

**DESARROLLO DE UNA HERRAMIENTA ACADÉMICA PARA EL
APRENDIZAJE DE ALGORITMOS GENÉTICOS EN EL ENTORNO DE
MATLAB, APLICADA A TELECOMUNICACIONES**

MÓNICA ROCÍO REYES ABRIL

CARLOS ANDRÉS PARDO RODRÍGUEZ

**FUNDACIÓN UNIVERSITARIA SAN MARTÍN
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ
2009 II**

**DESARROLLO DE UNA HERRAMIENTA ACADÉMICA PARA EL
APRENDIZAJE DE ALGORITMOS GENÉTICOS EN EL ENTORNO DE
MATLAB, APLICADA A TELECOMUNICACIONES**

**MÓNICA ROCÍO REYES
051079
MONIKA-190@HOTMAIL.COM**

**CARLOS ANDRÉS PARDO
051068
CARLOSAN_1@HOTMAIL.COM**

MONOGRAFÍA DE GRADO

**ASESOR TÉCNICO
ING. IVÁN LADINO VEGA
ALGORITMOS GENÉTICOS**

**FUNDACIÓN UNIVERSITARIA SAN MARTÍN
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ
2009 II**

Nota de aceptación

Ing. IVÁN LADINO VEGA
Asesor

Ing. DIEGO DÍAZ
Jurado 1

Ing. FREUD ROMERO
Jurado 2

Bogotá, 14 de Diciembre de 2009

Es preferible morir de agotamiento, antes que de aburrimiento.

Dedicamos este proyecto y toda nuestra carrera universitaria a Dios, quien se ha encargado día a día de brindarnos las herramientas necesarias para solucionar todos los inconvenientes que se presenten. Les agradecemos a nuestros padres quienes nos brindan su apoyo incondicional, gracias a ellos somos lo que somos hoy en día, fueron ellos los que nos brindaron el cariño y calor humano que nos hace hoy en día ser unos seres de sociedad. Ellos quienes han velado por nuestra salud, nuestra alimentación, nuestra educación y lo más importante nuestros sueños. A ellos les debemos todo, nuestra alegría, muchos regaños, horas de consejos de los cuales aseguramos han sido tomados muy en cuenta. También agradecemos a todas aquellas personas, amigos y compañeros, quienes nos brindaron su apoyo, tiempo e información para poder lograr nuestros sueños.

AGRADECIMIENTOS

Nuestro agradecimiento a la Fundación Universitaria San Martín.
Institución que nos acogió como hijos, para poder lograr nuestros sueños y de la cual siempre hemos recibido apoyo.

Al Ing. Iván Ladino,
Por su dirección, paciencia, colaboración y apoyo constante que nos brindo a lo largo del desarrollo de nuestra carrera universitaria y en este proyecto.

Al Ing. Hans López,
Quien nos sirvió como modelo de profesional a seguir a lo largo de nuestra carrera universitaria.

Al Ing. Diego Huertas,
Quien nos incito a ser buenos estudiantes, para poder ser mejores profesionales.

CONTENIDO

| | pág. |
|--|------|
| INTRODUCCIÓN | 18 |
| 1. PROBLEMA | 20 |
| 2. JUSTIFICACIÓN | 23 |
| 3. OBJETIVOS | 24 |
| 3.1 OBJETIVO GENERAL | 24 |
| 3.2 OBJETIVOS ESPECÍFICOS | 24 |
| 4. MARCO REFERENCIAL | 25 |
| 4.1 ANTECEDENTES | 25 |
| 4.2 MARCO CONCEPTUAL | 26 |
| 4.2.1 PROGRAMAR | 26 |
| 4.2.2 LENGUAJES DE PROGRAMACIÓN | 26 |
| 4.2.3 ALGORITMO | 27 |
| 4.2.4 OPTIMIZACIÓN DE ALGORITMOS | 27 |
| 4.2.5 INTELIGENCIA EVOLUTIVA | 27 |
| 4.2.6 EVOLUCIÓN BIOLÓGICA Y ARTIFICIAL | 28 |
| 4.2.7 GENÉTICA | 29 |
| 4.2.8 EL ADN | 30 |
| 4.3 MARCO TEÓRICO | 30 |

| | | |
|-------|---|----|
| 4.3.1 | INTRODUCCIÓN | 30 |
| 4.3.2 | COMPUTACIÓN EVOLUTIVA | 31 |
| 4.3.3 | ALGORITMOS EVOLUTIVOS | 33 |
| 4.3.4 | ALGORITMOS GENÉTICOS | 33 |
| 4.3.5 | ARREGLOS LINEALES DE ANTENAS | 41 |
| 4.3.6 | SERIE TRIGONOMÉTRICA DE FOURIER | 44 |
| 4.4 | ESTADO DEL ARTE | 49 |
| 4.4.1 | EXPRESIÓN GÉNICA DE PROGRAMACIÓN (GEP) | 49 |
| 4.4.2 | SISTEMAS DE PREVISIÓN DE APRENDIZAJE | 49 |
| 4.5 | LIMITACIONES Y ALCANCES | 50 |
| 5. | DISEÑO METODOLÓGICO | 52 |
| 6. | DESARROLLO | 53 |
| 6.1 | DESARROLLO DEL OBJETIVO NÚMERO UNO | 53 |
| 6.2 | DESARROLLO DEL OBJETIVO NÚMERO DOS | 55 |
| 6.2.1 | CARACTERÍSTICAS DE LOS ALGORITMOS GENÉTICOS | 55 |
| 6.2.2 | VENTAJAS DE LOS ALGORITMOS GENÉTICOS | 56 |
| 6.2.3 | DESVENTAJAS DE LOS ALGORITMOS GENÉTICOS | 59 |
| 6.3 | DESARROLLO DEL OBJETIVO NÚMERO TRES | 61 |
| 6.3.1 | ARQUITECTURA GENERAL | 61 |
| 6.3.2 | ALOJAMIENTO WEB O WEBHOSTING | 62 |

| | | |
|-------|---------------------------------------|----|
| 6.3.3 | NOMBRE DE DOMINIO | 63 |
| 6.3.4 | CONTENIDO INTERNO | 63 |
| 6.4 | DESARROLLO DEL OBJETIVO NÚMERO CUATRO | 66 |
| 6.4.1 | GENERAR LA POBLACIÓN INICIAL | 68 |
| 6.4.2 | OPERADOR DE CRUCE | 69 |
| 6.4.3 | OPERADOR DE MUTACIÓN | 71 |
| 6.4.4 | SELECCIÓN | 72 |
| 6.5 | DESARROLLO DEL OBJETIVO NÚMERO CINCO | 72 |
| 6.5.1 | GENERAR POBLACIÓN INICIAL | 73 |
| 6.5.2 | OPERADOR CRUCE | 73 |
| 6.5.3 | OPERADOR DE MUTACIÓN | 75 |
| 6.5.4 | SELECCIÓN | 75 |
| 7. | PRUEBAS Y RESULTADOS | 77 |
| 7.1 | OBJETIVO NÚMERO UNO | 77 |
| 7.1.1 | POBLACIÓN INICIAL | 77 |
| 7.1.2 | OPERADOR DE CRUCE | 77 |
| 7.1.3 | OPERADOR DE MUTACIÓN | 78 |
| 7.1.4 | SELECCIÓN | 79 |
| 7.2 | OBJETIVO NÚMERO DOS | 92 |
| 7.2.1 | ARREGLO DE ANTENAS | 92 |

| | | |
|-------|----------------------|-----|
| 7.2.2 | POBLACIÓN INICIAL | 92 |
| 7.2.3 | OPERADOR DE CRUCE | 93 |
| 7.2.4 | OPERADOR DE MUTACIÓN | 94 |
| 7.2.5 | SELECCIÓN | 95 |
| 8. | CONCLUSIONES | 109 |
| 9. | RECOMENDACIONES | 110 |
| | GLOSARIO | 111 |
| | BIBLIOGRAFÍA | 113 |

LISTA DE TABLAS

| | pág. |
|--|------|
| Tabla 1. Paradigmas de computación evolutiva | 32 |
| Tabla 2. Comparación entre la programación habitual y los algoritmos genéticos | 53 |
| Tabla 3. Características del plan contratado | 63 |
| Tabla 4. Selección de la señales que van a realizar el operador de cruce | 70 |
| Tabla 5. Proceso de mutación en una señal | 72 |
| Tabla 6. Criterio de cruce seleccionado | 74 |
| Tabla 7. Proceso de mutación en un patrón de radiación | 75 |
| Tabla 8. Ejecución número 1 del algoritmo genético para encontrar la señal | 80 |
| Tabla 9. Ejecución número 2 del algoritmo genético para encontrar la señal | 81 |
| Tabla 10. Ejecución número 3 del algoritmo genético para encontrar la señal | 81 |
| Tabla 11. Ejecución número 4 del algoritmo genético para encontrar la señal | 81 |
| Tabla 12. Ejecución número 5 del algoritmo genético para encontrar la señal | 82 |
| Tabla 13. Ejecución número 6 del algoritmo genético para encontrar la señal | 82 |
| Tabla 14. Ejecución número 7 del algoritmo genético para encontrar la señal | 82 |
| Tabla 15. Ejecución número 8 del algoritmo genético para encontrar la señal | 83 |
| Tabla 16. Ejecución número 9 del algoritmo genético para encontrar la señal | 83 |
| Tabla 17. Ejecución número 10 del algoritmo genético para encontrar la señal | 83 |
| Tabla 18. Ejecución número 11 del algoritmo genético para encontrar la señal | 84 |
| Tabla 19. Ejecución número 12 del algoritmo genético para encontrar la señal | 84 |
| Tabla 20. Ejecución número 13 del algoritmo genético para encontrar la señal | 84 |
| Tabla 21. Ejecución número 14 del algoritmo genético para encontrar la señal | 85 |
| Tabla 22. Ejecución número 15 del algoritmo genético para encontrar la señal | 85 |
| Tabla 23. Ejecución número 1 para un segundo criterio de selección | 86 |
| Tabla 24. Ejecución número 2 para un segundo criterio de selección | 87 |
| Tabla 25. Ejecución número 3 para un segundo criterio de selección | 87 |
| Tabla 26. Ejecución número 4 para un segundo criterio de selección | 87 |
| Tabla 27. Ejecución número 5 para un segundo criterio de selección | 88 |
| Tabla 28. Ejecución número 6 para un segundo criterio de selección | 88 |

