competir entre ellas mediante la aplicación de la función de ajuste, hasta que después de la aplicación de un criterio de detención el algoritmo finalice.

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

DIAGRAMA NRO 1 ALGORITMO GENETICO CANONICO

```
begin
  generar población inicial, P(t);
  evaluar P(t);
  t:=0;
  repeat
    t:=t+1;
    Selección P( t ) usando P(t-1);
    Cruza( P(t) );
    Mutación(P(t) );
    Evaluar( P(t) );
  until encontrar una solución
end
```

REPRESENTACIÓN DE SOLUCIONES POTENCIALES

Hay una gran dependencia entre la forma de representación y el problema a resolver. Si, por ejemplo, el AG se utiliza para optimizar una función simple, de una sola variable en un dominio fijado por un intervalo, se puede usar una representación en un vector binario que represente los valores reales que puede tomar la variable independiente. Es este el caso de determinar el mínimo (o máximo) de una función $f(x)$ donde $x \in [a, b]$. La población de soluciones potenciales estará formada por un conjunto de vectores binarios que tomara cada x_i se divide el dominio $[a, b]$ en la cantidad de intervalos que requiera la precisión deseada a fin de determinar la cantidad de bits que se utilizara. Posteriormente se asigna a $x = a$ el valor binario de $000\dots0000_a$ y a $x = b$ el valor binario 1111.1111_b . Valores intermedios se obtienen por interpolación lineal:

$$x_R = a + x_B \frac{(b - a)}{\max_b}$$

Donde:

X_R es el valor real de la solución potencial.

X_B es el valor en binario representado en k bits.

\max_b es el máximo número representable con k bits.

A es el mínimo valor real que puede tomar x .

b es el máximo valor real que puede tomar x .

Típicamente cada organismo en la población consiste de un cromosoma, el cual es asimismo un vector de la forma:

$$\langle x_1, x_2, \dots, x_n \rangle$$

Donde a cada x_i se le llama *gen* y al valor que toman se le denomina *alelo*.

Normalmente estos cromosomas se representan usando un alfabeto binario, pero no

tiene que ser forzosamente de esa manera, ya que alfabetos de cualquier cardinalidad son posibles. Típicamente, se utilizan la llamada representación real (o de punto flotante) como alternativa para problemas de optimización numérica en el que las variables toman valores reales. El uso de letras o cualquier otro símbolo es también válido.

Cromosoma con representación binaria

1	0	0	1	1	1	0	1	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Cromosoma con representación flotante

Vector flotante (23.234 , 12.230 , 0.234 , 12.232)

Cromosoma con representación entera

Vector entero [23 , 34 , 45 , 1 , 2 , 90 , 0 , 23]

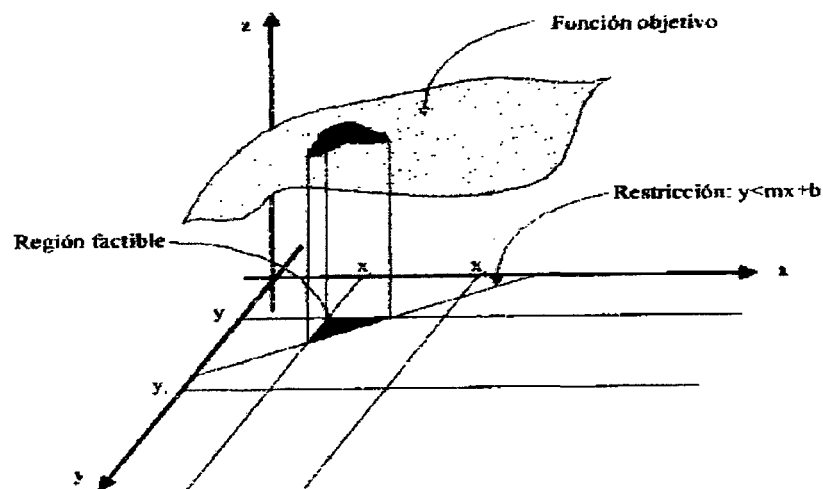
CREACIÓN DE UNA POBLACIÓN INICIAL

De manera similar al problema de la representación, la creación de una población inicial tiene que ver con el tipo de problema que se quiere resolver, una generación de soluciones potenciales en el caso de representación por vectores binarios, puede realizarse la generación de una población inicial por medio de una generación aleatoria adecuada, la única restricción tiene que ver con los límites impuesto en el espacio de búsqueda por lo que basta la generación aleatoria dentro del intervalo válida para cada variable independiente.

Se debe indicar que la resolución de problemas de optimización que incluyen restricciones que no tienen que ver con los límites de los intervalos entre los cuales se les permite moverse a las variables independientes, sino que delimitan el espacio de soluciones factibles dentro del dominio acotado por los límites de validez de las

variables. Puede verse en el ejemplo mostrado en el grafico nro 3 que dichos limites son diferentes mientras las variables x e y podrían moverse dentro del área rectangular delimitadas por los valores x_1 y x_2 para x y los valores y_1 e y_2 para la variable y , el espacio de soluciones factibles es el área marcada en la figura que cumple la restricción.

GRAFICO NRO 3



En este caso se suele aplicar penalizaciones a la solución que puede ir desde la pena de muerte (desechar la solución de plano) hasta la aplicación de funciones mas complejas que acorde con el grado de violación de la restricción, disminuyen el fitness del individuo. De esta manera se preservan algunos individuos que, si bien no son factibles, pueden poseer material genético que ayude con la futuras generaciones.

FUNCIÓN DE EVALUACIÓN

Una vez representadas la variables y generada la población inicial, es necesario ponderar cuan buenos son los candidatos (cada uno de los individuos de la población). La calidad de los mismos se refiere a cuan buena es la solución que proponen y si se intenta minimizar una función, costo en este caso, los individuos de menor costo serán

los mas apropiados. El valor de la función objetivo a minimizar, en este caso los mejores individuos serán los que tengan mayor valor.

La evaluación de cada individuo en función de su adaptabilidad al medio le otorga un merito o competencia que , o bien le ayudara a sobrevivir en las próximas generaciones o bien significara su muerte como resultado de la competencia con sus congéneres.

CICLO DE EVOLUCION

Puede observarse en el en el pseudo código la presencia de un bucle con un criterio de detención que es el encargado de poner fin a la evolución de le población. Cada ciclo representa una nueva generación. Este ciclo esta controlado por el criterio de detención adoptado que puede ser uno de los siguientes también.

- Detener después de N iteraciones.
- Detener cuando el *fitness* medio de la población cae dentro de una banda predefinida
- Detener cuando la diversidad población es baja
- Otros

En general, los criterios de detención pueden clasificarse en dos grupos:

- Condiciones independientes del proceso de evolución
- Condiciones atadas a dicho proceso

Dentro del ciclo de evolución pueden observarse tres acciones concretas:

Seleccionar

Cruzar

Mutar

La selección implica determinar cuales cromosomas serán considerados para procrear y así formar la siguiente generación. Es importante aquí el criterio con que esa solución es hecha. Una forma de selección consiste en asignar a cada cromosoma una

probabilidad de ser seleccionado, proporcional a su función de ajuste, aunque este no es el único criterio utilizado.

Seleccionar es escoger individuos para entrar en una etapa de modificación de la población que viene de la mano de los llamados operadores genéticos.

OPERADORES QUE ALTEREN LA COMPOSICIÓN DE LOS INDIVIDUOS

Los individuos de una población i deben generar descendencia transmitiendo sus características genéticas a sus hijos. Para ello deberán contarse con operadores que permitan dicha transformación y que necesariamente estarán influidos por el tipo de representación y de problema a resolver. Dos operadores utilizados en algoritmos genéticos son los operadores de cruzamiento y mutación, puede considerarse que el cruzamiento permite generar hijos a partir de padres seleccionados y la mutaciones consiste en realizar una modificación, normalmente pequeña, en el cromosoma de un individuo.

PARÁMETROS GENERALES QUE USA EL AG

Distintos parámetros y condiciones iniciales son necesarios para el ajuste de un algoritmo genético. La velocidad de convergencia se ve afectada entre otros por la cantidad de individuos en una población, la probabilidad de cruzamiento, la probabilidad de mutación, el elitismo, etc.

ESTRATEGIAS DE SELECCIÓN

SELECCIÓN PROPORCIONAL

Este nombre describe a un grupo de esquemas de selección originalmente propuestos por Holland que eligen individuos de acuerdo a los valores de sus funciones objetivo. En este caso, el "valor esperado" de un individuo (es decir, el número esperado de

veces que un individuo será seleccionado para reproducirse) es la aptitud del individuo dividida entre la aptitud promedio de la población. Existen algunas variantes de esta técnica:

- **La Ruleta (De Jong, 1975):** Es el método más común de implementación de esta técnica. A cada individuo se le asigna una rebanada de una "ruleta" circular proporcional a su aptitud. La rueda se gira N veces, donde N es el número de individuos en la población. A cada vuelta, el individuo que se encuentre bajo el marcador de la rueda es seleccionado como uno de los padres para la siguiente generación. Esta técnica puede implementarse de la manera siguiente:

1. Sume el valor esperado total de los individuos en la población. Llame a esta suma T .

2. Repetir N veces:

- Escoger un número aleatorio r entre 0 y T .

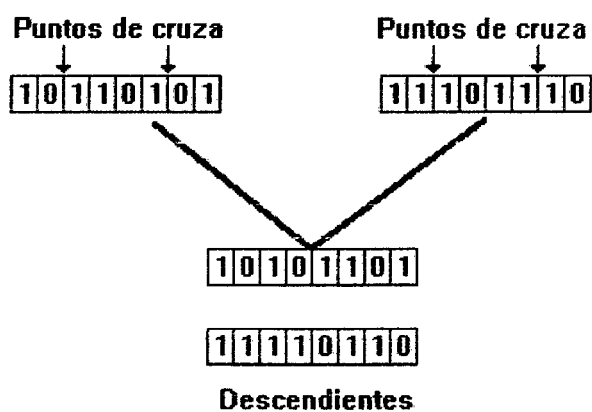
- Clicar a través de los individuos de la población, sumando los valores esperados, hasta que la suma sea mayor o igual a r . El individuo cuyo valor esperado haga que esta suma exceda este límite es el seleccionado.

El problema de la ruleta es que normalmente los AGs usan poblaciones pequeñas y entonces el número de hijos que se asignan a un individuo está normalmente bastante lejos de su valor esperado, y es incluso probable (aunque muy remotamente) que una serie de giros desafortunados seleccionen repetidamente al peor individuo de la población. La implementación de esta técnica es $O(n^2)$, aunque puede mejorarse su eficiencia si se usa búsqueda binaria para localizar la posición correcta de la rueda en vez de recorrer la población secuencialmente con una búsqueda lineal. Esto requiere memoria extra y un recorrido $O(n)$ a través de la lista para calcular los totales acumulados por cada segmento de la rueda. La complejidad total se reduce entonces a $O(n \log n)$.

Uso de un solo punto de cruce entre 2 individuos. Observe que cada pareja de cromosomas da origen a 2 descendientes para la siguiente generación. El punto de cruce puede ser cualquiera de los 2 extremos de la cadena, en cuyo caso no se realiza la cruce.

REPRODUCCIÓN USANDO DOS PUNTOS DE CRUZA

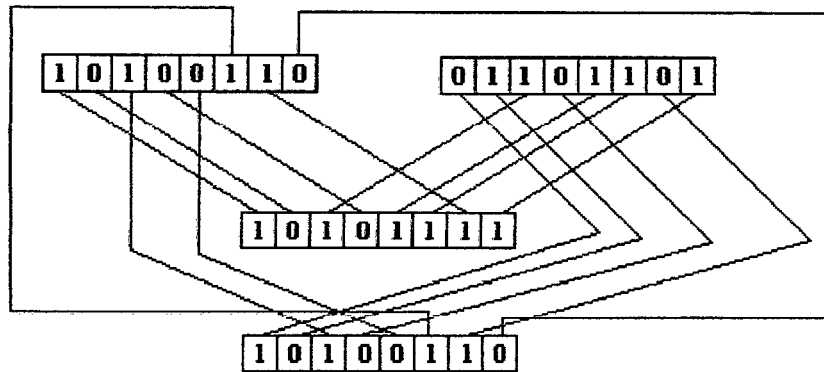
GRAFICO NRO 6



Uso de 2 puntos de cruce entre 2 individuos. Note como en este caso se mantienen los genes de los extremos, y se intercambian los del centro. También aquí existe la posibilidad de que uno o ambos puntos de cruce se encuentren en los extremos de la cadena, en cuyo caso se hará una cruce usando un solo punto, o ninguna cruce, según corresponda.

REPRODUCCIÓN USANDO CRUZA UNIFORME

GRAFICO NRO 7



Cruza Uniforme con probabilidad de 0.5. Nótese cómo la mitad de los genes de cada hijo proviene de cada uno de sus padres. La idea de este algoritmo es ir eligiendo, posición por posición, el padre que aportará un gen a cada hijo, usando la probabilidad definida por el usuario.

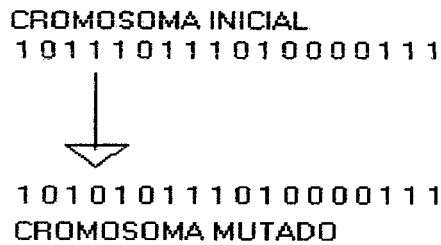
MUTACIÓN

La Mutación es otro operador genético de gran importancia que cambia aleatoriamente un gen de un cromosoma. Si usamos representación binaria, una mutación cambia un 0 en 1 y viceversa.

Este operador permite la introducción de nuevo material cromosómico en la población y desde la perspectiva teórica, asegura que—dada cualquier población— el espacio de búsqueda se encuentre totalmente conectado.

MUTACIÓN SIMPLE

GRAFICO NRO 8



CRITERIOS DE DETENCIÓN

Si supiéramos de antemano la solución final de nuestro problema de optimización, sería trivial determinar cómo detener un algoritmo genético. Sin embargo, puesto que esto no es normalmente posible, tenemos que usar uno de los siguientes criterios de detención:

- Correr el algoritmo genético por un cierto número (fijo) de generaciones.
- Detenerlo cuando la población se haya estabilizado, es decir, cuando la mayoría de los individuos tengan el mismo valor de aptitud.