| | |
|--|-----|
| Tabla 29. Ejecución número 7 para un segundo criterio de selección | 88 |
| Tabla 30. Ejecución número 8 para un segundo criterio de selección | 89 |
| Tabla 31. Ejecución número 9 para un segundo criterio de selección | 89 |
| Tabla 32. Ejecución número 10 para un segundo criterio de selección | 89 |
| Tabla 33. Ejecución número 11 para un segundo criterio de selección | 90 |
| Tabla 34. Ejecución número 12 para un segundo criterio de selección | 90 |
| Tabla 35. Ejecución número 13 para un segundo criterio de selección | 90 |
| Tabla 36. Ejecución número 14 para un segundo criterio de selección | 91 |
| Tabla 37. Ejecución número 15 para un segundo criterio de selección | 91 |
| Tabla 38. Ejecución número 1 del algoritmo genético para encontrar la antena | 97 |
| Tabla 39. Ejecución número 2 del algoritmo genético para encontrar la antena | 97 |
| Tabla 40. Ejecución número 3 del algoritmo genético para encontrar la antena | 97 |
| Tabla 41. Ejecución número 4 del algoritmo genético para encontrar la antena | 98 |
| Tabla 42. Ejecución número 5 del algoritmo genético para encontrar la antena | 98 |
| Tabla 43. Ejecución número 6 del algoritmo genético para encontrar la antena | 98 |
| Tabla 44. Ejecución número 7 del algoritmo genético para encontrar la antena | 99 |
| Tabla 45. Ejecución número 8 del algoritmo genético para encontrar la antena | 99 |
| Tabla 46. Ejecución número 9 del algoritmo genético para encontrar la antena | 99 |
| Tabla 47. Ejecución número 10 del algoritmo para encontrar la antena | 100 |
| Tabla 48. Ejecución número 11 del algoritmo para encontrar la antena | 100 |
| Tabla 49. Ejecución número 12 del algoritmo para encontrar la antena | 100 |
| Tabla 50. Ejecución número 13 del algoritmo para encontrar la antena | 101 |
| Tabla 51. Ejecución número 14 del algoritmo para encontrar la antena | 101 |
| Tabla 52. Ejecución número 15 del algoritmo para encontrar la antena | 101 |
| Tabla 53. Ejecución número 1 para un segundo criterio de selección | 103 |
| Tabla 54. Ejecución número 2 para un segundo criterio de selección | 103 |
| Tabla 55. Ejecución número 3 para un segundo criterio de selección | 104 |
| Tabla 56. Ejecución número 4 para un segundo criterio de selección | 104 |
| Tabla 57. Ejecución número 5 para un segundo criterio de selección | 104 |
| Tabla 58. Ejecución número 6 para un segundo criterio de selección | 105 |
| Tabla 59. Ejecución número 7 para un segundo criterio de selección | 105 |

| | |
|---|-----|
| Tabla 60. Ejecución número 8 para un segundo criterio de selección | 105 |
| Tabla 61. Ejecución número 9 para un segundo criterio de selección | 106 |
| Tabla 62. Ejecución número 10 para un segundo criterio de selección | 106 |
| Tabla 63. Ejecución número 11 para un segundo criterio de selección | 106 |
| Tabla 64. Ejecución número 12 para un segundo criterio de selección | 107 |
| Tabla 65. Ejecución número 13 para un segundo criterio de selección | 107 |
| Tabla 66. Ejecución número 14 para un segundo criterio de selección | 107 |
| Tabla 67. Ejecución número 15 para un segundo criterio de selección | 108 |

LISTA DE FIGURAS

| | pág. |
|---|------|
| Figura 1. Estructura del ADN | 30 |
| Figura 2. Taxonomía de técnicas de búsqueda | 32 |
| Figura 3. Estructura de algoritmos evolutivos | 33 |
| Figura 4. Ciclo realizado por un algoritmo genético | 34 |
| Figura 5. Etapas del cruce | 37 |
| Figura 6. Mutación de una cadena de bits | 37 |
| Figura 7. Cada cromosoma codifica una posible solución al problema | 38 |
| Figura 8. Clasificación de algoritmos genéticos en paralelo | 40 |
| Figura 9. Disposición física de un arreglo lineal de antenas | 42 |
| Figura 10. Dipolo y su patrón de radiación de campo eléctrico | 43 |
| Figura 11. Coeficientes de una serie relacionados mediante un triángulo | 45 |
| Figura 12. Anatomía del algoritmo genético representado en las aplicaciones | 67 |
| Figura 13. Proceso para generar la población inicial | 68 |
| Figura 14. Gráfica del criterio de cruce utilizado para el generador de señales | 70 |
| Figura 15. Gráfica del criterio de cruce seleccionado para las antenas | 74 |
| Figura 16. Representación de los módulos y las fases de dos señales | 77 |
| Figura 17. Ejemplo del cruce entre dos señales | 78 |
| Figura 18. Ejemplo del cruce entre la señal padre s_4 y la señal padre s_6 | 78 |
| Figura 19. Gráfica de la mutación realizada a una señal | 79 |
| Figura 20. Mutación realizada a la señal 17 | 79 |
| Figura 21. Señales seleccionadas que cumplen el criterio de selección definido | 80 |
| Figura 22. Relación entre el número de ejecuciones y de iteraciones | 86 |
| Figura 23. Gráfica del Número de ejecuciones vs. Número de iteraciones | 92 |
| Figura 24. Patrones de radiación de dos antenas padres | 93 |
| Figura 25. Representación de dos antenas padres y su respectivo cruce | 93 |
| Figura 26. Ejemplo de cruce entre dos antenas | 94 |
| Figura 27. Proceso que atraviesa una antena en un algoritmo genético | 95 |
| Figura 28. Gráfica del proceso que hace una antena en un algoritmo genético | 95 |

| | |
|---|-----|
| Figura 29. Patrones de radiación de las antenas seleccionadas | 96 |
| Figura 30. Número de iteraciones vs. Número de ejecuciones | 102 |
| Figura 31. Número de iteraciones vs. Número de ejecuciones | 108 |

LISTA DE FÓRMULAS

| | pág. |
|---|------|
| Fórmula 1. Campo eléctrico | 43 |
| Fórmula 2. Factor de arreglo normalizado | 43 |
| Fórmula 3. Factor de arreglo normalizado | 44 |
| Fórmula 4. Directividad | 44 |
| Fórmula 5. Estructura general de la serie trigonométrica de Fourier | 44 |
| Fórmula 6. Representación de la serie trigonométrica de Fourier | 45 |
| Fórmula 7. Serie de Fourier | 45 |
| Fórmula 8. Representación de los coeficientes de un triángulo rectángulo | 46 |
| Fórmula 9. Coeficientes de Fourier simplificados | 46 |
| Fórmula 10. Representación de la serie de Fourier con coeficientes | 46 |
| Fórmula 11. Conjunto de funciones ortogonales | 47 |
| Fórmula 12. Representación general de los coeficientes de la serie de Fourier | 48 |
| Fórmula 13. Primer coeficiente de Fourier | 48 |
| Fórmula 14. Segundo coeficiente de Fourier | 48 |
| Fórmula 15. Función de integración | 48 |
| Fórmula 16. Magnitud de la serie de Fourier | 69 |
| Fórmula 17. Fase de la serie de Fourier | 69 |
| Fórmula 18. Directividad | 73 |

LISTA DE ANEXOS

| | pág. |
|---|------|
| Anexo A. Código en Matlab de la aplicación del generador de señales | 114 |
| Anexo B. Código en Matlab de la aplicación de la antena tipo panel | 114 |

RESUMEN

La programación habitual sólo puede dar una solución en una dirección al mismo tiempo a una situación dada, y si ésta solución resulta no ser la óptima, se debe iniciar de nuevo para encontrar otra solución.

En los sistemas de computación modernos, esto se ve traducido en un problema de efectividad y eficiencia, ya que sólo se tiene un camino para encontrar la solución adecuada al problema al cual se enfrenta. Es por esto que se hace necesaria una herramienta que contribuya a la exploración de soluciones en múltiples direcciones a la vez; así, si el camino elegido resultó no ser el correcto, se puede eliminar y continuar con otro camino; aumentando la probabilidad de éxito. Este es un factor muy importante que no se presenta en la programación habitual.

Al trabajar con algoritmos genéticos, se intenta resolver ésta problemática, ya que es una herramienta poderosa y eficiente, que permite modelar y resolver problemas que necesitan un análisis complejo.

Se muestran los conceptos básicos involucrados en la utilización de algoritmos genéticos, tales como cromosomas, genes, población, mutación, apareamiento, aptitud, entre otros. Además, se profundiza en tema de los operadores que utiliza el algoritmo genético como mutación, cruce y selección.

Estos permitirán comprender más a fondo los elementos que conforman la estructura de un algoritmo generado en el entorno de Matlab.

La herramienta que permite realizar este estudio es la versión académica del software Matlab, que posee un capítulo dedicado a la explicación de Algoritmos Genéticos, modelando así 2 aplicaciones básicas que conllevarán a dejar un arraigo académico para posteriores estudios en la facultad de ingeniería de la Fundación Universitaria San Martín.

INTRODUCCIÓN

Programar es un proceso en el cuál se enumeran una serie de instrucciones detalladas. Estas instrucciones son interpretadas por una maquina mediante un lenguaje de programación específico.

Los métodos de programación convencionales se encargan de brindar una solución única a un problema dado; si en el proceso de uso de la aplicación el problema cambia, el algoritmo deja de ser eficiente. Es decir, los métodos de programación habituales poseen deficiencias en la manera de optimización.

El siguiente documento pretende mostrar como la computación natural ha surgido como respuesta a la necesidad de modelar sistemas complejos y como una alternativa de solución a otros problemas, para los cuales la computación tradicional presenta grandes limitaciones, desde el punto de vista de modelado y de complejidad computacional.

La computación evolutiva es una emulación parcial del proceso de selección natural; ésta no requiere un modelo matemático para atacar a un problema en específico. Los Algoritmos evolutivos mantienen una población de estructuras que evolucionan según las reglas de selección que se establezcan, además utiliza otros operadores como la recombinación y la mutación. Esto permite obtener una mayor probabilidad de éxito frente a los problemas que se puedan plantear.

En La Fundación Universitaria San Martín, en el área de Ingeniería, no existe la cátedra ni el eje temático de Computación Evolutiva, por ende, no se tiene ninguna alternativa metodológica conocida para enfrentar esta situación. Diferente a otras universidades que actualmente incluyen en su plan de estudios éste importante eje temático.

Las circunstancias anteriores pueden ocasionar que la facultad de ingeniería de La Fundación Universitaria San Martín pierda terreno frente a otras facultades de ingeniería de otras universidades a nivel de actualidad y modernidad en su plan de estudios.

De continuar esta problemática, los egresados de ingeniería no tendrán un valor académico agregado en cuanto a su nivel de programación y no podrán enfrentarse a la situación laboral actual, ya que existirán múltiples competidores con conocimientos de nivel superior al promedio, necesario para obtener un punto extra a destacar en su perfil profesional.

Esta situación hace necesaria la implementación de una nueva metodología académica, por parte de la facultad de ingeniería, para la programación de algoritmos que permitan solucionar los diferentes problemas presentados de forma múltiple y eficiente, para conseguir optimizar los resultados obtenidos de manera segura.

Para lograr esto, se utilizará el paradigma de algoritmos genéticos. Una nueva técnica de programación que posee múltiples ventajas sobre la programación habitual. Así el estudiante de ingeniería San Martiniano tendrá un conocimiento básico de que tan poderosa es ésta herramienta y además, podrá utilizar una solución adicional para diseñar algoritmos optimizados genéticamente.

1. PROBLEMA

Programar es un proceso en el cuál se enumeran una serie de instrucciones detalladas. Estas instrucciones son interpretadas por una maquina mediante un lenguaje de programación específico.

Los métodos de programación habituales poseen deficiencias en la manera de optimización, es decir, en la búsqueda de diferentes caminos o situaciones para encontrar una solución a un problema presentado.

Los métodos de programación convencionales se encargan de brindar una solución única a un problema dado; si en el proceso de uso de la aplicación el problema cambia, el algoritmo deja de ser eficiente.

La computación natural ha surgido como respuesta a la necesidad de modelar sistemas complejos y como una alternativa de solución a otros problemas, para los cuales la computación tradicional presenta grandes limitaciones, desde el punto de vista de modelado y de complejidad computacional.