TÉCNICAS DE REPRESENTACIÓN

- Cadenas Binarias (Tradicional)
- Códigos de Gray (Binaria)
- Punto Flotante (Binaria)
- Punto Flotante (Real)
- Punto Flotante (Entera)

REPRESENTACIÓN DE PUNTO FLOTANTE

- **Binaria** : Se discretiza el rango de cada variable fijando una precisión deseable (por ejemplo, 2 decimales), y posteriormente se tratan estos valores como si fueran enteros. Por ejemplo, si $-1.500 \leq x \leq 2.000$ y se asumen 3 dígitos de precisión, entonces el número de posiciones cromosómicas es:

$$\text{tamaño} = (\text{int}) \{ \log_2((\text{límite_superior} - \text{límite_inferior}) * 10^{\text{precisión}}) + 0.9 \}$$

Para este ejemplo: tamaño = 12 bits

- **Real** : Cada alelo es realmente un número real. Por ejemplo, un cromosoma podría ser: **(3.5, 2.76, -1.1679)**. Aunque los operadores genéticos tradicionales pueden modificarse para funcionar con esta representación, suelen usarse operadores especialmente diseñados para ella.
- **Entera** : Cada alelo es un entero, pero en su conjunto, el cromosoma representa un número real. Por ejemplo, un cromosoma podría ser: **(1, 6, 2, 9)** para representar 1.629, asumiendo que el punto decimal se encuentra fijo en la segunda posición. Los operadores tradicionales de cruce y mutación funcionan con cambios insignificantes. Particularmente, la mutación tiene que modificarse para que en vez de un NOT lógico lo que haga sea colocar un número aleatorio entre 0 y 9 (el rango de cada alelo). Asimismo, también puede usarse un mutador que perturbe en una cantidad más pequeña al alelo (por ejemplo que le sume o le reste un valor entre 1 y 3).

TÉCNICAS DE CRUZA PROPUESTAS

PARA LA REPRESENTACIÓN REAL

- **Cruza Simple** : Se define de la manera siguiente: si $s_v^t = \langle v_1, \dots, v_m \rangle$ y $s_w^t = \langle w_1, \dots, w_m \rangle$ se cruzan después de la posición k , los hijos resultantes son:

$$s_v^{t+1} = \langle v_1, \dots, v_k, w_{k+1}, \dots, w_m \rangle \text{ y } s_w^{t+1} = \langle w_1, \dots, w_k, v_{k+1}, \dots, v_m \rangle$$

Debe advertirse que los únicos puntos de cruce permisibles son entre cada valor numérico (usando esta representación es imposible hacerlo de otra manera). Sin embargo, este operador puede producir hijos que estén fuera del espacio de soluciones permisibles. Para evitar este problema, usamos las siguientes expresiones en la que $\alpha \in [0,1]$:

$$s_v^{t+1} = \langle v_1, \dots, v_k, w_{k+1} \cdot a + v_{k+1} \cdot (1-a), \dots, w_m \cdot a + v_m \cdot (1-a) \rangle$$

$$s_w^{t+1} = \langle w_1, \dots, w_k, v_{k+1} \cdot a + w_{k+1} \cdot (1-a), \dots, v_m \cdot a + w_m \cdot (1-a) \rangle$$

Resulta ahora de interés saber que valor de a usar para obtener el mayor intercambio posible de información. Sin embargo, debido a que se usan valores reales no se puede usar una búsqueda exhaustiva, y suele usarse una búsqueda binaria con profundidad limitada, en donde a tomará el valor más grande que se encuentre o 0 si ningún valor satisface las restricciones.

- **Cruza Aritmética Simple** : Se define de la manera siguiente: si $s_v^t = \langle v_1, \dots, v_m \rangle$ y $s_w^t = \langle w_1, \dots, w_m \rangle$ se cruzan, los hijos resultantes son:

$$s_v^{t+1} = \langle v_1, \dots, v_k', \dots, v_m \rangle \text{ y } s_w^{t+1} = \langle w_1, \dots, w_k', \dots, w_m \rangle$$

donde $k \in [1, m]$, $v_k' = a \cdot w_k + (1-a) \cdot v_k$ y $w_k' = a \cdot v_k + (1-a) \cdot w_k$. En este caso, a es un parámetro dinámico que se calcula en el contexto dado (vectores s_v, s_w) de tal manera que el operador sea cerrado (es decir, que los hijos que produzca sean soluciones válidas). Normalmente a se elige aleatoriamente a partir de un cierto rango (Michalewicz, 1992).

- **Cruza Aritmética Total** : Se define como una combinación lineal de 2 vectores: si $s_v^t = \langle v_1, \dots, v_m \rangle$ y $s_w^t = \langle w_1, \dots, w_m \rangle$ se cruzan, los hijos resultantes son:

$$s_v^{t+1} = a \cdot s_w^t + (1-a) \cdot s_v^t \text{ y } s_w^{t+1} = a \cdot s_v^t + (1-a) \cdot s_w^t$$

donde $a \in [0 \dots 1]$.

TÉCNICAS DE MUTACIÓN PROPUESTAS

PARA LA REPRESENTACIÓN REAL

• **No Uniforme** : Este operador se define de la manera siguiente: si $s_v^t = \langle v_1, \dots, v_m \rangle$ es un cromosoma (t es el número de generación) y el elemento v_k se seleccionó para ser mutado, el resultado es un vector $s_v^{t+1} = \langle v_1, \dots, v'_k, \dots, v_m \rangle$, donde

$$v'_k = \begin{cases} v_k + \Delta(t, LS - v_k) & \text{si un número aleatorio es } 0 \\ v_k - \Delta(t, v_k - LI) & \text{si un número aleatorio es } 1 \end{cases}$$

y **LS** y **LI** son el Límite Superior y el Límite Inferior de la variable v_k respectivamente. La función $\Delta(t, y)$ regresa un valor en el rango $[0, y]$ de tal manera que la probabilidad de $\Delta(t, y)$ se acerque a 0 conforme t se incrementa. Esto hace que este operador busque uniformemente al principio (cuando t es pequeña), y muy localmente en etapas posteriores; esto incrementa la probabilidad de generar el nuevo número más cerca de su sucesor que si se usara una elección meramente aleatoria. La función sugerida por Michalewicz es: $\Delta(t, y) = y \cdot \left(1 - r^{\left(\frac{t}{T}\right)^b}\right)$, donde r es un número aleatorio de $[0..1]$, T es el número máximo de generaciones, y b es un parámetro del sistema que determina el grado de dependencia a cada iteración (Michalewicz sugiere $b=5$).

• **Uniforme** : Se define de la manera siguiente: si $s_v^t = \langle v_1, \dots, v_m \rangle$ es un cromosoma y el elemento v_k se seleccionó para ser mutado, el resultado es un vector $s_v^{t+1} = \langle v_1, \dots, v'_k, \dots, v_m \rangle$, donde v'_k es un valor aleatorio (con distribución de probabilidad uniforme) en el rango $[LI..LS]$, donde **LI** y **LS** son el límite inferior y el límite superior de la variable respectivamente.

III RESULTADOS Y DISCUSIÓN

3.1 IDENTIFICACIÓN DE LAS FUNCIONES DE EVALUACIÓN

Las funciones de evaluación que se ha utilizado para verificar el funcionamiento del algoritmo genético simple y los operadores de clonación y mutación han sido determinados considerando los siguientes criterios.

- Que permite ver su funcionamiento en forma grafica
- Que sea ya probado y comprobado en trabajos anteriores
- Que sea simple de programar

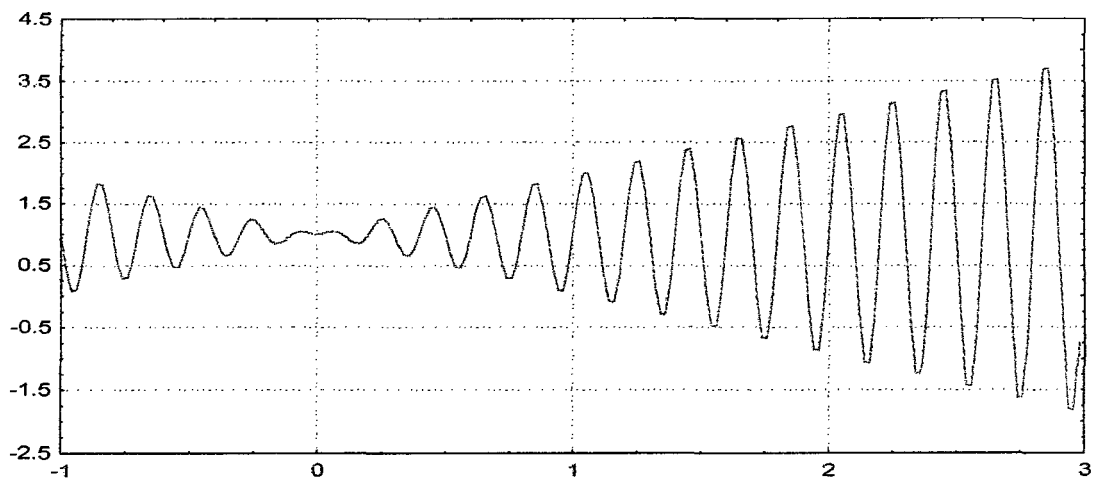
Bajo los criterios establecidos anteriormente se ha encontrado conveniente utilizar para el caso de una sola variable dependiente la función siguiente

$$Y = x \sin(10 \pi x) + 2$$

representado en la siguiente grafica

GRAFICO NRO 9

Grafica de la Funcion de evaluacion
 $y = x * \sin(10 * \pi * x) + 1$



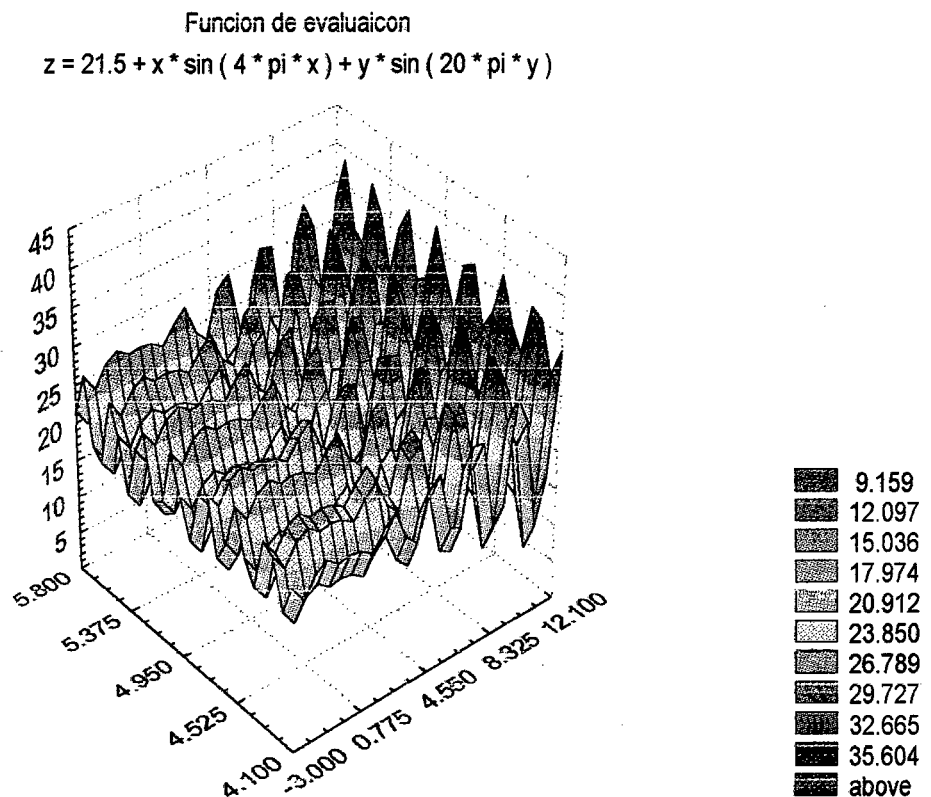
En esta función se puede observar la gran oscilación por lo que nos va a permitir observar de que manera el algoritmo genético nos va a dar una respuesta para encontrar el punto más alto es decir el punto máximo que alcanza la función y para que valor de x es que le corresponde.

Así mismo para evaluar en dos dimensiones se ha utilizado la función determinada como sigue

$$Z = 21.5 + x \sin(4\pi x) + y \sin(20\pi y)$$

representado en la siguiente gráfica

GRAFICO NRO 10



En esta función se puede observar las irregularidades que presenta en por lo que también es una buena prueba para verificar el funcionamiento del algoritmo genético.

Ambas funciones han sido utilizadas como parte de la investigación publicada por Zbigniew Michalewicz los cuales ya han sido estudiados verificados y cumplen los criterios de la presente investigación.

La incorporación del operador de mutación y clonación se hizo en la forma que se indica en el siguiente algoritmo.

DIAGRAMA NRO 2

ALGORITMO GENETICO CON OPERADORES DE CLONACION Y MUTACIÓN

```
begin  
  generar población inicial aleatoria G(0);  
  t:=0;  
  repeat  
    t:=t+1;  
    Evaluar G(t); // aplicando fitness  
    Seleccionar el mejor;  
    Clonar a todos los individuos de la población con el material  
    genético del mejor individuo;  
    Realizar la mutación de la población clonada;  
    Hacer reporte;  
  until encontrar una solución satisfactoria  
end
```

3.2 DESCRIPCIÓN DEL ALGORITMO GENETICO CON OPERADOR DE CLONACION Y MUTACION

Después de generar una población aleatoria, se selecciona el mejor individuo para que su material genético sirva en el proceso de clonación de los demás individuos en la siguiente generación, luego de estar determinado el mejor material genético se realiza la clonación esto es duplicar en forma exacta el cromosoma del individuo en la nueva población, en seguida se realiza la mutación para generar cambio en la estructura genética del individuo por lo que luego se evalúa, se selecciona al mejor individuo y se continúa con la siguiente generación y así sucesivamente hasta que se logre un determinado número de generaciones en nuestro caso 150, para una población de 50 individuos.

PROCESO DE SELECCIÓN.

Se selecciona el cromosoma del mejor individuo en la población inicial

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

PROCESO DE CLONACION

Clonar a todos los individuos de la siguiente población con el material genético del mejor individuo de la anterior población lo que significa realizar una duplicación exacta del genotipo.

Clonación del Primer individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Clonación del Segundo individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Clonación Tercer individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

.

.

.

.

n-1 individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

n-esimo individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

PROCESO DE MUTACIÓN

Se realiza la mutación simple en cada individuo alterando un bit elegido al azar;

Primer individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Este cromosoma cambia al siguiente ya clonado

0	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---



Alelo mutado

Segundo individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

1	0	0	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Alelo mutado ▲

Tercer individuo

1	0	1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

1	0	1	1	0	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---

Alelo mutado —▲

y así sucesivamente con todos los individuos de la población.

1	0	1	0	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Alelo mutado —▲

1	0	1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Alelo mutado —▲

3.3 IMPLEMENTACION DEL ALGORITMO GENETICO

OPTIMIZACION DE UNA SIMPLE FUNCION

La primera función seleccionada utilizaremos para optimizarla utilizando un algoritmo genético esta función esta denotada como

$$Y = x \sin (10 \pi x) + 2$$

El problema es encontrar un x del rango $[-1,2]$ que maximice la función $f(x)$ tal que el x_0 buscado cumpla.

$$f(x_0) \geq f(x), \text{ para todo } x \in [-1, 2].$$

Esta claro que un análisis de la función $f(x)$, los ceros de la primera derivada f' debe ser determinado por:

$$f'(x) = \sin(10\pi x) + 10\pi x \cdot \cos(10\pi x) = 0$$

La formula es equivalente a.

$$\tan(10\pi x) = -10\pi x$$

Esta claro que la ecuación de encima tiene un infinito numero de soluciones.

$$x_i = \frac{2i-1}{20} + \varepsilon_i, \text{ para } i = 1, 2, 3, \dots$$

$$x_0 = 0$$

$$x_i = \frac{2i+1}{20} - \varepsilon_i, \text{ para } i = -1, -2, -3, \dots$$

donde los términos ε_i representan secuencias de decremento de números reales que se aproximan a cero.