El área de investigación en Computación Natural comprende actividades en el desarrollo de modelos teóricos inspirados en la naturaleza y en la aplicación de dichos modelos para la solución de problemas del mundo real.

Algunos de los sistemas naturales que han servido de inspiración para estas técnicas de computación incluyen:

- *La Teoría de la Evolución*, que constituye la base para los desarrollos de Algoritmos Genéticos, Programación Genética y demás modelos encierra el eje de la de Computación Evolutiva.
- *El Cerebro y El Sistema Nervioso Central*, que han inspirado el trabajo en el área de redes neuronales artificiales.
- *La Computación con ADN*, que consiste en el uso de procesos de biología molecular con herramientas computacionales para la solución de problemas.
- *El Funcionamiento del Sistema Inmunológico de los Vertebrados*, que ha inspirado el área de Sistemas Inmunológicos Artificiales.
- *El Comportamiento Social de Algunas Especies*, también ha dado lugar al desarrollo de modelos computacionales conocidos como modelos de inteligencia colectiva (Swarm Intelligence).

- *Los Sistemas Físicos*, tales como las partículas en un gas, o en un cristal, modelados por la mecánica estadística.

La computación evolutiva es una emulación parcial del proceso de selección natural; ésta no requiere un modelo matemático para atacar a un problema en específico. Los Algoritmos evolutivos mantienen una población de estructuras que evolucionan según las reglas de selección que se establezcan, además utiliza otros operadores como la recombinación y la mutación. La selección enfoca la atención en los individuos de aptitud alta, mientras aprovechan de así la información de aptitud disponible. En la recombinación y mutación perturba aquellos individuos, manteniendo la característica general de exploración de individuos.

En La Fundación Universitaria San Martín, en el área de Ingeniería, no existe la cátedra ni el eje temático de Computación Evolutiva, por ende, no se tiene ninguna alternativa metodológica conocida para enfrentar esta situación. Diferente a otras universidades que actualmente incluyen en su plan de estudios éste importante eje temático.

Las circunstancias anteriores pueden ocasionar que la facultad de ingeniería de La Fundación Universitaria San Martín pierda terreno frente a otras facultades de ingeniería de otras universidades a nivel de actualidad y modernidad en su plan de estudios.

De continuar esta problemática, los egresados de ingeniería no tendrán un valor académico agregado en cuanto a su nivel de programación y no podrán enfrentarse a la situación laboral actual, ya que existirán múltiples competidores con conocimientos de nivel superior al promedio, necesario para obtener un punto extra a destacar en su perfil profesional.

Esta situación hace necesaria la implementación de una nueva metodología académica, por parte de la facultad de ingeniería, para la programación de algoritmos que permitan solucionar los diferentes problemas presentados de forma múltiple y eficiente, para conseguir optimizar los resultados obtenidos de manera segura.

Para lograr esto, se utilizará el paradigma de algoritmos genéticos. Una nueva técnica de programación que posee múltiples ventajas sobre la programación habitual. Así el estudiante de ingeniería San Martiniano tendrá un conocimiento básico de que tan poderosa es ésta herramienta y además, podrá utilizar una solución adicional para diseñar algoritmos optimizados genéticamente.

Se plantea así una serie de interrogantes con el fin de identificar claramente las necesidades que existen para abordar este importante eje temático:

¿Qué ventajas académicas les brinda la utilización de algoritmos genéticos a los estudiantes de la facultad de ingeniería de La Fundación Universitaria San Martín, a lo largo de su proceso de formación profesional?

- ¿Qué es un algoritmo?
- ¿Cuáles son las ventajas y desventajas que ofrece la programación habitual?
- ¿Qué es un algoritmo genético?
- ¿Qué métodos de representación de algoritmos genéticos existen?
- ¿Qué técnicas utilizan los algoritmos genéticos para seleccionar diferentes opciones de resultado?
- ¿Qué otras técnicas de resolución de problemas se pueden encontrar?
- ¿Cuáles son las ventajas que ofrecen los algoritmos genéticos?
- ¿Cuáles son las limitaciones de los algoritmos genéticos?
- ¿Qué aplicaciones se implementan por parte de los algoritmos genéticos?

2. JUSTIFICACIÓN

La programación habitual sólo puede dar una solución en una dirección al mismo tiempo a una situación dada, y si ésta solución resulta no ser la óptima, se debe iniciar de nuevo para encontrar otra solución.

En los sistemas de computación modernos, esto se ve traducido en un problema de efectividad y eficiencia, ya que sólo se tiene un camino para encontrar la solución adecuada al problema al cual se enfrenta. Es por esto que se hace necesaria una herramienta que contribuya a la exploración de soluciones en múltiples direcciones a la vez; así, si el camino elegido resultó no ser el correcto, se puede eliminar y continuar con otro camino; aumentando esa probabilidad de éxito. Este es un factor muy importante que no se presenta en la programación habitual. La computación evolutiva se encarga precisamente de resolver ésta problemática, resultando ser para los programadores una herramienta poderosa y eficiente, que tiene un gran auge en la actualidad.

Es por esto que se hace importante que La Fundación Universitaria San Martín cuente con el eje temático de Computación Evolutiva en su plan de estudios, ya que para los ingenieros, tanto electrónicos como de sistemas, contribuye como una herramienta académica para su formación profesional, logrando así un valor agregado frente a otros profesionales. Cabe destacar que, como ingenieros, es trascendental que nuestro conocimiento vaya avanzando conforme la tecnología lo hace, para mantener el grado de competitividad en el entorno profesional y no quedar obsoletos en un futuro. Además hay que tener en cuenta que ya existen universidades que incluyen ésta cátedra dentro del plan de estudios básico de sus estudiantes.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Implementar una herramienta académica en el entorno de Matlab para el aprendizaje, de la metodología para el desarrollo de algoritmos genéticos para aplicación a las telecomunicaciones.

3.2 OBJETIVOS ESPECÍFICOS

- Realizar un análisis comparativo entre los algoritmos genéticos y la programación habitual.
- Identificar las ventajas y desventajas que ofrece la utilización de Algoritmos genéticos y sus aplicaciones, para el aprovechamiento correcto de la programación avanzada.
- Proporcionar a los lectores un tutorial básico de programación, en el cual se recopile la información necesaria acerca de los algoritmos genéticos, sus métodos de representación y las técnicas de selección utilizadas para la resolución de problemas.
- Implementar un generador de señales con base en algoritmos genéticos en el entorno de Matlab.
- Diseñar una antena tipo panel a partir del uso de algoritmos genéticos.

4. MARCO REFERENCIAL

4.1 ANTECEDENTES

Entre los años 1950 y 1960, varios científicos informáticos estudiaron sistemas evolutivos con la idea de que la evolución podía ser utilizada como una herramienta de optimización para poder solucionar los problemas de ingeniería (Mitchel, 1998).

La idea de estos sistemas de estudio fue tratar de encontrar una solución a un problema dado, utilizando operadores inspirados en la variación genética natural y la selección natural.

A mediados de los 60 aparece un científico llamado Rechenberg quien presentó las "Estrategias de Evolución", una técnica utilizada para optimizar los parámetros de un valor real para diferentes dispositivos tales como los dispositivos aerodinámicos. Esta idea fue desarrollada dos años después por Schwefel. Este campo se ha mantenido como una área activa de investigación, independiente del campo de algoritmos genéticos. Lawrence J. Fogel, en 1966, desarrolló la técnica de "Programación Evolutiva", con el fin de simular el uso de la evolución como un proceso de aprendizaje, con el objetivo de generar inteligencia artificial. Para el desarrollo de estos, Fogel utilizó máquinas de estados finitos y herramientas de predicción evolutiva.

Los "Algoritmos Genéticos" fueron inventados y desarrollados en la universidad de Michigan por John Holland, sus colegas y estudiantes, en los años 60. A diferencia de las "Estrategias de evolución" y la "Programación evolutiva" el objetivo de Holland no era el diseño de algoritmos para resolver problemas específicos, sino para estudiar formalmente el fenómeno de la adaptación producido en la naturaleza y así desarrollar una forma en la cual los mecanismos de adaptación natural pueden ser adaptados a los sistemas informáticos. Holland, en el libro de Adaptación Natural y Sistemas Artificiales, presentó el Algoritmo Genético como una abstracción de la evolución biológica y le dio un marco teórico para la adaptación en los algoritmos genéticos (Sumathi, 2008).

4.2 MARCO CONCEPTUAL

4.2.1 PROGRAMAR

Programar es un proceso en el cuál se enumera una serie de instrucciones detalladas que son interpretadas por una maquina mediante un lenguaje de programación específico (Louden, 2004).

4.2.2 LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación es una notación que cumple con una serie de reglas y condiciones, tanto semánticas como sintácticas, y define claramente las acciones a realizar por una máquina.

Existen diversos lenguajes para programar en un computador. El más sencillo es el lenguaje de máquina, que consiste en una serie de instrucciones detalladas las cuales se encargan del control de la circuitería interna de un computador (Louden, 2004)

Los lenguajes de programación se clasifican en:

4.2.2.1 LENGUAJE MÁQUINA

La totalidad de las instrucciones son entendidas por la máquina y no es necesaria alguna traducción posterior para que sea comprendida y pueda ejecutar un programa.

Éstas son escritas como una sucesión de unos y ceros. Cada máquina posee su propio lenguaje de máquina, el cual a su vez tiene su repertorio particular de instrucciones. Las instrucciones en este tipo de lenguaje hacen referencia a los diferentes registros y unidades funcionales de la máquina, e incluyen un código de operación, encargado de ir mostrando la operación que se va ejecutando. Todas estas instrucciones forman las instrucciones del procesador de la máquina.

4.2.2.1.1 Lenguaje de Bajo Nivel (Assembler). En este lenguaje las instrucciones son expresadas en códigos alfabéticos para las operaciones y direcciones simbólicas.

4.2.2.1.2 Lenguaje de Alto Nivel. En los lenguajes de programación de alto nivel tales como C++, Pascal, JAVA, fortran, entre otros, las instrucciones

brindadas a una máquina se escriben con palabras similares al lenguaje humano, usualmente en inglés. Esto con el objetivo de facilitar la escritura y la comprensión del programa.

4.2.3 ALGORITMO (Louden, 2004)

Es una forma de resolver un problema en particular llevando a cabo un número finito de pasos.

Es una descripción de los pasos de una tarea usando un método específico. Un algoritmo está diseñado para resolver un problema en particular con variables globales sean cual sean los valores iniciales.

El algoritmo es una serie exacta y detallada de una secuencia de pasos elementales a aplicar, partiendo de valores iniciales del problema para encontrar la solución esperada.

Un algoritmo es completo cuando contempla todas las combinaciones de los datos presentados en la formulación del problema para así poder brindar una solución.

Este debe ser un método global para resolver todos los problemas de la misma clase y por supuesto debe ser solucionado con recursos finitos.

Hay que tener en cuenta que un problema puede tener diferentes formas de resolverse y por consiguiente cuenta con innumerables algoritmos que permitan llegar a encontrar una solución.

4.2.4 OPTIMIZACIÓN DE ALGORITMOS

En la actualidad la optimización en el diseño de algoritmos es considerada una herramienta poderosa a la hora de minimizar los costos en cuanto a lo que se refiere problema de ingeniería. La realización de un proceso óptimo busca acoplar a un problema una solución que satisfaga las restricciones y limitaciones impuestas.

4.2.5 INTELIGENCIA EVOLUTIVA

La inteligencia evolutiva nace a partir de las necesidades de las personas, quienes obligadas por sus mismas exigencias desarrollan la habilidad de aprovechar los recursos existentes para mejorar la calidad de vida propia y en general de todo el entorno que los rodea. En este campo se estudian los avances teóricos y las

aplicaciones en algoritmos evolutivos, algoritmos genéticos, programación genética, en general, el eje temático de Computación Evolutiva.

En el transcurso de los últimos años los algoritmos de optimización han seguido algunos principios de la naturaleza, y así han demostrado la utilidad de éstos en diversas aplicaciones. Tales fenómenos se encuentran en: Algoritmo de Recocido o Temple Simulado (SA), Redes Neuronales Artificiales (ANNs) y el campo de Computación Evolutiva (EC). Este último caracterizado por tener varios modelos de búsqueda, tales como: Algoritmos Genéticos (GA), Programación Evolutiva (EP), Estrategias de evolución (ES), Programación Genética (GP) y Sistema Clasificador (CFS).

Estas técnicas de Inteligencia Evolutiva han sido empleadas para desarrollar sistemas inteligentes de previsión, soporte a decisión, modelado, control, optimización, clasificación y reconocimiento de patrones en general, que se aplican por ejemplo en sectores como: Industrial, Comercial, Medio Ambiente, entre otros (Sumathi, 2008).

4.2.6 EVOLUCIÓN BIOLÓGICA Y ARTIFICIAL (Sánchez, 2006)

La computación evolutiva está basada en el principio de evolución biológica.

4.2.6.1 EVOLUCIÓN NATURAL

Evolución son todos los cambios originados en los seres vivos desde sus orígenes hasta la actualidad. Los seres vivos evolucionan a través de la interacción de competencia, selección, reproducción y procesos de mutación. La evolución de una población de organismos destaca las diferencias entre genotipo y fenotipo.

4.2.6.2 TEORÍA DE LA EVOLUCIÓN DE DARWIN

Darwin no fue quien comenzó a hablar de evolución en el mundo, pero si fue quien trajo grandes evidencias para demostrarla. Uno de sus intereses fue el estudio de plantas y animales, por ejemplo cruzaba entre sí diferentes tipos de palomas encontrando que las palomas descendientes tenían un gran número de cambios con sólo realizar un cruce artificial y que provenían de un mismo ancestro; con esto pudo decir que muy probablemente ocurrieron cambios semejantes en los organismos por acción de las fuerzas naturales que actuaban sin la intervención del ser humano. Tiempo después se aceptó que todos los animales domésticos se desarrollaron por cruzamiento selectivo a partir de un solo ancestro.

Su conocimiento acerca del tiempo en el que el ser humano habría logrado obtener nuevas y diversas variedades de plantas y animales a través de la selección natural, hizo que argumentara que se podrían obtener miles de cambios

a lo largo de muchos años, y que estos cambios podrían surgir debido a las necesidades que tuvieran los seres. El ejemplo clásico de su teoría es el cuello de las jirafas que surgió de la necesidad que tenían estos animales para conseguir alimento a gran altura, así su cuello fue alargándose hasta tener el tamaño que se conoce. A medida que los descendientes van obteniendo un rasgo físico diferente de sus padres, se van desechando los rasgos que sean inservibles, quedando solo aquellos que traigan un beneficio consigo (Sánchez, 2006).

Finalmente la teoría de Darwin se puede resumir como:

- Las especies no son estáticas, por lo contrario mantienen un continuo proceso de evolución, es decir, van cambiando constantemente; y en este proceso unas pueden sobrevivir y otras pueden ser extinguidas.
- Este proceso evolutivo es lento y constante, nunca se detiene.
- Los seres con rasgos semejantes descienden de un ancestro común.
- La selección natural puede ser resumida en dos fases: La primera añade que las modificaciones en los seres es un proceso espontáneo, es decir no tiene ningún tipo de manipulación. Y la segunda es que los seres mejor dotados, o que cuentan con estas modificaciones van a tener más posibilidades de sobrevivir.

4.2.6.3 SELECCIÓN NATURAL

La selección natural puede ser entendida mejor con un ejemplo: Un miembro de una especie desarrolló la ventaja funcional de aprender a volar; su descendiente heredaría aquella ventaja y de igual manera la heredaría a su cría y así sucesivamente. El miembro inferior (desventaja) de toda ésta cadena irá falleciendo, dejando sólo a los miembros superiores (ventajas).

4.2.7 GENÉTICA

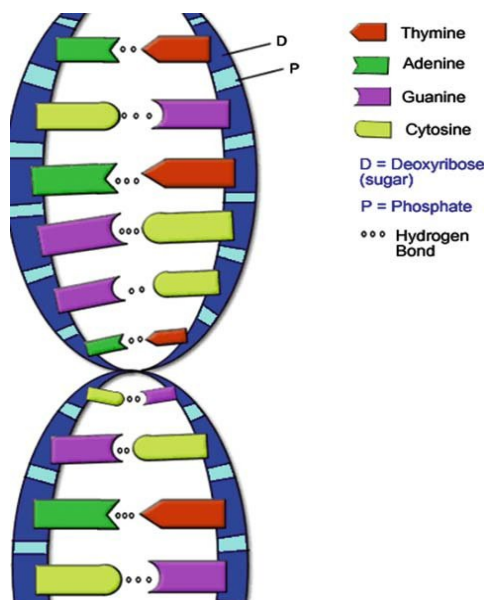
La genética es una ciencia que estudia la herencia y su variación. Trata de comprender como la herencia biológica se transmite de generación en generación, es decir los rasgos y las características que heredan los hijos de los padres y la variación en los organismos vivos.

La información genética de los organismos celulares está contenida dentro de la estructura química de ADN. Los rasgos heredados corresponden a regiones en la secuencia del ADN, llamados genes (Etienne, 2001).

4.2.8 EL ADN

Es una sustancia química también llamada molécula que recopila las instrucciones encargadas de dirigir el completo desarrollo de un organismo, manteniendo su funcionamiento y por consiguiente permitiendo el proceso hereditario. Esta molécula está formada por azúcares denominados desoxirribosas, el ácido fosfórico y los cuatro tipos de bases nitrogenadas llamados adenina, guanina, timina y citosina estructuradas de tal manera que forman una doble hélice, generando el código genético de la célula. La cantidad aproximada de genes que forman las características de las personas son aproximadamente 100.000, los cuales se transmiten de generación en generación.

Figura 1. Estructura del ADN



(Sumathi, 2008)

4.3 MARCO TEÓRICO

4.3.1 INTRODUCCIÓN

A lo largo del siglo 20 y en el nuevo milenio, las naciones comenzaron a traspasar los límites de sus fronteras, expandiendo sus operaciones alrededor del mundo

para habilitar el intercambio global de ideas, opiniones y conocimientos; todo esto gracias a las necesidades y aspiraciones que van surgiendo por cada generación.

Hoy, las nuevas tecnologías están cambiando la manera de vivir y aprender de las personas; evoluciona el pensamiento, evolucionan las aspiraciones, las actitudes y por supuesto evoluciona la inteligencia.

En la investigación actual, la evolución natural se ha convertido en la base de estudio para muchas técnicas que buscan optimizar los recursos en el área de la programación, particularmente el principio de la supervivencia de Charles Darwin. Las técnicas de Computación Evolutiva resumen este principio evolutivo en algoritmos que son utilizados para buscar soluciones óptimas a un problema (Haupt, 2007).

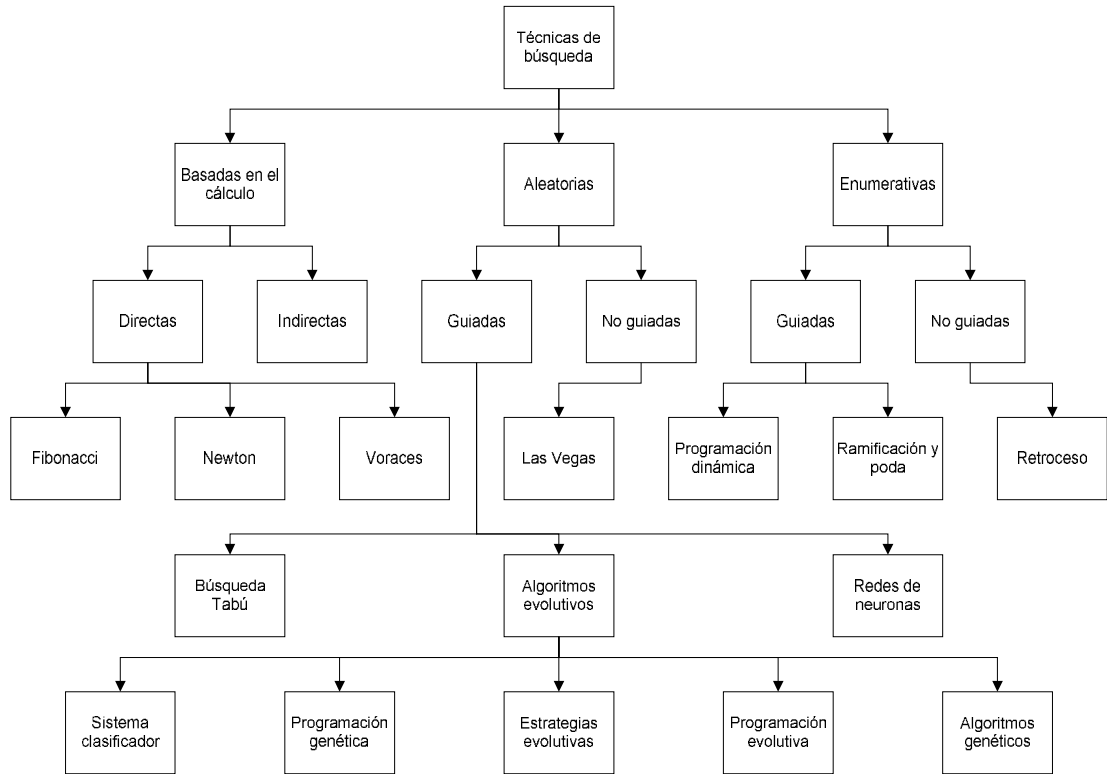
4.3.2 COMPUTACIÓN EVOLUTIVA

La computación evolutiva es una emulación parcial del proceso de selección natural; ésta no requiere un modelo matemático para atacar a un problema en específico. Su alcance en los resultados no puede llegar a ser del todo óptimo pero si cuentan con la cercanía suficiente para llegar a serlo en un determinado tiempo.

Puede ser reconocida gracias a los criterios del progreso iterativo, de crecimiento o desarrollo; al enfoque que tiene hacia la población, manteniendo siempre una guía de búsqueda aleatoria, además realiza todos los procesos en paralelo y por supuesto la inspiración biológica que maneja la identifica.

La siguiente figura presenta las diferentes técnicas de búsqueda en donde se incluyen los algoritmos evolutivos (Kuri, 2000).