Observando el dominio de x, la función es máximo para $x_{19} = \frac{2*19-1}{20} + \varepsilon_{19}$, donde f (x19) = f (1.85) = 1.85 sin (10 π x) +2 = 3.85

REPRESENTACION DEL GENOTIPO

Para ambos casos se va a utilizar la representación canónica con un vector de bits que represente el cromosoma para representar el valor real de la variable x considerando los siguientes criterios.

Para determinar la longitud del cromosoma vamos a utilizar el criterio de precisión de seis decimales por lo tanto aplicando la formula respectiva tendremos lo siguiente.

$$\text{tamaño} = (\text{int}) \{ \log_2((\text{límite_superior} - \text{límite_inferior}) * 10^{\text{precisión}}) + 0.9 \}$$

$$\text{tamaño} = (\text{int}) \{ \log_2 ((2 - (-1)) * 10^6) + 0.9 \}$$

$$\text{tamaño} = 22$$

otra forma de determinar es considerar que la longitud de la variable x es 6; la precisión requerida que implica el rango de [-1 , 2] deberá ser dividido dentro de $3 * 1000000 = 3000000$ rangos de igual tamaño en vista que hay 1000000 divisiones en un espacio entero, a esto también se le denomina el espacio de búsqueda, es decir se ha dividido el dominio de x en 3000000 de divisiones.

$$2097152 = 2^{21} < 3000000 < 2^{22} = 4194304$$

22 bits

desde [0]

hasta [1]

El mapeo de una cadena binaria (b21, b20, ... b0) para un numero real x de el rango [-1 , 2] se completa en dos pasos.

Convertir la cadena binaria ($b_{21}b_{20}...b_0$) de base dos a base 10

$$((b_{21}b_{20}...b_0))_2 = \left(\sum_{i=0}^{21} b_i \cdot 2^i \right)_{10} = x_B$$

Buscar la correspondiente numero real x dado por $x_R = a + x_B \frac{(b-a)}{\max_b}$

$$x_R = -1 + x_B \frac{(2 - (-1))}{2^{22} - 1}$$

por ejemplo el cromosoma [1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1] representa el numero 0.637197, es decir $x_B = [1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1]_2 = 2288967$ entonces el valor real de $x = -1.0 + 2288967 \cdot 3/4194303 = 0.637197$ esta claro que los cromosomas :

[0] y

[1]

Representa los limites del dominio de x, -1 , 2.0 respectivamente.

POBLACIÓN INICIAL

El proceso de inicialización de la población es muy simple, se crea una población de cromosomas donde cada cromosoma es un vector binario de 22 bits, todos los 22 bits de cada cromosoma es inicializado aleatoriamente.

FUNCION DE EVALUACIÓN.

La evaluación de la aptitud de cada vector binario esta dado por la función $f(x)$ dado que en este caso ambos coinciden para este caso de optimización de encontrar un máximo.

$$Fitness(v) = f(x)$$

Donde el cromosoma v representa el valor real x ,

OPERADORES GENETICOS

La alteración del material genético se hizo con los operadores de cruzamiento de un punto y la mutación del bit seleccionado de manera aleatoria determinado por un valor probable establecido en los parámetros del AG.

$V1 = [1010110010100101010101]$

$V2 = [1110011010101110011100]$

Se elige un número al azar como 15 dentro de 1 y 22 que son los límites de la cadena binaria, entonces a partir de este punto se realiza la cruce para determinar el material genético de los hijos.

PARÁMETROS

Para este problema de optimización se ha considerado los siguientes parámetros

Tamaño de la población = 50

Probabilidad de cruzamiento = 0.80

Probabilidad de mutación = 0.15

3.4 RESULTADOS EXPERIMENTALES DEL ALGORITMO GENETICO CANONICO EN OPTIMIZACION FUNCIONAL

En la tabla de proporciona la evaluación del funcionamiento del AG en 150 generaciones encontrándose el máximo identificado con el cromosoma. Que corresponde al valor real de 1.850773 y como valor máximo de la función $f(x) = 2.85$

CUADRO NRO 2

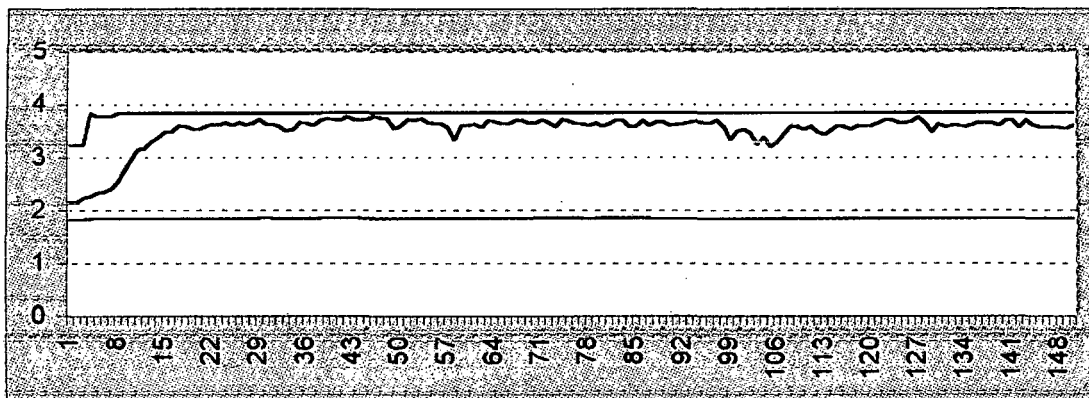
RESULTADOS DE APLICACIÓN DEL ALGORITMO GENETICO CANONICO EN OPTIMIZACION FUNCIONAL

Generación	Cromosoma	Valor de x	Fitness	Promedio De la población	Desviación estándar
1	1111000011101111101111	1.823475	3.226155	2.145023	0.559004
5	1111001001111010110100	1.84156	3.777205	2.327923	0.646315
10	1111001011101111111111	1.846924	3.838306	2.931693	0.743229
15	1111001011111111111011	1.84765	3.84262	3.491645	0.570487
20	1111001011111111111111	1.847656	3.84265	3.54079	0.484981
25	1111001011111101110110	1.847558	3.842125	3.603622	0.618695
47	1111001100111001111111	1.850311	3.850223	3.758076	0.23556
65	1111001100111001111110	1.850311	3.850222	3.64893	0.416796
81	1111001101110001110110	1.852868	3.845351	3.643504	0.457433
100	1111001101001001010111	1.851015	3.850074	3.503018	0.671811
150	1111001100111111110111	1.85058	3.850273	3.600401	0.491358

Se puede observar que los individuos mas aptos están siendo mejorados en las generaciones sucesivas encontrándose en la 150 ava generación un fitness de 3.850723 para un individuo con valor real de 1.85058.

GRAFICO NRO 11

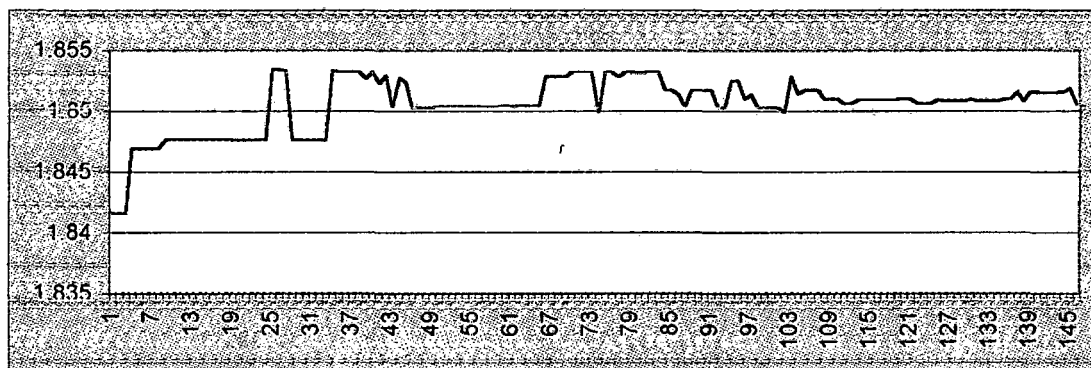
GRAFICO DE LOS VALORES DE FITNESS , PROMEDIO, VALOR DE X PARA EL CASO DEL ALGORITMO GENETICO CANONICO CON 150 GENERACIONES



La parte superior del grafico representa los valores de los Fitness, la línea de en medio es el valor promedio y el de la parte inferior son los valores de x a través de las distintas generaciones, es claro observar que los valores de aptitud se concentran tan pronto en las primeras generaciones y marcan una tendencia constante luego en cambio los valores promedio tienden a sufrir variaciones a lo largo del proceso evolutivo de los individuos. Analizaremos por separado en los siguientes cuadros los resultados de cada uno.

GRAFICO NRO 12

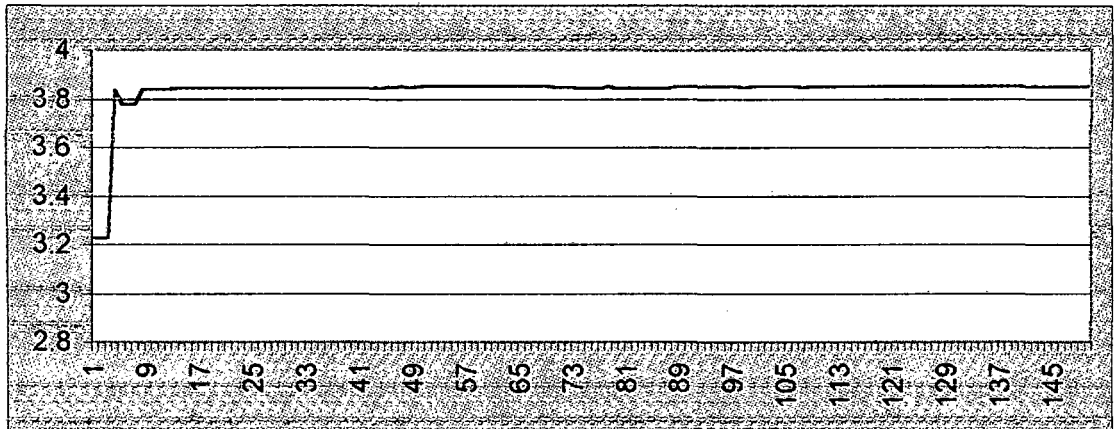
GRAFICO DE LOS VALORES VALORES DE X QUE TOMA EN TODO EL PROCESO EVOLUTIVO EN LA APLICACIÓN DEL ALGORITMO GENETICO CANONICO



En este grafico se presenta los distintos valores de la variable x representado en el gen correspondiente en el cromosoma de cada individuo, una variación y tendencia hacia el valor que toma x para que la función sea máxima.

GRAFICO 13

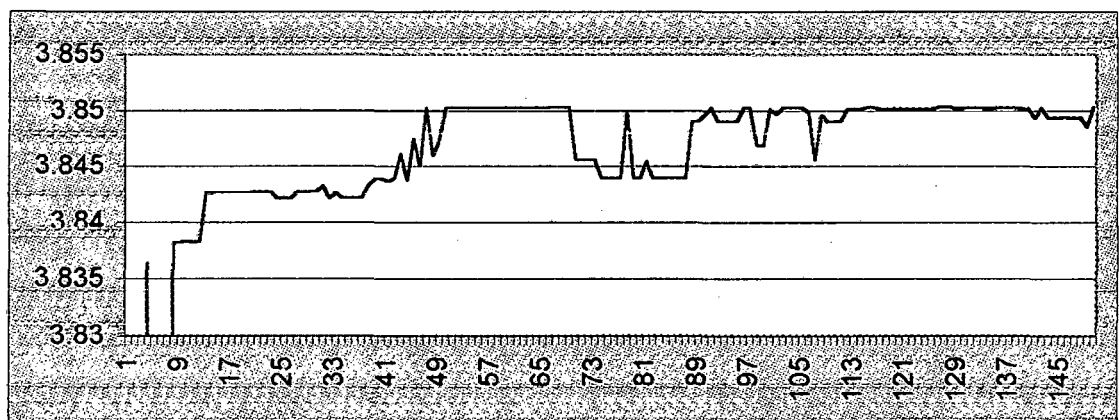
GRAFICO DE LOS VALORES QUE TOMA EL VALOR DE APTITUD DE CADA INDIVIDUO EN LA APLICACIÓN DEL ALGORITMO GENETICO CANONICO



Se observa que a partir de la 10 generación una tendencia hacia el valor que se esta buscando en la función a optimizar, la aparente invariabilidad del valor fitness se puede analizar en el siguiente cuadro mas detalladamente.

GRAFICO NRO 14

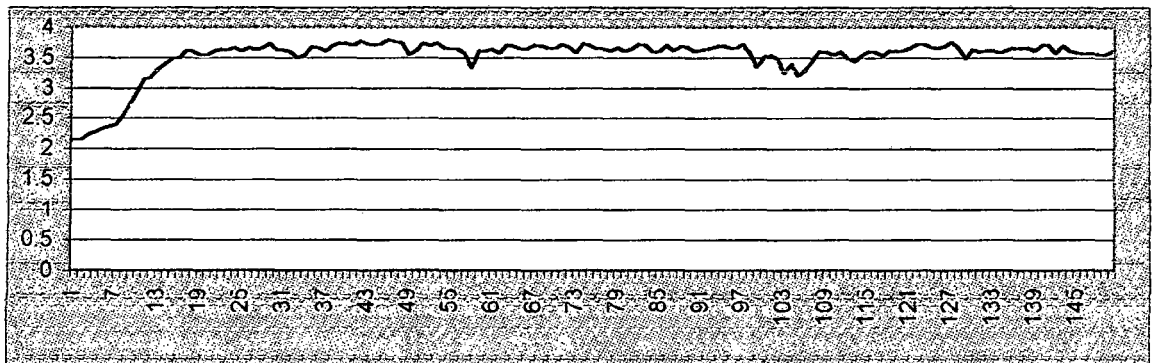
VARIACIÓN DEL VALOR FITNES A LO LARGO DEL PROCESO EVOLUTIVO EN EL ALGORITMO GENETICO CANONICO



En el grafico Nro 13 observamos que existe variación el cuando a los mejores valores de generación en generación se observa a de más una tendencia creciente a obtener el valor optimo en la maximización.