Figura 2. Taxonomía de técnicas de búsqueda



(Alba)

4.3.2.1 PARADIGMAS EN COMPUTACIÓN EVOLUTIVA

Tabla 1. Paradigmas de computación evolutiva

| PARADIGMA | CREADOR |
|--------------------------|------------------------------------|
| Algoritmos Genéticos | J.H. Holland |
| Programación Evolutiva | L.J. Fogel, A.J. Owens, M.J. Walsh |
| Estrategias de Evolución | I. Rechenberg and H.P. Schwefel |

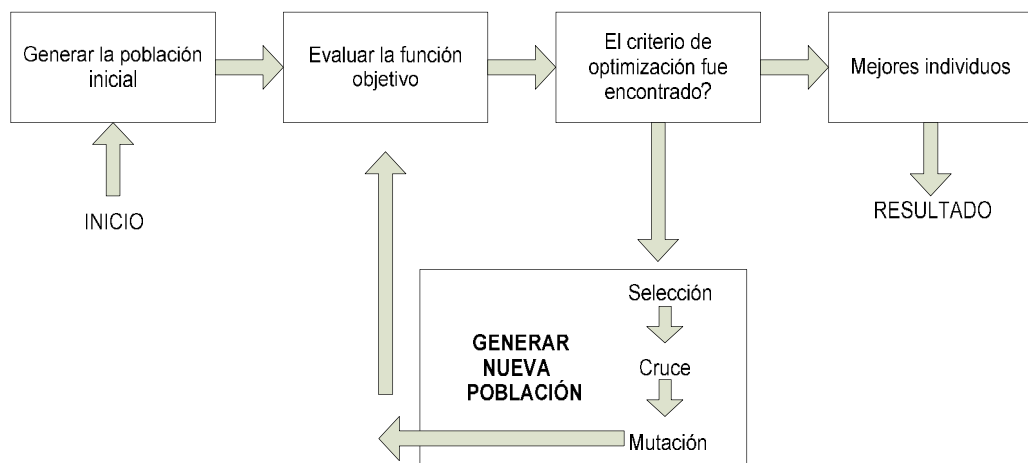
4.3.3 ALGORITMOS EVOLUTIVOS

Este tipo de algoritmos generan un conjunto de posibles soluciones, evaluando el desempeño del sistema para cada individuo. Si el algoritmo verifica que se ha alcanzado algún criterio de convergencia el programa llega a su fin, pero si por lo contrario la verificación de convergencia resulta negativa se procede a seleccionar los mejores individuos de acuerdo con su evaluación. Finalmente se modifica a cada uno de los individuos en base a su desempeño para obtener un nuevo conjunto de posibles soluciones y en este punto el proceso en general vuelve a realizar los mismos pasos mencionados anteriormente (Goldberg, 1989).

4.3.3.1 ESTRUCTURA DE LOS ALGORITMOS EVOLUTIVOS

En el inicio del algoritmo el número de individuos de la población es seleccionado aleatoriamente. La figura 3 muestra la estructura de un algoritmo evolutivo simple.

Figura 3. Estructura de algoritmos evolutivos



(Goldberg, 1989)

(Gold

Los orígenes de estos algoritmos evolutivos se remontan a finales de los años 50, desde ésta época tres metodologías han surgido: Programación Evolutiva (EP), Estrategias de evolución (ES) y Algoritmos Genéticos (GA).

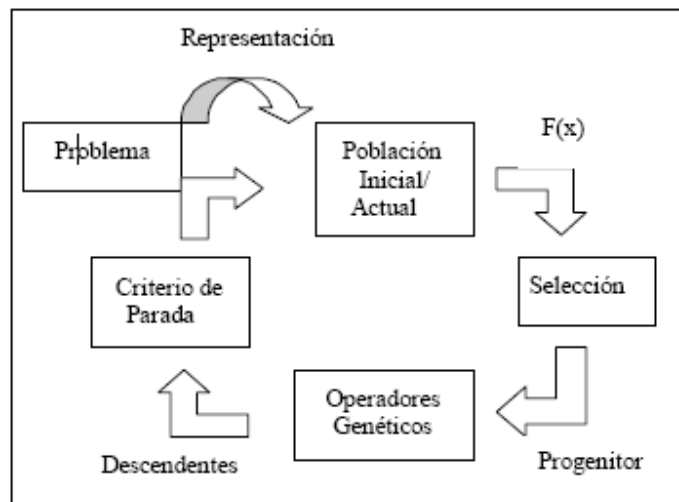
4.3.4 ALGORITMOS GENÉTICOS (Haupt, 2007)

El algoritmo genético es una técnica utilizada para la solución de problemas de búsqueda y optimización, basada en el planteamiento que hizo Charles Darwin sobre la supervivencia de los más fuertes por medio de la selección natural, es decir se inspira en la evolución biológica. Este método algorítmico imita

precisamente este proceso creando soluciones para problemas actuales, evolucionando estas soluciones con el fin de ir encontrando los valores óptimos necesarios para el problema.

El algoritmo parte de la creación dentro de la máquina de una población de individuos representados por cromosomas, los cuales pasan por un proceso de simulación llamado evolución, esta evolución comienza desde una población aleatoria. En cada generación la selección natural será la encargada de elegir a los individuos más aptos, realizándoles modificaciones y mutaciones con el fin de ser transmitidos a otras generaciones. Estos algoritmos realizan un proceso continuo el cual repite ciclos de evolución controlados siempre por los criterios de parada. La figura 4 ilustra el ciclo realizado por un algoritmo genético.

Figura 4. Ciclo realizado por un algoritmo genético



(Goldberg, 1989)

Los algoritmos genéticos tienen terminación, renovación, evolución, inicio e impresión, los cuales también son algoritmos que actúan como moderadores. La síntesis de esta clase de algoritmos es la siguiente: Definición de tipos de algoritmos, es decir se identifica el tipo de algoritmo y los algoritmos moderadores en los cuales se incluyen el prototipo inicial, de impresión, de evolución, de terminación y por último el prototipo de renovación (Sumathi, 2008).

- El algoritmo genético inicializador es quien inicia la población y sus respectivos parámetros.

- El algoritmo genético de terminación es una función booleana que decide cuando una condición de terminación es alcanzada. Por defecto se detiene si el número de la población es 100.
- El algoritmo genético de renovación se utiliza para decidir que población debe sobrevivir.
- El algoritmo genético de evolución es el encargado de realizar el proceso de evolución de una población
- El algoritmo genético de impresión proporciona el formato de salida del algoritmo genético construido. Por defecto todos los miembros del algoritmo se imprimen. Este algoritmo moderador es llamado cuando la condición de terminación del algoritmo genético es alcanzada.

4.3.4.1 ELEMENTOS DE UN ALGORITMO GENÉTICO (Goldberg, 1989)

Los elementos básicos de un algoritmo genético son los siguientes:

4.3.4.1.1 POBLACIÓN

Conjunto de individuos que representan todas las posibles soluciones a un problema dado, éstos forman una cadena de bits que se evalúan después de la decodificación a números reales o enteros, representando las variables existentes de un problema. A partir del proceso de selección natural que se aplica sobre la población inicial y el uso de operadores genéticos como el cruce y la selección, se originan los descendientes de una nueva generación.

4.3.4.1.2 GEN

Es la unidad funcional de la herencia, que tiene la capacidad de almacenar información capaz de soportar mutación, replicación y expresión.

4.3.4.1.3 CROMOSOMA

El material genético se encuentra empaquetado en unidades discretas llamados cromosomas. Estos son los portadores de la mayor parte de este material y condicionan la organización de la vida y los caracteres heredados por cada especie. En los algoritmos genéticos es un individuo o elemento de una población, quién representa una posible solución al problema planteado.

4.3.4.1.4 FUNCIÓN FITNESS O DE APTITUD

Se considera una expresión matemática encargada de evaluar la aptitud o calidad de los individuos que conforman una población. A la hora en que una función Fitness se define, ésta debe retornar los valores más altos de los individuos que más se aproximan a la solución óptima.

4.3.4.1.5 SELECCIÓN

Escoge cromosomas dentro de la población con el fin de que se reproduzcan. Durante el proceso evaluativo, el gen es decodificado convirtiéndose en un conjunto de parámetros derivados del problema, los cuales proporcionarán la solución al mismo. La solución óptima está dada por la puntuación que tendrán éstos parámetros. A esta puntuación se le denomina fitness. El fitness entonces será el encargado de escoger los cromosomas que se van a reproducir, quienes conformarán la población de la siguiente generación. Después de haberse evaluado el fitness la nueva población creada deberá tener en cuenta que sean transmitidos los mejores rasgos.

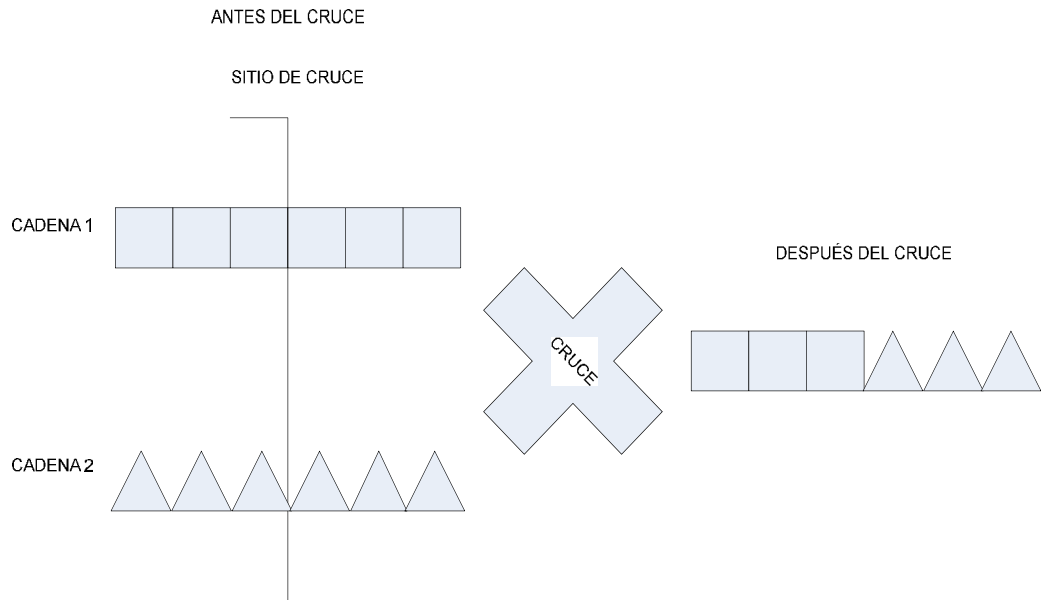
Permiten obtener una nueva generación partiendo de una población inicial. Los más comunes son el crossover o cruce y la mutación.

4.3.4.1.6 CRUCE

Elige un lugar e intercambia las secuencias que se encuentran antes y después de la posición ubicada en el medio de dos cromosomas, a fin de que se produzca una descendencia. Los miembros escogidos son seleccionados aleatoriamente, por lo que sin importar pueden formarse parejas de descendientes de los mismos padres. Este hecho garantiza que un individuo con una excelente puntuación permanezca siempre constante en la población. Ejemplo:

Cruce entre [10010] y [11111] después del segundo lugar. La descendencia que se produce es la siguiente: [10111 y 11010].