GRAFICO NRO 15

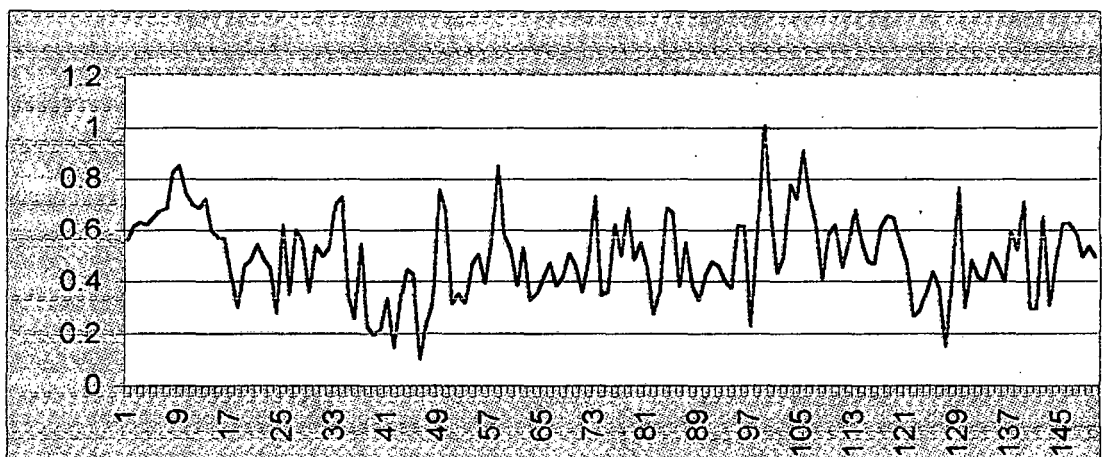
GRAFICO DEL PROMEDIO DE APTITUD EN CADA POBLACIÓN DE TODO EL PROCESO EVOLUTIVO EN EL ALGORITMO GENETICO CANONICO



El valor promedio de las aptitudes de cada población se ha encontrado que tiene una variación leve tal como se muestra en el grafico, los picos hacia arriba representan los puntos en los que la población ha encontrado valores máximos, en igual forma existe una tendencia hacia el grupo de valores en el que se encuentra el punto máximo de función que tratamos de encontrar.

GRAFICO NRO 16

GRAFICO DE LA DIVERSIDAD DE LA CALIDAD DE LOS INDIVIDUOS EN CADA POBLACIÓN EN TODO EL PROCESO EVOLUTIVO DEL ALGORITMO GENETICO CANONICO



La diversidad de los valores fitness de los individuos de cada una de las poblaciones en cada generación se muestra ampliamente diversificada, los picos hacia arriba

representan cuando existe bastante heterogeneidad y en el caso en que este es mínimo representa que la población tiene muy poca diversidad.

3.5 RESULTADOS CON LA INCORPORACIÓN DE LOS OPERADORES DE CLONACION Y MUTACIÓN Y LA INCORPORACIÓN DE LOS PRINCIPIOS DE INGENIERIA GENETICA EN EL ALGORITMO GENETICO CANONICO

En este caso se aplico el algoritmo genético canónico incorporando los operadores que estamos estudiando, Clonación y Mutación, se encontró los siguientes resultados para los mismos parámetros genéticos de tamaño de población y numero de generaciones, obteniéndose los siguientes resultados.

PROCEDIMIENTO DE APLICACIÓN

Primero se aplico el procedimiento de clonación simple tal como se utiliza en el caso canónico del Algoritmo Genético, esto es utilizando un numero aleatorio para cambiar el valor de un bit en el cromosoma del individuo. Obteniéndose los siguientes cuadros de los resultados.

CUADRO NRO 3

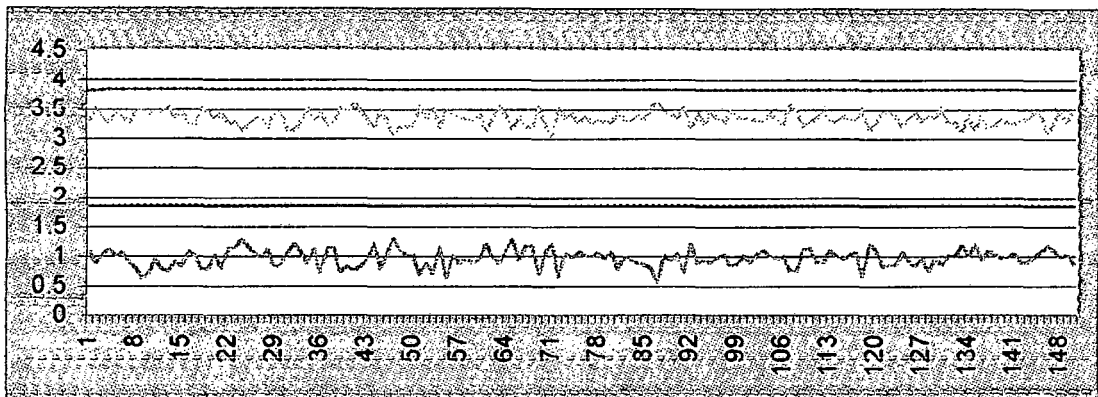
CUADRO DEL PROCESO EVOLUTIVO PARA EL CASO DE MUTACIÓN SIMPLE DE UN BIT HASTA QUE SE OBTENGA UN DETENIMIENTO EVOLUTIVO.

	Cromosoma	Valor	X	fitness	promedio	desviación
1	1111001010111110101111	3977135	1.844669	3.818862	3.277593	1.076599
2	1111001010111111101111	3977199	1.844715	3.819348	3.518514	0.882184
3	1111001011111111101111	3981295	1.847645	3.842589	3.380146	1.050584
4	1111001011111111111111	3981311	1.847656	3.84265	3.280516	1.115071
5	1111001011111111111111	3981311	1.847656	3.84265	3.411249	1.009022
6	1111001011111111111111	3981311	1.847656	3.84265	3.381502	1.069553
7	1111001011111111111111	3981311	1.847656	3.84265	3.278813	0.918736
8	1111001011111111111111	3981311	1.847656	3.84265	3.461993	0.814259
9	1111001011111111111111	3981311	1.847656	3.84265	3.489596	0.636915
10	1111001011111111111111	3981311	1.847656	3.84265	3.418928	0.719179

Se observa en el cuadro que los cromosomas cambian de generación en generación un solo bit esto debido al proceso de mutación simple, también se observa que los valores fitness convergen en la cuarta población y de ahí en adelante se mantiene constante.

GRAFICO NRO 17

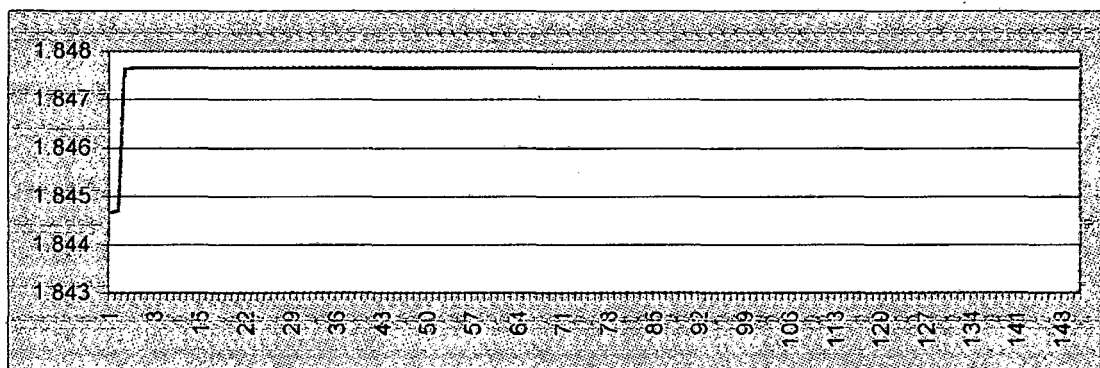
GRAFICO DEL PROCESO EVOLUTIVO PARA EL CASO DE MUTACIÓN SIMPLE CANONICA DEL AG CON ING. GENETICA



Se muestra en este grafico en primera linea el valor fitness luego el promedio de cada población, el valor de x y la variación de fitness en cada población todos ellos para cada ciclo evolutivo, se puede observar una amplia variación de los valores fitness promedio y el la diversidad de la población en cada generación. En tanto los valores de x y el fitness tiende a una convergencia muy rápida. Estas comparaciones los vamos a ver en los gráficos siguientes de manera individual.

GRAFICO NRO 18

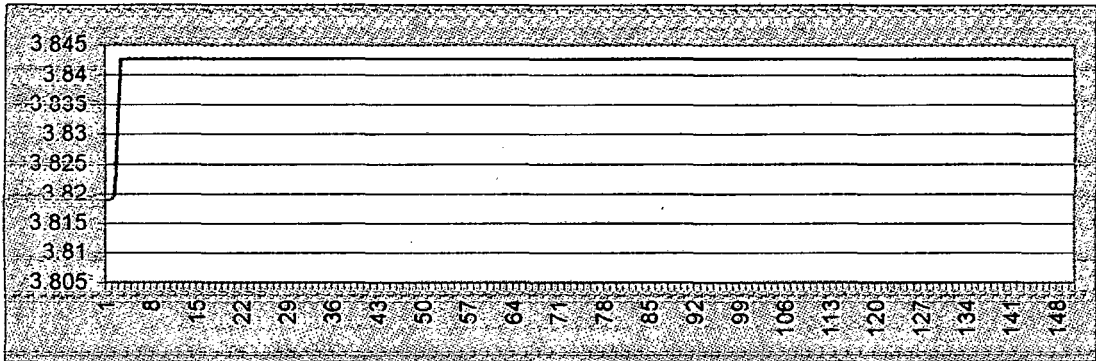
VALOR DE X EN CADA CICLO EVOLUTIVO PARA EL CASO DE MUTACIÓN SIMPLE DEL AG CON INGENIEIRA GENETICA



Se encontró que el valor de x que representa al individuo converge rápidamente apenas en la 4 generación manteniéndose constante de ahí en adelante.

GRAFICO NRO 19

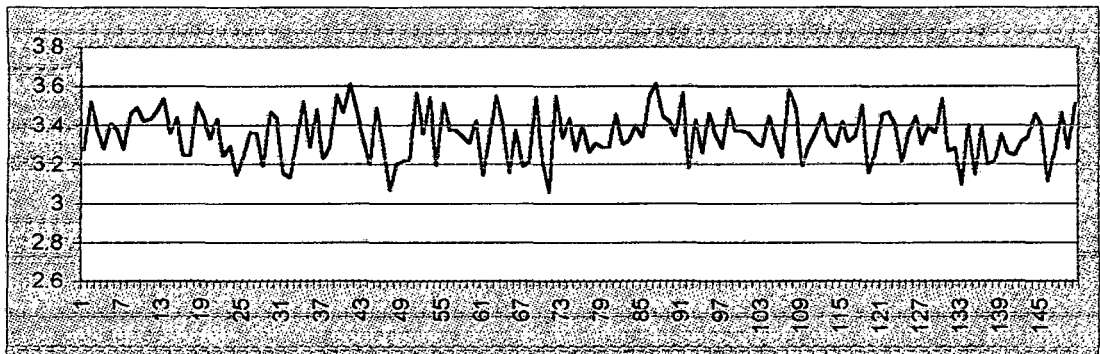
VALOR FITNESS EN CADA PROCESO EVOLUTIVO PARA EL CASO DE MUTACIÓN SIMPLE EN AG CON INGENIERIA GENETICA



En este grafico se puede observar que también los valores fitness convergen rápidamente al mismo ritmo que el valor de x, se detiene en 3.846

GRAFICO NRO 20

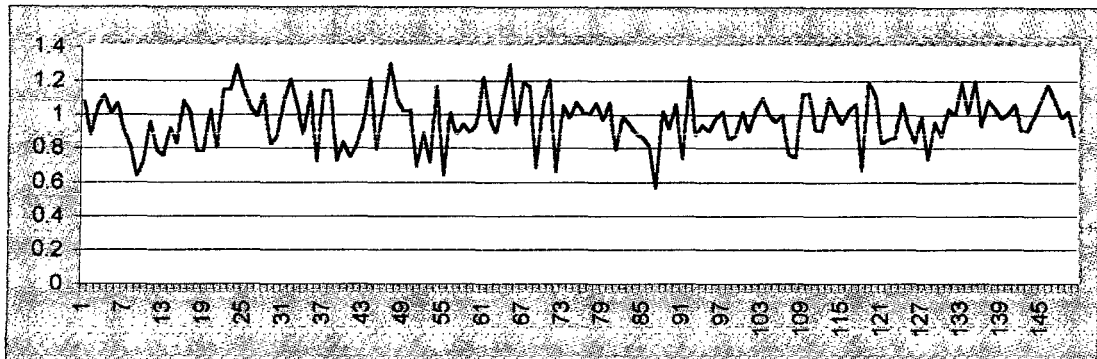
VALOR PROMEDIO DE FITNESS PARA MUTACIÓN SIMPLE DEL AG CON INGENIERIA GENETICA



En este grafico podemos observar que los valores promedio de los fitness tiene una variación de generación en generación lo que muestra que el proceso de mutación los ha hecho variar a los individuos en cada población de los que se ha escogido el mejor de ellos.

GRAFICO NRO 21

VALOR DE VARIACIÓN DE LOS FITNESS DE CADA GENERACIÓN PARA EL CASO DE MUTACIÓN SIMPLE DEL AG CON INGENIERIA GENETICA



La variación de los individuos que se muestra en cada generación se ve muy dispersa a muy homogénea es decir que en algunas poblaciones los individuos son muy similares y en otros caso tienen una variación esto se debe al operador de mutación que en forma aleatoria e irregular permite obtener diversidad en el material genético de los individuos.

3.6 COMPARACIÓN DE APLICACION DEL PROCESO DE INGENIERIA GENETICA EN OTRAS POBLACIONES PARA SU ANALISIS

Se ha realizado una replica del proceso evolutivo utilizando los operadores de ingeniería genética aplicado a otras población con el objeto de analizar la convergencia del valor fitness teniendo se los siguientes resultados.

CUADRO NRO 5

RESULTADOS DE LA PRIMERA POBLACIÓN COMPARATIVA

cromosoma	Valor	X	Fitness	promedio	desviación
1100000011000010011101	3158173	1.258902	3.209992	3.089784	0.612158
1101000011000010011101	3420317	1.446402	3.437171	3.124012	0.817589
1101000011100010011101	3422365	1.447867	3.444617	2.986233	0.976747
1101000011110010011101	3423389	1.448599	3.447197	3.092458	0.889273
1101000011110011011101	3423453	1.448645	3.447333	2.935357	1.073317
1101000011111011011101	3423965	1.449011	3.448312	3.018283	0.852424
110100001111111011101	3424221	1.449194	3.44873	3.110643	0.690166
1101000011111111111101	3424253	1.449217	3.448779	3.114335	0.673314
1101000011111111111111	3424255	1.449219	3.448782	3.07562	0.719371

Convergen a 3.448782, en este caso la primera población llega a una convergencia en la 9va generación para un valor $x = 1.449219$

CUADRO NRO 6

RESULTADOS DE LA SEGUNDA POBLACIÓN COMPARATIVA

cromosoma	Valor	X	fitness	promedio	desviación
1111000010101010001010	3943050	1.82029	3.083303	3.046944	0.345345
1111001010101010001010	3975818	1.843727	3.808044	3.204396	0.657343
1111001011101010001010	3979914	1.846657	3.836482	3.428753	0.674544
1111001011111010001010	3980938	1.847389	3.84118	3.483338	0.567454
1111001011111110001010	3981194	1.847573	3.842203	3.082325	0.453444
1111001011111111001010	3981258	1.847618	3.842449	3.457734	1.342322
1111001011111111101010	3981290	1.847641	3.84257	3.547133	1.032332
1111001011111111111010	3981306	1.847653	3.842631	3.357083	0.75343
1111001011111111111110	3981310	1.847655	3.842646	3.523741	0.678565
1111001011111111111111	3981311	1.847656	3.84265	3.403777	0.845445
1111001011111111111111	3981311	1.847656	3.84265	3.403777	0.845445

Convergen a 3.84265, en esta población en la generación numero 10 para un valor de $x = 1.847656$

CUADRO NRO 7

RESULTADOS DE LA TERCERA POBLACIÓN COMPARATIVA

cromosoma	Valor	X	fitness	promedio	desviación
1011111101101010111001	3136185	1.243175	3.214707	2.947987	0.643338
1011111111101010111001	3144377	1.249034	3.248459	2.944585	0.744265
1011111111101110111001	3144633	1.249217	3.24884	2.842011	0.791563
10111111111110111001	3145657	1.24995	3.249948	3.029167	0.579519
10111111111111111001	3145721	1.249996	3.249996	2.894958	0.805769
101111111111111111101	3145725	1.249998	3.249998	2.945226	0.597882
101111111111111111111	3145727	1.25	3.25	2.909415	0.605497
1011111111111111111111	3145727	1.25	3.25	2.866907	0.660165
1011111111111111111111	3145727	1.25	3.25	2.803364	0.730479
1011111111111111111111	3145727	1.25	3.25	2.844724	0.815293

Convergen a 3.25 en la generación numero 7 para un $x = 1.25$

De las replicas aplicadas a diversas poblaciones se ha encontrado que el operador de clonación y mutación aceleran la convergencia pero se ha encontrado el problema que los valores a los que convergen distan unos de otros es decir no se acercan al valor optimo de la función de evaluación como se muestra en el siguiente cuadro comparativo

CUADRO NRO 8

CUADRO COMPARTIVO DE LOS MEJORES INDIVIDUOS EN CADA POBLACIÓN EVALUADA APLICANDO LA MUTACIÓN SIMPLE DE INGENIEIRA GENETICA

Numero de Generaciones	Fitness	Cromosoma
9	3.448782	1101000011111111111111
10	3.84265	1111001011111111111111
7	3.250000	1011111111111111111111

En este cuadro se muestra que en las primeras generaciones es cuando converge el valor fitness, lo que significa una convergencia realmente rápida, sin embargo se ha encontrado que este valor al que convergen tiene el mismo esquema que el primer individuo que resulta ser mayor en la primera población, por lo que podemos decir que esto depende fundamentalmente de la primera generación, si en la primera generación existe un buen esquema entonces este permanece y en base a el se logra mejorar hasta encontrar el mejor, de lo contrario si en la primera generación el esquema no es bueno este también se mantiene. Esto se ve en el siguiente cuadro comparativo de cada replica con la primera generación y el mejor individuo de esta.

CUADRO NRO 9

CUADRO COMPARATIVO DEL PRIMER MEJOR INDIVIDUO CON EL VALOR DE CONVERGENCIA EN LA PRIMERA POBLACIÓN

cromosoma	Valor	X	fitness	Promedio	desviación
1100000011000010011101	3158173	1.258902	3.209992	3.089784	0.612158
1101000011111111111111	3424255	1.449219	3.448782	3.116027	0.757542

Se muestran 8 bits de cambiaron igual al numero de generación en que empieza la convergencia menos uno

CUADRO NRO 10

CUADRO COMPARATIVO DEL PRIMER MEJOR INDIVIDUO CON EL VALOR DE CONVERGENCIA EN LA SEGUNDA POBLACIÓN

Cromosoma	Valor	X	fitness	promedio	desviación
1111000010101010001010	3943050	1.82029	3.083303	3.046944	0.345345
1111001011111111111111	3981311	1.847656	3.84265	3.403777	0.845445

Se muestran 9 bits de cambiaron igual al 10 -1 generaciones de evolución a partir del cual converjan los valores fitness.

CUADRO NRO 11

CUADRO COMPARATIVO DEL PRIMER MEJOR INDIVIDUO CON EL VALOR DE CONVERGENCIA PARA TERCER POBLACIÓN

cromosoma	Valor	X	fitness	Promedio	desviación
1011111101101010111001	3136185	1.243175	3.214707	2.947987	0.643338
1011111111111111111111	3145727	1.250000	3.250000	2.844724	0.815293

Se muestran 6 bits de cambio igual al 6 +1 que es el numero de generación en que converge el valor fitness de la población.

CONCLUSIÓN

Estos resultados podemos establecer la siguiente ecuación que relaciona el numero de generación con el numero de genes cambiados en el cromosoma del individuo después de aplicar los operadores de clonación y mutación.

$$n_g = n_c + 1$$

donde :

n_g = Numero de generación en que comienza el proceso de convergencia y a partir del cual no se tiene ningún cambio genético de mejoramiento.

n_c = Numero de bits de cambio en el cromosoma del mejor individuo a partir de la primera población hasta la generación en que se inicia la convergencia.

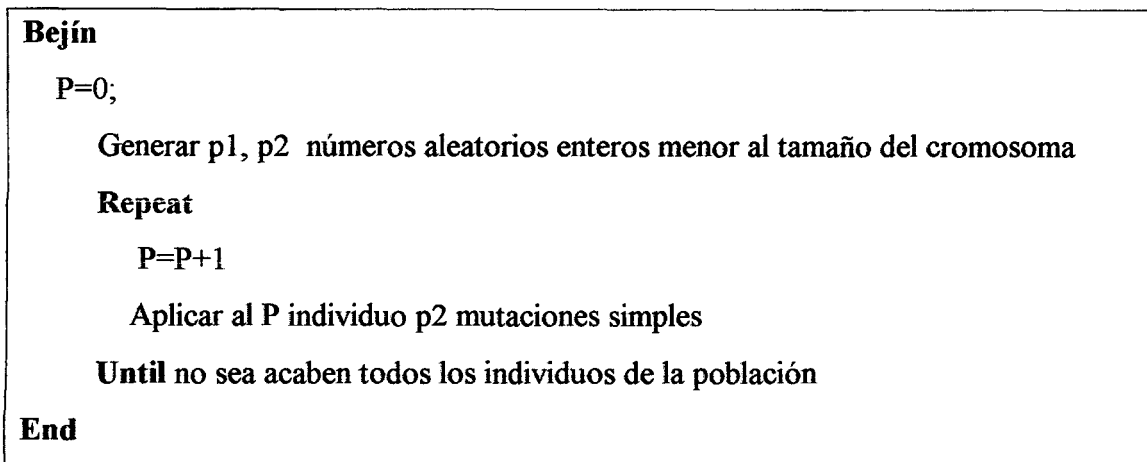
Estos resultados no ha permitido comparar detenidamente los cambios producidos en cada generación y en cada individuo encontrándose que el operador simple de mutación cambia una parte mínima de información genética, sin embargo el proceso evolutivo no siempre se da en un solo aspecto por lo que hemos replanteado el operador para que permita mejores resultados por lo que la mutaciones en mas de un alelo ha sido la primera variación del planteamiento inicial del operador de mutación y se ha considerado una cantidad aleatoria de mutaciones por cada cromosoma.

DIAGRAMA NRO 3

ALGORITMO GENETICO CON MUTACIÓN SIMPLE

```
begin
  generar población inicial aleatoria G(0);
  t:=0;
  repeat
    t:=t+1;
    Evaluar G(t); // aplicando fitness
    Seleccionar el mejor;
    Clonar a todos los individuos de la población con el material
      genético del mejor individuo;
    Aplicar el operador de mutación Múltiple
    Hacer reporte;
  until encontrar una solución satisfactoria
end
```

DIAGRAMA NRO 4
ALGORITMO DE MUTACIÓN MÚLTIPLE



3.7 RESULTADOS DE APLICACIÓN DEL OPERADOR DE MUTACIÓN MÚLTIPLE.

Luego de aplicar este algoritmo se ha encontrado los siguientes resultados del operador de mutación múltiple aplicados a 5 poblaciones diferentes, para cada caso para la presentación se ha eliminado las generaciones en los que no se ha encontrado cambios mostrándose únicamente las generaciones en los que ha habido cambio evolutivo de mejoramiento.

CUADRO NRO 12

CUADRO DE RESULTADOS DE APLICACIÓN DEL OPERADOR DE
MUTACIÓN MÚLTIPLE EN LA PRIMERA POBLACIÓN

Gener.	Cromosoma	Valor Gen	X	Fitness	Promedio	Variacion
1	1110001000100110001001	3705225	1.650184	3.650156	2.617287	1.06913
2	1111001011100010001001	3979401	1.84629	3.833764	1.958347	0.780191
4	1111001011110010001001	3980425	1.847022	3.838948	2.620573	1.089104
5	1111001011110100010011	3980563	1.847121	3.838572	2.01431	0.922295
11	1111001101110101000011	3988803	1.853015	3.844709	3.071254	1.299087
12	1111001100110101000011	3984707	1.850085	3.850079	1.903104	0.849851
19	1111001100111101000011	3985219	1.850451	3.850265	2.061315	0.986555
22	1111001100111111010001	3985361	1.850553	3.850274	1.716473	0.858134
23	1111001100111111010001	3985361	1.850553	3.850274	2.778031	1.282961

En esta población en la generación numero 22 se encontró el valor optimo = 3.850274 para $x = 1.850553$.

CUADRO NRO 13
CUADRO DE RESULTADOS DE APLICACIÓN DEL OPERADOR DE
MUTACIÓN MÚLTIPLE EN LA SEGUNDA POBLACIÓN

Gener.	Cromosoma	Valor Gen	X	Fitness	Promedio	Variacion
1	1110001001110100011011	3710235	1.653768	3.642197	2.666845	1.24264
2	1111001011110100011011	3980571	1.847127	3.839608	2.040122	0.970006
6	1111001101110100010011	3988755	1.852981	3.844863	2.154914	0.944176
7	1111001100011101100000	3983200	1.849007	3.848108	2.31342	1.066103
8	1111001100111101100101	3985253	1.850476	3.850269	1.98469	0.86841
10	1111001100111111100101	3985381	1.850567	3.850273	2.126735	0.807107
11	1111001100111111100101	3985381	1.850567	3.850273	1.852524	0.930676
12	1111001100111111100101	3985381	1.850567	3.850273	1.990928	0.890854

En esta población en la generación numero 10 se encontró el valor optimo = 3.850273 para $x = 1.850567$.

CUADRO NRO 14
CUADRO DE RESULTADOS DE APLICACIÓN DEL OPERADOR DE
MUTACIÓN MÚLTIPLE EN LA TERCERA POBLACIÓN

Gener.	Cromosoma	Valor Gen	X	Fitness	Promedio	Variation
1	1111010011010100100001	4011297	1.869104	3.542463	2.181695	0.76406
2	1111000111111111010010	3964882	1.835905	3.658843	2.191841	0.864109
3	1111001111001001101001	3994217	1.856887	3.81359	1.869226	0.955451
4	1111001011011011111000	3979000	1.846003	3.831471	2.033171	0.918867
8	1111001011111011111000	3981048	1.847468	3.841627	2.916331	1.250492
9	1111001011111111111010	3981306	1.847653	3.842631	2.160349	1.127372
11	1111001101011101100110	3987302	1.851941	3.848498	2.42634	1.277301
12	1111001101001101100110	3986278	1.851209	3.849874	2.699074	1.345494
13	1111001101000001100110	3985510	1.85066	3.850262	2.138497	1.176976
17	1111001101000001000000	3985472	1.850632	3.850267	1.910193	0.933129
21	1111001101000000100001	3985441	1.85061	3.85027	2.620156	1.086124
22	1111001101000000100001	3985441	1.85061	3.85027	3.081889	1.142348

En esta población en la generación numero 21 se encontró el valor optimo = 3.85027 para $x = 1.85061$.

CUADRO NRO 15

CUADRO DE RESULTADOS DE APLICACIÓN DEL OPERADOR DE MUTACIÓN MÚLTIPLE EN LA CUARTA POBLACIÓN

Genr.	Cromosoma	Valor Gen	X	Fitness	Promedio	Variation
1	1001111010111000111111	2600511	0.860031	2.81768	2.174826	0.743353
2	1111000111100001110110	3962998	1.834558	3.622873	2.022042	0.936168
3	1111001011100000000000	3979264	1.846192	3.832997	2.230401	0.954896
4	1111001100100001011011	3983451	1.849187	3.848584	2.021436	0.813165
6	1111001100110001111001	3984505	1.849941	3.849938	2.167636	0.844734
8	1111001100110100100001	3984673	1.850061	3.850058	2.186522	0.995402
9	1111001101000110100101	3985829	1.850888	3.850168	2.346647	1.185918
14	1111001101000000100100	3985444	1.850612	3.85027	1.823241	0.991099
17	1111001101000000000110	3985414	1.850591	3.850272	1.892813	0.74358
18	1111001101000000000110	3985414	1.850591	3.850272	2.031877	1.039222

En esta población en la generación numero 17 se encontró el valor optimo = 3.850272 para $x = 1.850591$.

CUADRO NRO 16

CUADRO DE RESULTADOS DE APLICACIÓN DEL OPERADOR DE MUTACIÓN MÚLTIPLE EN LA QUINTA POBLACIÓN

Genr.	Cromosoma	Valor Gen	X	Fitness	Promedio	Variation
1	1011111000011010100101	3114661	1.22778	2.940581	1.873496	0.763702
2	11001111111111111110101	3407661	1.437493	3.327944	1.925552	0.771342
3	1101000101011011001010	3430090	1.453392	3.445147	2.04155	0.858798
4	11110011010111111111110	3987454	1.85205	3.84821	2.143369	1.207819
5	1111001101011011011100	3987164	1.851843	3.848741	2.125784	1.066473
6	1111001100111010101100	3985068	1.850343	3.850236	1.873243	0.692619
9	1111001100111011111101	3985149	1.850401	3.850254	2.40494	0.804945
16	1111001100111111111101	3985405	1.850584	3.850273	1.848875	0.82825
29	1111001100111111100001	3985377	1.850564	3.850274	2.415316	1.199053
30	1111001100111111100001	3985377	1.850564	3.850274	3.009416	1.137498

En esta población en la generación numero 29 se encontró el valor optimo = 3.850274 para $x = 1.850547$.