Figura 5. Etapas del cruce

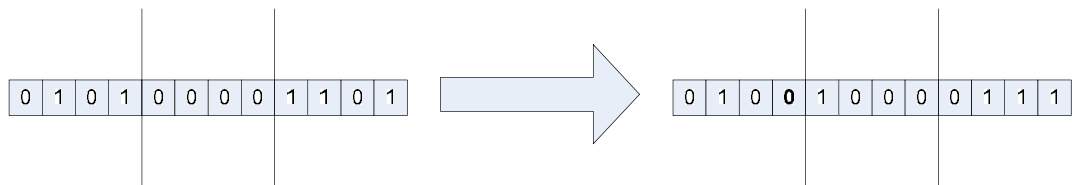


(Annicchiarico, 1996)

4.3.4.1.7 MUTACIÓN

Causa variaciones aleatorias en un cromosoma, las cuales contribuyen a la diversidad genética de la especie. Estas variaciones son poco probables de ocurrir, es decir tienen una frecuencia diminuta, a esta frecuencia se le denomina frecuencia de mutación. Una vez se establezca esta frecuencia se examinan los bits de las diferentes cadenas sólo en el momento en que se produzca un nuevo descendiente, si el número generado al azar está por debajo de esa frecuencia, inmediatamente el bit será cambiado, de lo contrario el bit seguirá igual. La figura 6 ilustra un ejemplo de la mutación en una cadena de bits.

Figura 6. Mutación de una cadena de bits

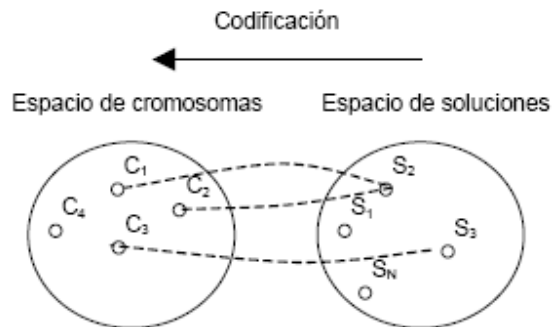


Un algoritmo genético utiliza estas operaciones con el fin de encontrar la solución más apta.

4.3.4.1.8 CODIFICACIÓN

Se requiere que el conjunto codificado se efectúe en un cromosoma. Los genes correspondientes al cromosoma caracterizan diversos parámetros planteados en el problema; si se desea trabajar con estos genes es indispensable codificarlos en una cadena de bits, comúnmente conformada por ceros y unos. El número de bits requerido por cada parámetro dependerá exclusivamente del número de posibles opciones deseado.

Figura 7. Cada cromosoma codifica una posible solución al problema



(Goldberg, 1989)

4.3.4.1.9 FUNCIONAMIENTO

El esquema de funcionamiento de los algoritmos genéticos puede ser descrito como:

- Creación de una población inicial generando individuos aleatoriamente.
- Elección de los individuos más aptos.

Se le da un valor de supervivencia a cada uno de los individuos elegidos y se selecciona a un conjunto de los mismos para que actúen como padres. Luego se empareja un grupo de estos padres con el objetivo de que se presente una descendencia y finalmente se combina la descendencia con la población actual para así poder crear una nueva población.

Los algoritmos genéticos tienen diversas ventajas sobre la aproximación de optimización numérica tradicional:

- 1) Optimización con parámetros continuos o discretos
- 2) No requiere información derivada.
- 3) Búsqueda simultánea de una muestra amplia de los gastos de superficie.
- 4) Trabaja con una gran cantidad de variables.
- 5) Es adecuado para realizar trabajos en paralelo.

4.3.4.2 ALGORITMOS GENÉTICOS PARALELOS (Sumathi, 2008)

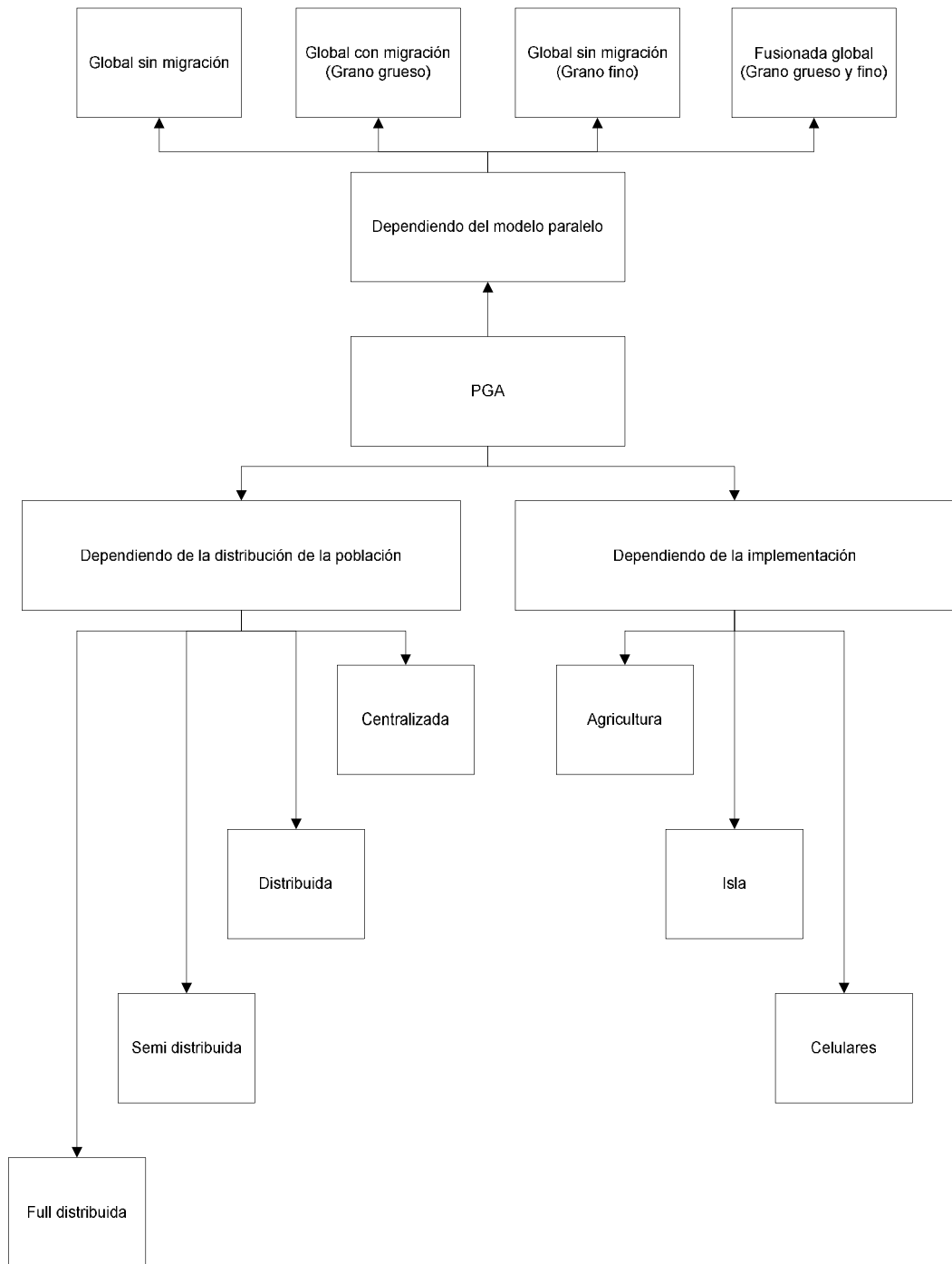
Los algoritmos genéticos como ya se ha dicho, son capaces de encontrar buenas soluciones en una cantidad razonable de tiempo, pero como sus aplicaciones suelen ser en problemas grandes, es decir de gran complejidad, el incremento en el tiempo requerido para encontrar las soluciones esperadas ha sido notorio. Como consecuencia de esto, se han hecho múltiples esfuerzos para realizar algoritmos genéticos más rápidos y una de las principales soluciones es la implementación en paralelo. Los algoritmos genéticos paralelos han mejorado el proceso para hacer frente a problemas más complejos, ya que pueden utilizar más memoria y recursos de la CPU.

4.3.4.2.1 Clasificación de los algoritmos genéticos en paralelo

Hay dos razones para el paralelismo en los algoritmos genéticos:

La primera es lograr el ahorro de tiempo, distribuyendo el esfuerzo de cálculos y la segunda es para beneficiarse de la configuración en paralelo entre el punto de vista algorítmico, en analogía con la evolución paralela natural de las poblaciones distribuidas espacialmente.

Figura 8. Clasificación de algoritmos genéticos en paralelo



(Sumathi, 2008)

4.3.4.2.2 APLICACIONES

Una de las aplicaciones puede ser la optimización de una función simple en donde los cromosomas representan los rangos de variación. Aquí la función de selección está dada por el valor de $f(x)$, en donde el resultado se muta con una probabilidad específica.

4.3.5 ARREGLOS LINEALES DE ANTENAS (Haupt, 2007)

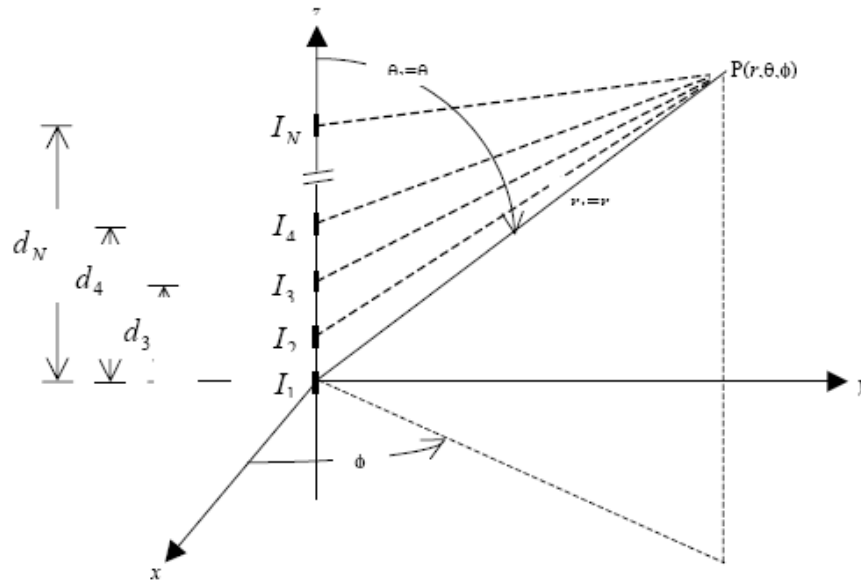
Generalmente el patrón de radiación de un elemento aislado tiene un ancho de haz relativamente grande, y cada elemento individual provee niveles muy bajos de directividad (ganancia). Existen algunas aplicaciones donde es necesario diseñar antenas con muy buenas características de directividad, es decir muy altas ganancias, para alcanzar las demandas actuales de comunicación a larga distancia que han surgido con el desarrollo de las tecnologías de la información.

La ampliación de las dimensiones de una antena aislada conduce frecuentemente a una mejor directividad. Sin embargo, existe otra manera para ampliar las dimensiones de la antena sin necesariamente incrementar el tamaño de los elementos individuales, y consiste en establecer un grupo de elementos radiantes en una configuración eléctrica y geométrica determinada. Esta nueva antena, formada por múltiples elementos se conoce como arreglo.

Los arreglos de antenas, localizadas en una posición específica en donde se excitan adecuadamente, son una alternativa importante para generar patrones de radiación con las características deseadas; por ejemplo la ubicación de los nulos, el ancho de haz, entre otros.

Los arreglos lineales de antenas se caracterizan porque sus elementos se encuentran ubicados a lo largo de una línea recta, con una separación uniforme o no uniforme. La figura 9 ilustra la disposición física de un arreglo lineal de antenas.