CUADRO NRO 17

CUADRO DE RESULTADOS DE APLICACIÓN DEL OPERADOR DE MUTACIÓN MÚLTIPLE EN LA SEXTA POBLACIÓN

Genr.	Cromosoma	Valor Gen	X	Fitness	Promedio	Variation
1	1111010001001001110001	4002417	1.862752	3.715252	3.105791	1.136888
2	1111010000001001110001	3998321	1.859823	3.77197	2.081183	0.978013
3	111101000000010110100	3997876	1.859504	3.777225	2.707961	1.072351
4	1111010000000001110100	3997812	1.859459	3.777966	2.303935	1.071558
6	1111001010000001110100	3973236	1.841881	3.782283	2.084735	0.733359
7	1111001011000010001110	3977358	1.844829	3.820537	2.127719	0.912968
9	1111001101011101101100	3987308	1.851946	3.848487	2.840589	1.194735
10	1111001101010101101100	3986796	1.851579	3.849301	2.222982	1.080168
11	1111001101000101100110	3985766	1.850843	3.850194	2.266761	1.0255
16	1111001101000001100110	3985510	1.85066	3.850262	2.185809	1.001758
31	1111001101000001100010	3985506	1.850657	3.850263	2.558292	1.115131
45	1111001101000001010100	3985492	1.850647	3.850265	2.177408	0.99684
51	1111001101000001001100	3985484	1.850641	3.850266	2.609788	1.251296
58	1111001101000001000101	3985477	1.850636	3.850267	1.937171	0.933666
68	1111001101000000000111	3985415	1.850592	3.850272	2.133419	0.885566
69	1111001101000000000111	3985415	1.850592	3.850272	2.129029	0.915948

En esta población en la generación numero 68 se encontró el valor optimo = 3.850272 para $x = 1.850592$.

CONCLUSIONES DE LA APLICACIÓN DEL OPERADOR DE MUTACIÓN MÚLTIPLE.

De todos los cuadros presentados podemos observar que en todas las poblaciones analizadas siempre convergen al valor optimo con lo que hemos resuelto el problema que la mutación simple no podía resolver, sin embargo tenemos que señalar que en algunos casos esta convergencia puede ser un poco lenta pero en comparación con el método canónico es realmente rápida.

El siguiente cuadro muestra la comparación de cada población y el numero de generación en que se obtiene el mejor resultado.

CUADRO NRO 18

COMPARACIÓN DE LOS RESULTADOS DE LAS DIFERENTES POBLACIONES ANALIZADAS EN EL QUE SE HA UTILIZADO EL OPERADOR DE MUTACIÓN MÚLTIPLE.

Pob.	Gen	Cromosoma	Valor Gen	X	Fitness	Promedio	Variation
1	22	1111001100111111010001	3985361	1.850553	3.850274	1.716473	0.858134
2	10	111100110011111100101	3985381	1.850567	3.850273	2.126735	0.807107
3	21	1111001101000000100001	3985441	1.85061	3.850270	2.620156	1.086124
4	17	1111001101000000000110	3985414	1.850591	3.850272	1.892813	0.74358
5	29	1111001100111111100001	3985377	1.850564	3.850274	2.415316	1.199053
6	68	1111001101000000000111	3985415	1.850592	3.850272	2.133419	0.885566

En todas las poblaciones siempre se llega al valor óptimo si bien esto no concuerdan en el último dígito esto se debe a que la precisión se pierde en el último dígito como en cualquier proceso de medición experimental, el número de generaciones sigue siendo menor a la aplicación del Algoritmo Genético Canónico que en muchos casos tiene que ser más de 100 o 1000 generaciones por lo menos, para lograr un valor aceptable de fitness, como en el caso presentado del AG canónico tenemos que en 150 generaciones se tiene similar valor de fitness.

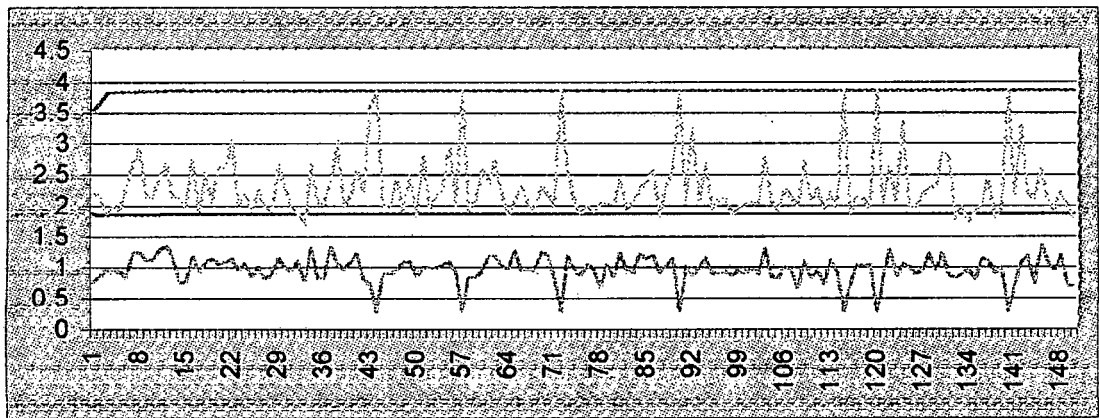
Aplicando los operadores de clonación y mutación se encontró que en la generación número 10 o menos ya se encontró una mejor solución por lo que se comprueba como mejor desempeño del algoritmo con estos operadores de Ingeniería Genética.

3.8 RESULTADOS DE UN CASO DE UNA POBLACIÓN TÍPICA CON APLICACIÓN DE INGENIERIA GENETICA.

Para realizar un análisis más exhaustivo se presenta los siguientes cuadros sobre la población número tres en el que se ha aplicado los principios de Ingeniería Genética con el operador de mutación múltiple.

GRAFICO NRO 22

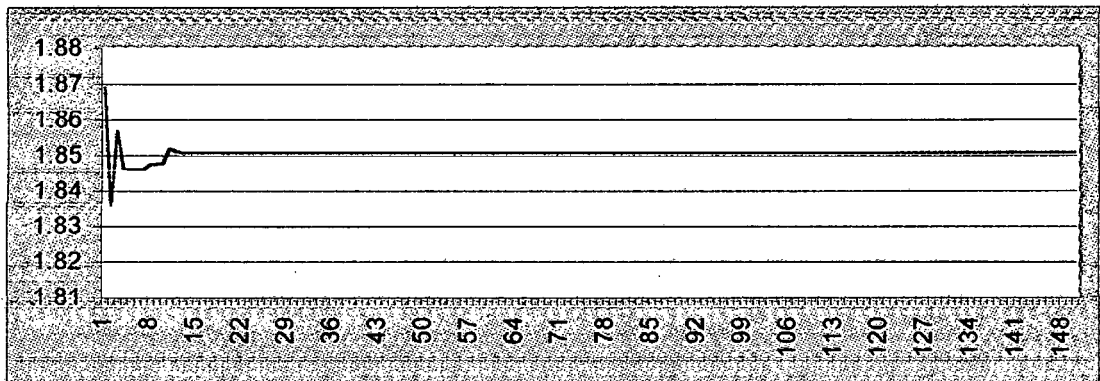
RESULTADOS DE APLICACIÓN DE ING. GENETICA CON OPERADOR DE MUTACIÓN MÚLTIPLE



Para la población numero 3 se ha encontrado una gran variación en cuanto al valor promedio fitness y la variación de los individuos, en cambio los valores de x y fitness se observa una convergencia rápida.

GRAFICO NRO 23

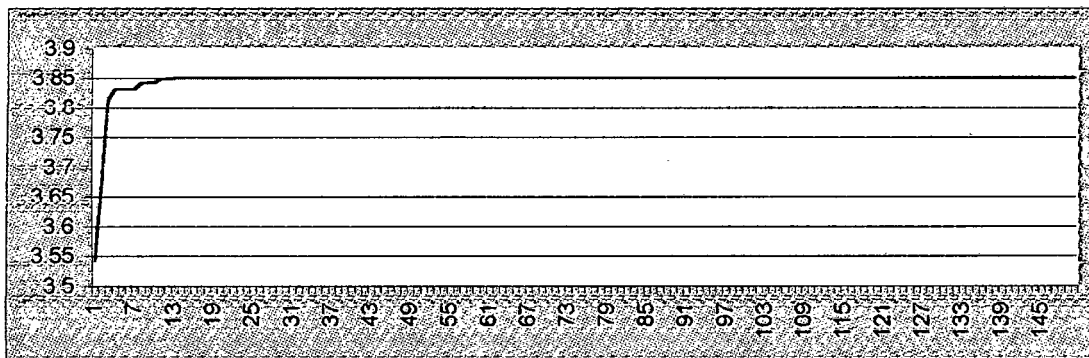
RESULTADOS DEL VALOR DE X EN APLICACIÓN DEL AG CON ING. GENETICA Y MUTACIÓN MÚLTIPLE



El valor de X sufre variaciones por exceso y por defecto alrededor del valor buscado y converge rápidamente antes de la generación numero 15.

GRAFICO NRO 24

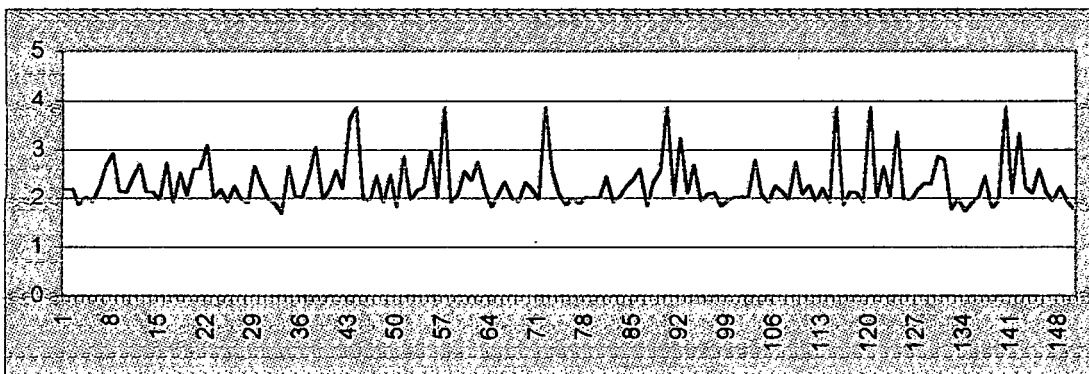
RESULTADOS DEL VALOR FITNESS EN APLICACIÓN DEL ALGORITMO GENETICO CON INGENIERIA GENETICA - MUTACIÓN MULTIPLE



La convergencia del valor fitness hacia el valor buscado converge rápidamente y a partir de la población 15 tiende a una variación menor.

GRAFICO NRO 25

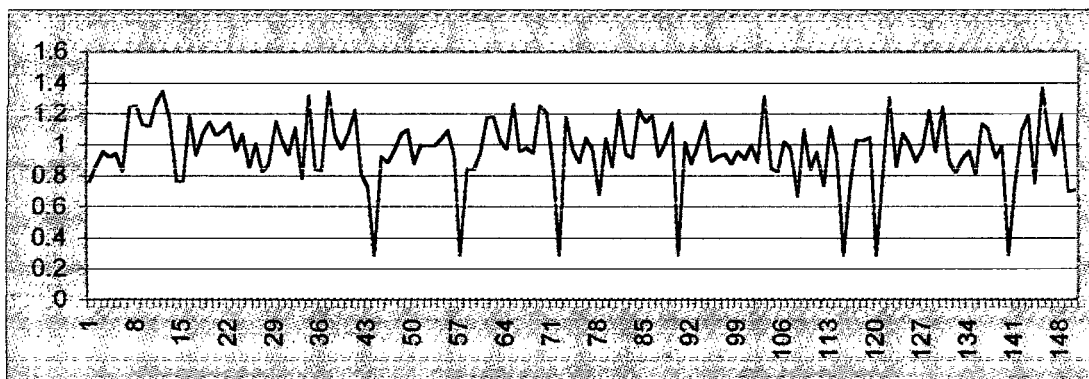
RESULTADOS DEL VALOR PROMEDIO FITNESS EN APLICACIÓN DEL ALGORITMO GENETICO CON INGENIERIA GENETICA - MUTACIÓN MULTIPLE



Los valores promedio de los fitness en cada generación de todo el proceso genético muestran variaciones a lo largo de las generaciones, los picos muestran las poblaciones en donde se ha obtenido un nuevo individuo mejor que el anterior registrado.

GRAFICO NRO 26

RESULTADOS DEL VALOR PROMEDIO FITNESS EN APLICACIÓN DEL ALGORITMO GENETICO CON INGENIERIA GENETICA - MUTACIÓN MÚLTIPLE



Los resultados de este grafico ilustran que los picos mas bajos corresponden cuando en la generación respectiva los individuos tienen mayor semejanza por lo tanto su variabilidad es mucho menor, en general se tiene una gran diversificación de las poblaciones.

Se muestra en el anexo los reportes completos sobre el proceso de clonación y mutación

3.9 RESULTADOS DE APLICACIÓN DE LOS PRINCIPIOS DE INGENIERIA GENETICA AL CASO DE UNA FUNCION DE DOS VARIABLES INDEPENDIENTES

La función de evaluación determinada es: $Z = 21.5 + x \sin(4\pi x) + y \sin(20\pi y)$

CUADRO 19
RESULTADOS DE APLICACIÓN DE INGENIERIA GENETICA EN OPTIMIZACION FUNCIONAL EN CASO DOS VARIABLES

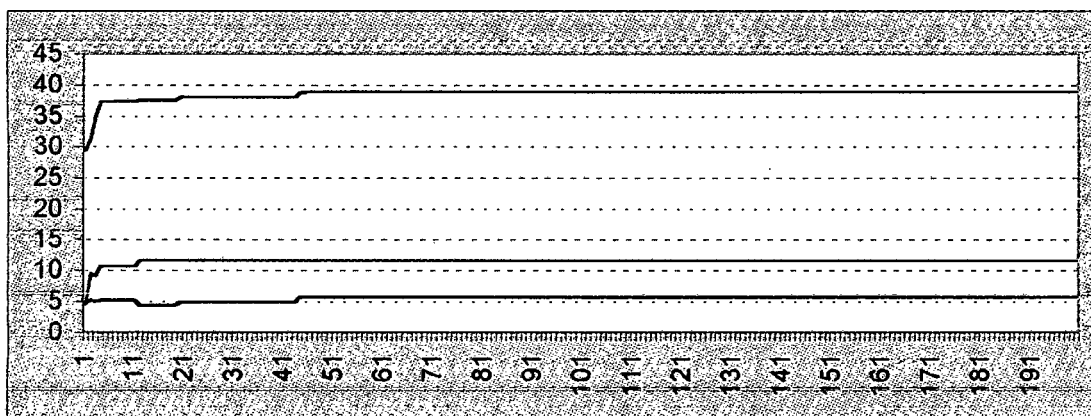
Gen	Cromosoma	X	Y	Fitenes	Promedio	Variación
1	100000010000100010011110100100101	4.610972	4.912101	29.424641	23.941882	6.460311
2	110101010000001101101010111011101	9.564469	5.240407	31.399279	24.271684	6.762532
3	110011010000001001100010110011101	9.092361	5.02458	34.861063	22.787534	9.201124
4	111001110011011001101010010101010	10.637942	5.224479	37.219239	23.478728	8.419375
6	111001110011010001101010010101010	10.637481	5.224479	37.228583	25.373726	7.522613
7	111001110010010011101010010101010	10.63391	5.224479	37.289001	25.22873	8.539726
8	111001110000010011101010010101111	10.626537	5.224738	37.348587	24.517703	8.363757
12	111101111111010110001100010100111	11.625762	4.427424	37.501405	23.95059	7.485337
15	111101111111010100001100010100111	11.625646	4.427424	37.501438	22.694342	7.746466
16	111101111111001100001100001100110	11.625185	4.424052	37.541354	24.422462	8.584006
19	111101111111001110001100001101110	11.625301	4.424467	37.547201	23.20563	7.963037
20	111101111111001010011111000101110	11.62507	4.925849	38.043905	24.689367	7.603523
21	111101111111001010011111000101100	11.62507	4.925745	38.04541	25.032204	9.097056
22	111101111111001010011111000100100	11.62507	4.92533	38.049335	23.860865	8.312742
24	111101111111001010011111000100000	11.62507	4.925123	38.050042	24.459335	7.218224
31	111101111111001110011111000100000	11.625301	4.925123	38.050194	24.261302	8.860175
44	111101111111001110111101000110000	11.625301	5.722852	38.796041	22.14176	9.804804
45	111101111111001110111101001110000	11.625301	5.726173	38.835854	24.136357	8.625238
46	111101111111001110111101001010000	11.625301	5.724512	38.847045	24.742361	8.837615
49	111101111111001110111101001011000	11.625301	5.724928	38.850086	22.475253	9.306745
53	111101111111001110111101001011001	11.625301	5.724979	38.850192	24.115482	9.072431
87	111101111111001111111101001011001	11.625358	5.724979	38.850215	24.057659	9.642984
102	111101111111001111111101001011011	11.625358	5.725083	38.850245	22.313263	8.204138
108	111101111111001111111101001011010	11.625358	5.725031	38.850261	24.341034	9.709928
109	111101111111001111111101001011010	11.625358	5.725031	38.850261	24.341034	9.709928

En este cuadro podemos observar como converge las soluciones encontrándose el optimo en la generación 108

Gen	Cromosoma	X	Y	Fitenes	Promedio	Variación
108	111101111111001111111101001011010	11.625358	5.725031	38.850261	24.341034	9.709928

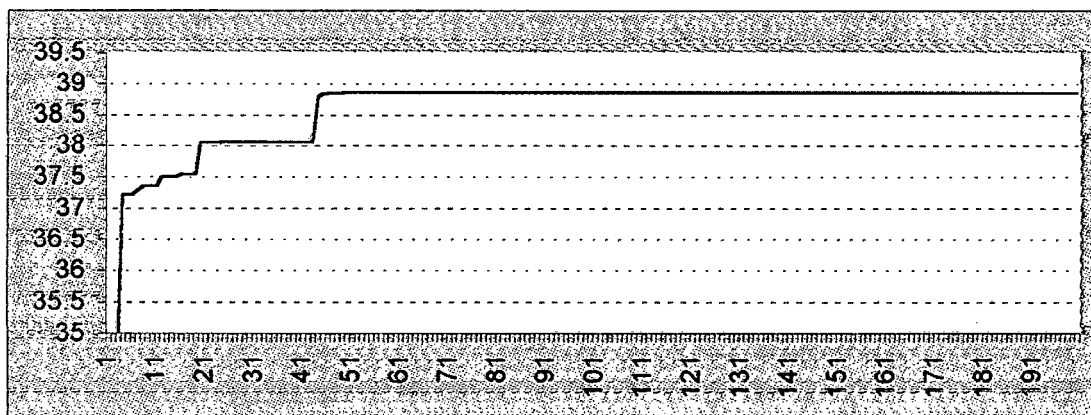
Este valor reportado es el valor que con mayor exactitud que el aplicado con un algoritmo genético simple, tal como lo publica Michalewicz encontró que este valor es de 35.477938 después de aplicar 1000 generaciones, con esto demostramos la robustez y mejor desempeño de los operadores de clonación y mutación para encontrar en valor optimo en una función.

GRAFICO 28
VALOR FITNESS, X , Y EN EL PROCESO EVOLUTIVO AL APLICAR
INGENIERIA GENETICA - MUTACIÓN MÚLTIPLE.



Se puede observar la rápida convergencia de los valores fitness, x e y

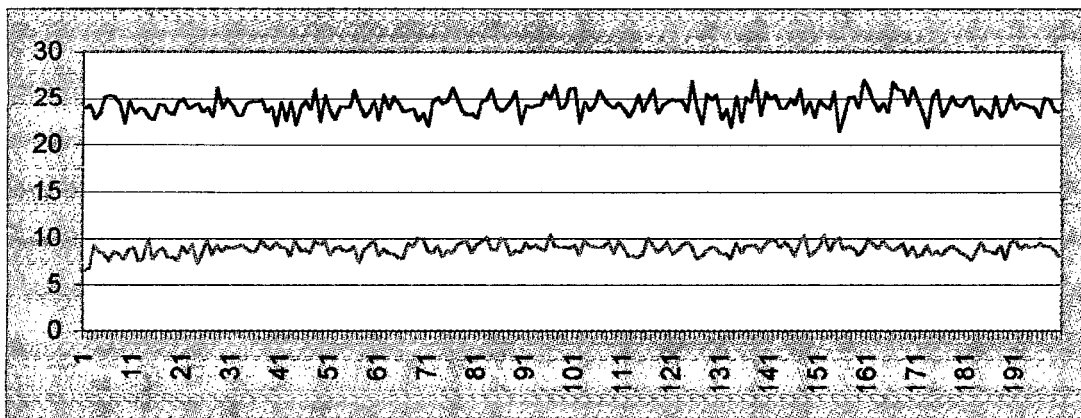
GRAFICO 29
VALORES DE FITNESS EN EL PROCESO EVOLUTIVO APLICANDO
INGENIERIA GENETICA - MUTACIÓN MÚLTIPLE



Al afinar la escala de los valores de Fitness se observa que hay una evidente convergencia antes de la 15 generación.

GRAFICO 30

VALORES PROMEDIO FITNESS Y VARIACIÓN DE LOS INDIVIDUOS EN
CADA GENERACIÓN AL UTILIZAR INGENIERIA GENETICA – MUTACIÓN
MULTIPLE



La variación es evidente en cada uno de las poblaciones a lo largo del proceso evolutivo.

IV CONCLUSIONES

Primera. Se ha superado los resultados de optimización encontrado por Michalewicz.

Segunda. El incorporar los principios de la ingeniería genética a través del operador de clonación y mutación influye en el funcionamiento del algoritmo genético.

Tercera. Se ha determinado como funciones de evaluación :

$$Y = x \sin(10\pi x) + 1$$

$$Z = 21.5 + x \sin(4\pi x) + y \sin(20\pi y)$$

Los que cumplen los requisitos de facilidad de implementación y tener pruebas de su desempeño.

Cuarta. El operador de clonación tiene un comportamiento similar al que se da en la ingeniería genética utilizándose para la implementación del algoritmo genético una duplicación de la cadena binaria de representación del genotipo del mejor individuo para que preserve las características en la siguiente generación.

Quinta. El operador de mutación ha tenido un comportamiento que ha permitido la alteración del material genético representado en el cromosoma del mejor individuo. Y se ha determinado dos tipos, el tipo simple el que cambia un bit de la cadena binaria y el tipo múltiple el que lo hace en más de un bit, determinados aleatoriamente. Encontrándose como mejor operador el de Mutación Múltiple

Sexta. Se encontró como ventajas de utilizar los principios de la Ingeniería genética de acelerar la convergencia a la mejor solución óptima y permitir una mayor robustez del desempeño del Algoritmo Genético. Encontrándose siempre a un valor óptimo determinado y en un mejor número de generaciones.

V RECOMENDACIONES

Primera. Recomendamos realizar la investigación utilizando algoritmos genéticos paralelos para evaluar su desempeño y funcionamiento.

Segunda. Investigar la incorporación de los principios de la Ingeniería Genética en el funcionamiento de algoritmos genéticos celulares

Tercera. Analizar el comportamiento de la cruce de dos puntos conjuntamente con el operador de clonación y mutación

Cuarta. Recomendamos que se pueda probar los operadores de clonación y mutación en problemas como el TSP y otros casos para estudiar, analizar sus resultados.

Quinta. Ampliar el presente trabajo de investigación a problemas de optimización de otros casos.

Sexta. A los futuros tesisas incorporar en sus investigaciones los procedimientos evolutivos y experimentar los operadores de mutación y clonación en lo diversos procesos computaciones que tengan que realizar.

VI BIBLIOGRAFIA CITADA

- 1 AARTS, E.H.L. Y KORST, J.H.M. (1989) Simulated Annealing and Boltzmann Machines, Wiley, Chichester.
- 2 CHAKRABORTY, U.K. Y DASTIDAR, D.G. (1993) Using reliability analysis to estimate the number of generations to convergence in genetic algorithms, Information Processing Letters, 4B, 199-209.
- 3 DARWIN, C. (1859) On the Origin of Species by Means of Natural Selection, Murray, London.
- 4 DAVIS, T.E. Y PRINCIPE, J.C. (1993) A Markov chain framework for the simple genetic algorithm, Evolutionary Computation, 1(3), 269-288.
- 5 DE JONG, K.A. (1975) An analysis of the behaviour of a class of genetic adaptive systems, Tesis doctoral, University of Michigan.
- 6 FOGEL, L.J. (1962) Autonomous automata, End. Res., 4, 14-19.
- 7 GOLDBERG, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA.
- 8 GOLDBERG, D.E. Y RICHARDSON, J.T. (1987) Genetic algorithms with sharing for multi-modal function optimization, en Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms and Their Applications, 41-49.
- 9 HOLLAND, J. (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.
- 10 KOZA, J.R. (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press.
- 11 MICHALEWICZ, Z. (1992) Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin Heidelberg.
- 12 CARLOS ARTEMIO CCOELLO C. "Notas del curso de Computación Evolutiva", LANIA, 1999 México.

REFERENCIA INTERNET

<http://www.etsimo.uniovi.es/ftp/pub/EC/FAQ/www/top.htm>

Guía del Autoestopista de la Computación Evolutiva ("The Hitch-Hiker's Guide to Evolutionary Computation") Heitkötter, Jörg and Beasley, David, eds. FAQ de comp.ai.genetic con definiciones y clasificaciones de los términos más comunes en Computación Evolutiva

<http://research.de.uu.net:8080/encore>

<ftp://zeus.etsimo.uniovi.es/pub/EC/Welcome.html>

Encore, the Electronic Appendix to "The Hitchhiker's Guide to Evolutionary Computation" by Jörg Heitkötter

<http://alife.santafe.edu/~joke/zooland/>

<http://www.krl.caltech.edu/~brown/alife/zooland/>

<http://research.de.uu.net:8080/zooland/>

<http://www.dl.ac.uk/CCP/CCP14/ccp/web-mirrors/alife/zooland/zooland/>

<ftp://ftp.de.uu.net/pub/research/ci/Alife/zooland/>

Zooland, "The Artificial Life Resource" by Jörg Heitkötter

ANEXO A

RESULTADOS DE LAS OPERACIONES DE CLONACION Y MUTACIÓN PARA LA POBLACIÓN NUMERO TRES SELECCIONADA, CON LA APLICACIÓN DE INGENIERIA GENETICA – MUTACIÓN MÚLTIPLE

PARÁMETROS

TAMAÑO DE LA POBLACIÓN = 50

NUMERO DE GENERACIONES = 150

PRIMERA POBLACIÓN GENERADA ALEATORIAMENTE

1101001110100101000100	0	0	0
1001000110110101000101	0	0	0
0001110111000011011001	0	0	0
0101000110000001110010	0	0	0
1100100000011011100100	0	0	0
1011010111110001100010	0	0	0
0101011011100000111110	0	0	0
0100011001001100001010	0	0	0
0100001010101101101001	0	0	0
1000011000111110000110	0	0	0
0010111100110010100010	0	0	0
0001000111101010001000	0	0	0
0101110101101011011110	0	0	0
1000011011111110000111	0	0	0
0101100101011011010100	0	0	0
0100000000110101010010	0	0	0
1001100100110110010010	0	0	0
0010110000000100111101	0	0	0
0111000000001110001010	0	0	0
1110101111010000000111	0	0	0
1001101000110001110111	0	0	0
0011001100101111001101	0	0	0
1011111100101010010101	0	0	0
1011111110011100010000	0	0	0
0111010100111000101100	0	0	0
1100110111100110011010	0	0	0
0000111011010101000101	0	0	0
1101011010000100010100	0	0	0
0001001000101111100101	0	0	0
1100110010100000010100	0	0	0
0111010100001111100101	0	0	0
1111101100010010010010	0	0	0
0000110111001110100101	0	0	0
0011111100101011011110	0	0	0
0101111100000101001110	0	0	0
1001110101101001001000	0	0	0
0100101011101110010011	0	0	0
1111010011010100100001	0	0	0
1001110011101101011001	0	0	0
1110111010000010100100	0	0	0
0110111101101010000001	0	0	0
0110001110100010100101	0	0	0

1000000011001000111100	0	0	0
0000100010100101101111	0	0	0
0001100000100001011010	0	0	0
0100010101110011110001	0	0	0
0011010100010001000010	0	0	0
0101110100111111111111	0	0	0
1100011010010001011011	0	0	0
0111101100110011100101	0	0	0