Figura 9. Disposición física de un arreglo lineal de antenas



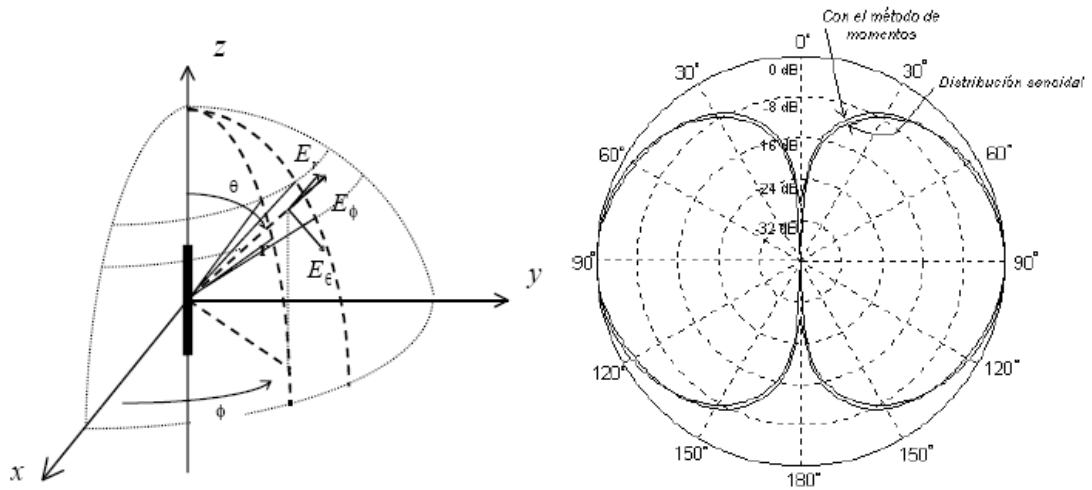
(Balanis, 2005)

Los arreglos son clasificados según la ubicación de sus elementos y las características de las señales de excitación.

4.3.5.1 PATRÓN DE RADIACIÓN DE UN DIPOLO

El dipolo consiste en dos alambres que se encuentran acomodados linealmente y separados por un espacio en el cual se aplica la excitación. Idealmente el diámetro del alambre y la separación entre éstos debe ser de cero, de modo que se va formando un alambre filamental continuo. La figura 10 ilustra un dipolo orientado en el eje z, con una ubicación simétrica con respecto al origen y la figura presenta su patrón de radiación producido al usar una distribución de corriente senoidal (Balanis, 2005).

Figura 10. Dipolo y su patrón de radiación de campo eléctrico



(Balanis, 2005)

En la región de campo lejano, el campo eléctrico producido por un dipolo con una distribución de corriente filamental de forma senoidal está dado por la fórmula 1.

Fórmula 1. Campo eléctrico

$$E_{\theta} \cong J\eta \frac{I_0 e^{-jkr}}{2\pi r} \left(\frac{\cos\left(\frac{kl}{2} \cos\theta\right) - \cos\left(\frac{kl}{2}\right)}{\sin\theta} \right)$$

4.3.5.2 FACTOR DE ARREGLO DE ARREGLOS CON FASE PROGRESIVA

Dentro de los arreglos lineales con separación uniforme, existen desarrollos en los cuales se maneja la amplitud uniforme y la fase progresiva. El factor de arreglo normalizado para arreglos con amplitud uniforme y fase progresiva está dado por la fórmula 2.

Fórmula 2. Factor de arreglo normalizado

$$AF_n = \frac{1}{N} \sum_{n=1}^N e^{j(n-1)\psi}$$

Donde $\psi = kd \cos \theta + \beta$, d es la distancia entre los elementos y β es la fase progresiva. Es posible demostrar que la fórmula 2 se puede escribir como la fórmula 3.

Fórmula 3. Factor de arreglo normalizado

$$AF_N = \frac{e^{j(n-1)\frac{\psi}{2}} \operatorname{sen} N \frac{\psi}{2}}{N \operatorname{sen} \frac{\psi}{2}}$$

4.3.5.3 DIRECTIVIDAD

El patrón de radiación de un dipolo se vuelve más direccional a medida que aumenta su longitud. Cuando la longitud total es mayor que la longitud de onda, el número de lóbulos aumenta y la antena pierde sus propiedades direccionales. La directividad está definida matemáticamente por la fórmula 4.

Fórmula 4. Directividad

$$D = 4\pi \frac{FA^2}{\frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} (FA^2) \operatorname{sen} \theta \, d\theta \, d\varphi}$$

4.3.6 SERIE TRIGONOMÉTRICA DE FOURIER

Una gran cantidad de señales periódicas representadas como $f(t)$, las cuales poseen un periodo definido T , se pueden expresar mediante la Serie Trigonométrica de Fourier, la fórmula 5 muestra la estructura general de esta serie.

Fórmula 5. Estructura general de la serie trigonométrica de Fourier

$$f(t) = \frac{1}{2} a_0 + a_1 \cos(\omega_0 t) + a_2 \cos(2\omega_0 t) + \dots \\ + b_1 \operatorname{sen}(\omega_0 t) + b_2 \operatorname{sen}(2\omega_0 t) + \dots$$

En donde de $\omega_0 = \frac{2\pi}{T}$

Esta serie se puede representar mediante la sumatoria representada en la fórmula 6.

Fórmula 6. Representación de la serie trigonométrica de Fourier

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega_0 t) + b_n \text{sen}(n\omega_0 t)]$$

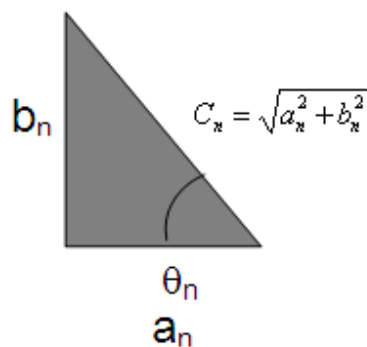
La Serie de Fourier, se puede representar de una forma similar, operando el término $a_n \cos(n\omega_0 t) + b_n \text{sen}(n\omega_0 t)$ se escribe como lo muestra la fórmula 7.

Fórmula 7. Serie de Fourier

$$\sqrt{a_n^2 + b_n^2} \left(\frac{a_n}{\sqrt{a_n^2 + b_n^2}} \cos(n\omega_0 t) + \frac{b_n}{\sqrt{a_n^2 + b_n^2}} \text{sen}(n\omega_0 t) \right)$$

Para comprender de manera simple como la compactación de los coeficientes de esta ecuación, se debe tomar como base la representación de un triángulo rectángulo. La figura 11 ilustra la compactación de los coeficientes de serie mediante la utilización de un triángulo rectángulo utilizando la relación que se demuestra en la fórmula 8.

Figura 11. Coeficientes de una serie relacionados mediante un triángulo.



Fórmula 8. Representación de los coeficientes de un triángulo rectángulo.

$$\frac{a_n}{\sqrt{a_n^2 + b_n^2}} = \cos \theta_n$$

$$\frac{b_n}{\sqrt{a_n^2 + b_n^2}} = \operatorname{sen} \theta_n$$

La fórmula 9 muestra el reemplazo de las nuevas variables.

Fórmula 9. Coeficientes de Fourier simplificados.

$$C_n [\cos \theta_n \cos(n\omega_0 t) + \operatorname{sen} \theta_n \operatorname{sen}(n\omega_0 t)]$$

$$= C_n [\cos(n\omega_0 t - \theta_n)]$$

Para esto, podemos expresar el primer término como $C_0 = \frac{a_0}{2}$.

Finalmente, la serie de Fourier se puede definir como lo muestra la fórmula 10.

Fórmula 10. Representación de la serie de Fourier con coeficientes.

$$f(t) = C_0 + \sum_{n=1}^{\infty} C_n [\cos(n\omega_0 t - \theta_n)]$$

En donde $C_n = \sqrt{a_n^2 + b_n^2}$ y $\theta_n = \tan^{-1}\left(\frac{b_n}{a_n}\right)$

4.3.6.1 COMPONENTES Y ARMÓNICAS

Una función periódica, denotada mediante la expresión $f(t)$, se puede representar como la suma de varios componentes sinusoidales de diferentes frecuencias $\omega_n = n\omega_0$.

En base a esto, se definen todos los términos que involucra la representación de la Serie Trigonométrica de Fourier.

Componente sinusoidal de frecuencia ($n\omega_0 = C_n \cos(n\omega_0 t + q_n)$), denominada la n -ésima armónica de la señal $f(t)$.

Primera armónica ($n=1$), denominada componente fundamental.

Frecuencia $\omega_0 = 2\pi f_0 \Rightarrow \omega_0 = \frac{2\pi}{T}$, denominada frecuencia angular fundamental.

Componente de frecuencia cero (C_0), llamada componente de corriente directa (cd) y correspondiente al valor promedio de $f(t)$ en cada periodo.

Coefficientes C_n y los ángulos q_n son respectivamente las amplitudes y los ángulos de fase de cada una de las armónicas.

4.3.6.2 ORTOGONALIDAD DE SENOS Y COSENOS

Los conjuntos de funciones $f_k(t)$ son ortogonales en el intervalo $a < t < b$ si dos funciones cualesquiera $f_m(t)$, $f_n(t)$ de dicho conjunto cumplen la fórmula 11 el siguiente es un conjunto de una infinidad de funciones ortogonales en el intervalo $-T/2 < t < T/2$.

Fórmula 11. Conjunto de funciones ortogonales.

$$\int_a^b f_m(t) f_n(t) dt = \begin{cases} 0 & \text{para } m \neq n \\ r_n & \text{para } m = n \end{cases}$$

Es decir que para cualquier valor de $\omega_0 = \frac{2\pi}{T}$, se obtiene:

$$1, \cos \omega_0 t, \cos 2\omega_0 t, \cos 3\omega_0 t \dots \sin \omega_0 t, \sin 2\omega_0 t, \sin 3\omega_0 t \dots$$

4.3.6.3 CÁLCULO DE LOS COEFICIENTES DE LA SERIE

Para el cálculo de los coeficientes de la Serie, de una función periódica, representada mediante la $f(t)$ que se muestra en la fórmula 12, primero se deben calcular los coeficientes $a_0, a_1, a_2, \dots, b_1, b_2, \dots$

Fórmula 12. Representación general de los coeficientes de la serie de Fourier.

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega_0 t) + b_n \text{sen}(n\omega_0 t)]$$

Para realizar esto, se debe considerar la ortogonalidad que poseen las funciones seno y coseno:

1) Multiplicando ambos miembros por $\text{Cos}(n\omega_0 t)$ y realizando la integral entre los intervalos $-\frac{T}{2}$ y $\frac{T}{2}$, se obtiene la fórmula 13.

Fórmula 13. Primer coeficiente de Fourier.

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega_0 t) dt \quad n = 0, 1, 2, 3, \dots$$

2) Este mismo proceso lo debemos hacer multiplicando por $\text{Sen}(n\omega_0 t)$ e integrando $-\frac{T}{2}$ y $\frac{T}{2}$, como se muestra en la fórmula 14.

Fórmula 14. Segundo coeficiente de Fourier.

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \text{sen}(n\omega_0 t) dt \quad n = 1, 2, 3, \dots$$

3) Finalmente, debemos integrar la función entre $-\frac{T}{2}$ y $\frac{T}{2}$, representado en la fórmula 15.

Fórmula 15. Función de integración.

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} f(t) dt$$

Se debe tener en cuenta que el intervalo de integración no necesita ser simétrico respecto al origen.

Gracias a que la ortogonalidad de las funciones seno y coseno no sólo se da en el intervalo de $-\frac{T}{2}$ y $\frac{T}{2}$, sino en cualquier intervalo que cubra un periodo completo, se puede realizar el cálculo en cualquier otro intervalo que cumpla este requisito.