VALORES FITNES APLICADO A LA POBLACIÓN INICIAL

1101001110100101000100	3467588	1.480213	2.862022
1001000110110101000101	2387269	0.707508	1.834657
0001110111000011011001	487641	-0.651212	2.65074
0101000110000001110010	1335410	-0.04484	2.044252
1100100000011011100100	3278564	1.345012	0.671466
1011010111110001100010	2980962	1.132151	1.041233
0101011011100000111110	1423422	0.018111	2.009758
0100011001001100001010	1151754	-0.176201	1.880196
0100001010101101101001	1092457	-0.218614	2.120676
1000011000111110000110	2199430	0.573155	1.571924
0010111100110010100010	773282	-0.446905	2.444795
0001000111101010001000	293512	-0.790064	1.757364
0101110101101011011110	1530590	0.094764	2.015519
1000011011111110000111	2211719	0.581945	1.687331
0101100101011011010100	1464020	0.047149	2.04696
0100000000110101010010	1051986	-0.247561	2.246834
1001100100110110010010	2510226	0.795454	1.886781
0010110000000100111101	721213	-0.484148	2.231263
0111000000001110001010	1835914	0.313148	1.874296
1110101111010000000111	3863559	1.763433	0.391286
1001101000110001110111	2526327	0.80697	2.175301
0011001100101111001101	838605	-0.400183	2.002298
1011111100101010010101	3132053	1.240219	3.182132
1011111110011100010000	3139344	1.245434	3.232645
0111010100111000101100	1920556	0.373689	1.725106
1100110111100110011010	3373466	1.412892	2.556703
0000111011010101000101	243013	-0.826183	2.605512
1101011010000100010100	3514644	1.51387	1.36103
0001001000101111100101	297957	-0.786884	1.684872
1100110010100000010100	3352596	1.397964	1.910647
0111010100001111100101	1917925	0.371807	1.712087
1111101100010010010010	4113554	1.942244	0.115131
0000110111001110100101	226213	-0.8382	2.78126
0011111100101011011110	1034974	-0.259729	2.247692
0101111100000101001110	1556814	0.11352	1.953218
1001110101101001001000	2579016	0.844656	2.832783
0100101011101110010011	1227667	-0.121904	1.922579
1111010011010100100001	4011297	1.869104	3.542463
1001110011101101011001	2571097	0.838992	2.789323
1110111010000010100100	3907748	1.79504	1.721413
0110111101101010000001	1825409	0.305635	1.94618
0110001110100010100101	1632421	0.167599	1.857371
1000000011001000111100	2110012	0.509199	1.854891
0000100010100101101111	141679	-0.898663	2.037728
0001100000100001011010	395354	-0.717221	1.630633

1111010011010100100001	4011297	1.869104	3.542463
1111010011010100100001	4011297	1.869104	3.542463
1111010011010100100001	4011297	1.869104	3.542463

salida despues de mutacion de la generacion 0

1001011010011101101000	2467688	0.765028	1.318665
1100001001010110101110	3184046	1.277408	2.832423
1100111001001001000001	3379777	1.417405	2.737
0110110011000100001111	1782031	0.274608	2.196553
1010010001110010010111	2694295	0.92711	1.302442
111110110111100110010	4157234	1.973486	0.539772
0010110010110011110100	732404	-0.476144	2.324372
1101100111110100000100	3570948	1.554142	0.458995
0101100101110111100000	1465824	0.048439	2.048381
1010100011001100111100	2765628	0.978132	1.37964
0010110111011110010011	751507	-0.46248	2.427386
1100110101110000100011	3365923	1.407496	2.328415
1110000011010110100101	3683749	1.634823	3.452494
0000100011011101110011	145267	-0.896097	2.109603
0111100000111010000111	1969799	0.40891	2.112976
0101111111011001110101	1570421	0.123253	1.917759
1100010001010000000110	3216390	1.300542	1.977858
1110010100111101000001	3755841	1.686387	2.699402
0011000101011100000000	808704	-0.42157	2.264303
0010111111111111100001	786401	-0.437522	2.404334
0001111001010010000011	496771	-0.644682	2.635704
1001110010100000001101	2566157	0.835459	2.749791
1010011110011101101001	2746217	0.964248	1.130747
1110111000010000100001	3900449	1.789819	1.437255
1010010111010000000000	2716672	0.943116	1.078856
0101010111010111011100	1406428	0.005956	2.001108
1101000111001111111110	3437566	1.458739	3.404103
0111011010011100101000	1943336	0.389983	1.879285
000111110110111000101	519621	-0.628338	2.488369
0011111011110100000100	1031428	-0.262265	2.243036
001011111100001000101	915525	-0.345165	1.658808
1011001100011000011101	2934301	1.098776	2.042237
0111010000000110100011	1900963	0.359675	1.656812
1111000111111111010010	3964882	1.835905	3.658843
0111000010011110100011	1845155	0.319758	1.814023
1001100110001111001010	2515914	0.799522	1.988003
0011001010000110000100	827780	-0.407925	2.100522
1101111110000011101001	3662057	1.619308	2.923098
1111000010110110001010	3943818	1.820839	3.108719
1110000111010110101101	3700141	1.646548	3.636873
0001011111000000111101	389181	-0.721636	1.5464
0000000010100001110101	10357	-0.992592	1.771077
0010110010111110111010	733114	-0.475636	2.32954
0011111111100110000101	1046917	-0.251186	2.251012
1111000001101011101000	3939048	1.817427	2.946064
1000111110101010000000	2353792	0.683564	2.33749
1110111110000000011100	3923996	1.806661	2.375329
1110010101011101101111	3757935	1.687885	2.627007
0101011010110000101011	1420331	0.0159	2.007616
1011110010110100100010	3091746	1.21139	2.424263
mejor individuo en generacion 1			
1111000111111111010010	3964882	1.835905	3.658843


```
salida despues de clonacion de la generacion 1
1111000111111111010010  3964882  1.835905  3.658843
1111000111111111010010  3964882  1.835905  3.658843
.....
.....

1111000111111111010010  3964882  1.835905  3.658843
1111000111111111010010  3964882  1.835905  3.658843
```

..... **SIGUE EL PROCESO DE CLONACION Y MUTACIÓN.**

ANEXO B

COFIGO DEL PROGRAMA DE IMPLEMENTACION DEL ALGORITMO GENETICO CON INCORPORACIÓN DE LOS OPERADORES DE CLONACION Y MUTACIÓN

```
// aplicacion de los algoritmos geneticos en optimizacion funcional
// aplicando clonacion y mutacion para una sola variable
// autor: Ing. Nayer Tumi Figueroa
// Fecha: 28 de Mayo del 2001
/*
  se ha sacrificado la eficiencia del codigo a favor de la claridad
  al ser este una demostracion de la simplicidad de la aplicacion del
  algoritmos geneticos para efectos netamente educativos por lo que
  se ha escrito este codigo pensando en que pueda ser comprendido por
  una mayor cantidad de estudiantes, el codigo esta en ANSI C

  se autoriza el uso de este software a todas las personas que lo
  mencionen como referenencia cualquier modificacion y alteracion es
  bajo responsabilidad de los que modifiquen el codigo.

  e-mail ntumi@latinmail.com

*/
#include <stdio.h>
#include <time.h>
#include <math.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>

#define MAXPOB 100
#define TAMPOB 50 // tamaño de la poblacion
#define MAXGENERA 150 // numero maximo de generaciones
#define PXOVER 0.80 // probabilidad de cruzamiento
#define PMUTACION 0.15 // probabilidad de mutacion
#define TAMCROMO 22 // tamaño del cromosoma

// definicion del genotipo para cada individuo de la poblacion
struct genotipo
{
  char cromosoma[TAMCROMO+1];
  unsigned long genA;
  long double x;
  long double fitness;
};

// arrays para almacenar la poblacion actual y la nueva
genotipo poblacion[TAMPOB];

// parametros globales para reportes de cada generacion
long double Gmfit, Gsdfit, GmejorFit, Gmejorx, GmejorA;

genotipo mejorIndividuoP, peorIndividuoP, mejorIndividuoTotal;
char Gpc[24];
```

```

ofstream Gxsalida, Nalcatorio;

// variable global a efectos de utilizarlo como control de generaciones
int ngeneracion=0;

// prototipos de los procedimientos utilizados
void inicio(void); // para inicializar la poblacion aleatoriamente
void select(void); // selecciona los nuevos individuos
void clonacion(void); // realiza el cruzamiento de los individuos
void mutacion(void); // se muta la poblacion
void salida(void); // para ver la poblacion en cada proceso
void fitness(void); // evalua los valores de aptitud
unsigned long calculoA(int); // calcula el valor real de la cadena binaria
void reporte(void);

// inicializa los cromosomas de la poblacion en forma aleatoria
void inicio(void)
{
randomize(); // inicializa la semilla aleatoria
for(int k=0;k<TAMPOB;k++)
for(int c=0;c<TAMCROMO;c++)
{
for(int c=0;c<TAMCROMO;c++) // recorre el vector binario
{
// para inicializarlo uno por uno
float cr;
cr = (double)rand()/RAND_MAX;
if(cr<= 0.5)
poblacion[k].cromosoma[c]='0';
else
poblacion[k].cromosoma[c]='1';
}
// inicializa a cero los demas valores del genotipo
poblacion[k].cromosoma[TAMCROMO]=0; //fin de cromosoma nulo
poblacion[k].genA=0;
poblacion[k].x =0;
poblacion[k].fitness=0;
}
}

long double funcionx(double xx)
{
long double fx;
fx = xx * sin ( 10*M_PI*xx ) + 2;
return fx;
}

// evalua los valores fitness para cada cromosoma
// el valor de cada fitness esta definido por f(x)
void fitness(void)
{
unsigned long a;
for(int c=0;c<TAMPOB;c++)
{
a = calculoA(c); // calcula el valor decimal de la cadena binaria
poblacion[c].genA =a;
poblacion[c].x = -1 + ( a * ((long double)3/4194303));
poblacion[c].fitness = funcionx(poblacion[c].x); // valor de la funcion para x
}
}

} // fin de fitness

```

```

// operacion de seleccion del mejor individuo
void selectBest(void)
{
    long double fit,x;
    int c;

    // mejorIndividuoP=poblacion[0];

    for( c = 0;c < TAMPOB ; c++)
    {
        if( poblacion[c].fitness > mejorIndividuoP.fitness )
        {
            mejorIndividuoP=poblacion[c];
            fit=mejorIndividuoP.fitness;
            x=mejorIndividuoP.x;
            Naleatorio<<endl<<"indv mejor es = "<<setw(4)<<c<<" fit = "<<setw(10)<<fit<<" x = "<<setw(10)<<x;
            if(mejorIndividuoP.fitness > mejorIndividuoTotal.fitness) // actualizando individuo
                mejorIndividuoTotal=mejorIndividuoP;
        }
    }
}

void clonacion(void)
{
    double x;
    int contador=0, uno;

    for(int c=0;c<TAMPOB ; c++)
    {
        poblacion[c]=mejorIndividuoTotal;
    }
}

void mutacion(void)
{
    unsigned p2,pn;
    pn=rand()%TAMCROMO;

    for(int c =0; c<TAMPOB; c++)
    {
        for(int k=0;k<pn;k++)
        {
            p2 = rand()%TAMCROMO;
            if(poblacion[c].cromosoma[p2]=='1')
                poblacion[c].cromosoma[p2]='0';
            else
                poblacion[c].cromosoma[p2]='1';
            Naleatorio<<endl<<"mutacion del individuo i = "<<setw(4)<<c<<" en el punto p2= "<<setw(4)<<p2;
        }
    }
}

void salida(void)
{
    char pc[24];
    float p,pf,pfr,pfc,px;
    unsigned long pa;
    printf("ngeneracion nr %2d",ngeneracion);
    for(int c=0;c<TAMPOB;c++)
    {
        strcpy(pc,poblacion[c].cromosoma);
    }
}

```

```

    pa = poblacion[c].genA;
    px = poblacion[c].x;
    pf = poblacion[c].fitness;

    cout<<endl<<pc<<setw(10)<<pa<<setw(10)<<px<<setw(10)<<pf;
    Gxsalida<<endl<<pc<<setw(10)<<pa<<setw(10)<<px<<setw(10)<<pf;
}
}

// retorna el valor decimal de la cadena binaria
unsigned long calculoA(int ka)
{
    int bita;
    unsigned long avalor=0,agen;
    for(int a=21;a>=0;a--)
    {
        bita = ((poblacion[ka].cromosoma[a]-48);
        avalor = avalor + bita*pow(2,21-a);
    }
    agen = avalor;
    return agen;
}

// calcula los siguientes parametros para cada generacion.
// el mejor individuo
// el promedio, desviacion estandar de los fitness

void reporte(void)
{
    long double fit,cfit, sfit=0,scfit=0;
    char pc[24];
    //variables globales double Gmfit,Gsdfit;

    for(int c=0;c < TAMPOB;c++)
    {
        fit = poblacion[c].fitness;
        sfit = sfit+fit; // suma de fitness
        cfit= fit*fit;
        scfit = scfit+cfit; // suma de cuadrados de los fitness
    }

    Gmfit = sfit/TAMPOB;

    // calculando la desviacion estandar de los fitness
    double ss=0,s1;
    for(c=0;c<=TAMPOB;c++)
    {
        ss = ss + pow( (poblacion[c].fitness-Gmfit),2);
    }
    s1 =(double)ss/(TAMPOB-1);
    Gsdfit =sqrt(s1);
    strcpy(pc,mejorIndividuoP.cromosoma);
    strcpy(Gpc,mejorIndividuoP.cromosoma);
    GmejorFit = mejorIndividuoP.fitness;
    Gmejorx = mejorIndividuoP.x;
    GmejorA = mejorIndividuoP.genA;
    cout<<endl<<" "<<pc<<setw(12)<<GmejorA<<setw(12)<<Gmejorx<<setw(12)<<GmejorFit;
    cout<<setw(12)<<Gmfit<<setw(15)<<Gsdfit<<endl;

} // fin de reporte

```

```

// modulo principal del programa
void main(void )
{
    char pc[TAMCROMO+1];
    long double p,pf,pfr,pfc,px,py;
    long pa,pb;
    // escribiendo el reporte en un archivo
    ofstream Gsalida;
    Gsalida.open("salida2.txt");
    if(!Gsalida)
    {
        cerr<<"No se puede abrir el archivo de salida";
    }

    Gxsalida.open("xsalida2.txt");
    if(!Gxsalida)
    {
        cerr<<"No se puede abrir el archivo de salida";
    }

    Naleatorio.open("Nazar2.txt");
    if(!Naleatorio)
    {
        cerr<<"No se puede abrir el archivo de salida";
    }

    inicio();
    salida(); getch();
    fitness();
    mejorIndividuoP=poblacion[0];
    mejorIndividuoTotal=poblacion[0];
    salida(); reporte();

    for( ngeneracion = 0 ; ngeneracion < MAXGENERA ; ngeneracion++ )
    {
        selectBest();
        Gxsalida<<endl<<"mejor individuo en generacion " <<ngeneracion;
        strcpy(pc,mejorIndividuoP.cromosoma);
        pa = mejorIndividuoP.genA;
        px = mejorIndividuoP.x;
        pf = mejorIndividuoP.fitness;
        Gxsalida<<endl<<pc<<setw(10)<<pa<<setw(12)<<px<<setw(12)<<pf;

        Gxsalida<<endl<<endl<<"salida despues de clonacion de la generacion " <<ngeneracion;
        clonacion();
        salida();

        Gxsalida<<endl<<endl<<"salida despues de mutacion de la generacion " <<ngeneracion;
        mutacion();
        fitness();
        salida();
        cout<<endl<<setw(4)<<ngeneracion;
        reporte();
        Gsalida<<" " <<Gpc<<setw(12)<<GmejorA<<setw(12)<<Gmejorx<<setw(12)<<GmejorFit;
        Gsalida<<setw(12)<<Gmfit<<setw(15)<<Gsdfit<<endl;
    }
    Gsalida.close();
    Gxsalida.close();
    Naleatorio.close();
}

```