4.4 ESTADO DEL ARTE

La programación evolutiva es un campo de investigación constante la cual trata de desarrollar inteligencia artificial basada en modelos matemáticos implementados en un sistema (Sumathi, 2008).

El desarrollo constante de estas técnicas, permite la formulación de nuevos algoritmos que emulan los mecanismos de la evolución biológica de una manera más acertada.

4.4.1 EXPRESIÓN GÉNICA DE PROGRAMACIÓN (GEP)

La Expresión Génica de Programación (GEP), es un nuevo algoritmo evolutivo que toma muchas formas: expresiones matemáticas, redes neuronales, árboles de decisión, polinomios constructivos, expresiones lógicas, etc.

Sin importar la complejidad del problema presentado, el algoritmo GEP, codifica la información en cromosomas lineales y la traduce a estructuras ramificadas. Así, en GEP, el Genotipo (cromosomas lineales) y el fenotipo (la expresión ramificada) son entidades independientes, tanto estructural como funcionalmente, esto los convierten en 2 factores triviales, que contribuyen con el óptimo diseño de modelos de simulación precisos enfocados a la inteligencia artificial (Ferreira, 2006).

4.4.2 SISTEMAS DE PREVISIÓN DE APRENDIZAJE (BUTZ, 2002)

Los Sistemas de previsión de Aprendizaje son nuevos modelos de de aprendizaje en sistemas adaptativos, basados en algoritmos genéticos. Este modelo de previsión, especifica todos los posibles efectos en un entorno respecto a determinadas situaciones. Se puede utilizar para simular el comportamiento adaptativo de cierta población anticipadamente.

Los Sistemas de previsión de Aprendizaje destacan la función de los algoritmos genéticos y evalúa la influencia de los sistemas cognitivos, brindando

características de optimización aún mayor a la solución de un problema presentado, con características como:

- Mayor rapidez de reacción.
- Comportamiento adaptativo de aprendizaje.
- Mecanismos de atención, con el fin de tener mayor probabilidad de supervivencia.
- La aplicación de un módulo de motivación

4.5 LIMITACIONES Y ALCANCES

Los algoritmos genéticos son una técnica robusta que permite tratar con éxito una gran cantidad de problemas con aplicaciones a diferentes áreas de trabajo, incluyendo aquellas en las que otros métodos encuentran una gran variedad de dificultades.

Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima, del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

Si para un determinado problema, existen técnicas especializadas para la encontrar una solución, es posible que está técnica supere al algoritmo genético, tanto en rapidez como en eficacia. El gran campo de utilización de los algoritmos genéticos, son aquellos problemas para los cuales no existe una técnica especializada para encontrar una solución adecuada. Aunque se debe tener en cuenta que si existiera dicha técnica, y está funcionara bien, pueden efectuarse mejoras cuando son analizadas mediante algoritmos genéticos.

Cuando se trabaja con algoritmos genéticos, la representación del problema resulta ser uno de los ítems más que requieren más atención. Es necesario identificar, por un lado, las variables adecuadas para ser analizadas en el criterio de selección, y también las variables deben ser dejadas fuera de proceso del algoritmo genético.

El proceso de determinar los parámetros convenientes para que funcione el algoritmo genético (tamaño de la población, criterio de cruce, mutación y selección) generalmente requiere de mucho tiempo de análisis.

Como en cualquier procedimiento de optimización, una buena identificación de cada una de las variables del problema, permite que el algoritmo no converja en un mínimo (o en un máximo) local.

Teniendo en cuenta esto, se recomienda emplear los algoritmos genéticos en sistemas que son parcialmente desconocidos, no-lineales o ruidosos.

Lo más importante en este caso debe ser la capacidad de poder programar correctamente la función de selección. En la mayoría de los casos esta acción resulta ser la más ardua del trabajo. Se debe recordar que la función de selección puede ser continua o discontinua y cambiante en el tiempo.

5. DISEÑO METODOLÓGICO

El desarrollo del proyecto está basado en el método científico, el cual incluye una fase de iniciación, hipótesis (planteamiento del problema), experimentación (desarrollo del problema) y conclusiones.

Se hizo uso de una investigación experimental, en la cual se recolectó información relacionada con los conceptos básicos de algoritmos genéticos; en este punto se logró comprender las ventajas y desventajas que ofrece la utilización de los algoritmos genéticos en la optimización de problemas de ingeniería; para esto fue necesario tener en cuenta las herramientas que facilitan el aprendizaje de los algoritmos evolutivos tales como el Toolbox de MATLAB.

A continuación se realizó una serie de pruebas pertinentes para llevar a cabo exitosamente las aplicaciones propuestas. Estas pruebas permitieron desarrollar las habilidades necesarias que facilitaron el proceso de aprendizaje necesario para culminar con éxito la investigación.

Posteriormente esta investigación fue ordenada, analizada y modelada con el fin de promover que en La Fundación Universitaria San Martín se realicen futuras investigaciones basadas en éste proyecto.

6. DESARROLLO

Debido a la falta de conocimiento existente en la Fundación Universitaria San Martín con respecto al eje temático de algoritmos genéticos, se debió realizar un proceso secuencial que en primer lugar permitiera comprender que tan poderosa es ésta herramienta. Para esto fue necesario realizar un análisis comparativo entre la programación habitual y los algoritmos genéticos, para así lograr identificar las ventajas y desventajas que ofrece la utilización de los algoritmos genéticos para la solución de una gran variedad de problemas presentes en la vida cotidiana.

Posteriormente, con base en el estudio adquirido, se realizaron un par de aplicaciones que simulan el proceso que un algoritmo genético realiza.

Finalmente se desarrolló un tutorial Web que recopila toda la información adquirida, con el fin de permitir que se realicen posteriores estudios con base al adelanto investigativo realizado.

A continuación se muestra en detalle el desarrollo de cada uno de los objetivos planteados:

6.1 DESARROLLO DEL OBJETIVO NÚMERO UNO

El objetivo número uno consiste en realizar un análisis comparativo entre los algoritmos genéticos y la programación habitual. A continuación se presenta su respectivo desarrollo.

Tabla 2. Comparación entre la programación habitual y los algoritmos genéticos

| PROGRAMACIÓN HABITUAL | ALGORITMOS GENÉTICOS |
|---|--|
| Optimiza únicamente parámetros continuos. Es decir, que admite valores intermedios. | Optimiza con parámetros continuos o discretos. Puede operar con valores intermedios o valores observables sucesivos. |
| Se crean limitaciones al trabajar con una gran cantidad de variables ya que el procesamiento de los cálculos se convierte más complejo. | Trabaja con un gran número de variables. Su tiempo de computo es reducido gracias a su atributo de paralelismo. |
| Realiza cálculos de forma secuencial. | Ejecutan simultáneamente los cálculos para obtener diferentes |

| | |
|--|--|
| | soluciones |
| Proporcionan una única solución a un problema. Si el resultado obtenido no es el apropiado, se debe desechar todo el proceso e iniciar de nuevo. | Proporcionan una lista de parámetros óptimos, no sólo una única solución. Si el resultado obtenido no es el esperado, este no es desechado, sino que utiliza las características de este para buscar una solución. |
| Es necesario brindarle información previa acerca del problema que se intenta resolver. Si no se posee esta información, el algoritmo no funciona. | No necesitan conocimientos específicos sobre el problema que intentan resolver. Utilizan el criterio de selección para evaluar los resultados obtenidos. |
| Cuando son utilizados para resolver problemas de optimización o maximización de un objetivo, pueden crear falsas soluciones al analizar únicamente un máximo local, más no un máximo global. | Cuando se ejecutan para resolver problemas de optimización y maximización de una función objetivo, pueden trabajar mediante el análisis de los máximos locales y de los máximos globales. |
| Utilizan operadores determinísticos. Es completamente predictivo si se conocen sus entradas. | Utilizan operadores probabilísticos. Es decir, que basa su resultado en la toma de algunas decisiones al azar. |
| Gracias a la estructura como están diseñados, siempre convergen hacia una solución. | Pueden tardar mucho tiempo en converger, o no converger en absoluto hacia ninguna solución. |
| El tiempo de convergencia está determinado por la complejidad de los cálculos que debe realizar. | Pueden converger prematuramente debido a una serie de problemas de diversa índole. Complejidad de la solución, tamaño de la población inicial, criterio de selección entre otros. |
| Están diseñados para resolver problemas lineales únicamente. Los problemas lineales son aquellos que presentan una aptitud independiente para cada componente. Es así como cualquier mejora en un proceso, da como resultado una mejora en todo el | Gracias a la estructura robusta que poseen, pueden resolver problemas que presentan no linealidad. La no linealidad se presenta donde cambios múltiples en combinación pueden conducir hacia una solución óptima en todo el sistema. |

| | |
|---|--|
| sistema. | |
| Pueden trabajar con un lenguaje de programación básico. | Deben trabajar sobre un lenguaje de programación robusto, debido a que debe ser capaz de tolerar cambios aleatorios para no producir constantemente errores fatales o resultados sin sentido. |
| Son perfectos en problemas que pueden resolverse de manera analítica ya consumen mucho menos tiempo y potencia computacional. | Debido a la gran cantidad de cálculos que deben resolver, no son recomendados en problemas que pueden resolverse de manera analítica, ya que el tiempo de resolución de problemas puede ser mayor. |

6.2 DESARROLLO DEL OBJETIVO NÚMERO DOS

El objetivo número dos consiste en identificar las ventajas y desventajas que ofrece la utilización de algoritmos genéticos y sus aplicaciones, para el aprovechamiento correcto de la programación avanzada. A continuación se presenta su respectivo desarrollo.

6.2.1 CARACTERÍSTICAS DE LOS ALGORITMOS GENÉTICOS

- Optimiza con parámetros continuos o discretos.
- No requiere la información derivada.
- Trabaja con un gran número de variables.
- Realiza cálculos en paralelo.
- Optimiza variables con superficies de costo sumamente complejas.
- Proporciona una lista de parámetros óptimos, no sólo una única solución.
- Puede codificar los parámetros, y la optimización es hecha con los parámetros codificados.
- Trabajos con datos numéricamente generados, datos experimentales, o funciones analíticas.
- No necesitan conocimientos específicos sobre el problema que intentan resolver.

- Ejecutan simultáneamente los cálculos para obtener diferentes soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Cuando se ejecutan para resolver problemas de optimización y maximización de una función objetivo, pueden trabajar mediante el análisis de los máximos locales y de los máximos globales, evitando así crear las falsas soluciones que podrían presentar las técnicas tradicionales al enfrentarse a un problema de una gran magnitud.
- Los algoritmos genéticos utilizan operadores probabilísticos, en vez de utilizar operadores determinísticos, típicos en otras técnicas.
- Pueden tardar mucho tiempo en converger, o no converger en absoluto hacia ninguna solución, dependiendo en cierta medida de los parámetros que se utilicen como tamaño de la población, número de generaciones, cantidad de iteraciones realizadas, adecuación de los operadores genéticos definidos como cruce, mutación y el criterio de selección utilizado.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

6.2.2 VENTAJAS DE LOS ALGORITMOS GENÉTICOS

Una de las características más importantes de los algoritmos genéticos radica en su propiedad de trabajar intrínsecamente de forma paralela.

Los algoritmos tradicionales trabajan de forma secuencial y sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo. Si la solución obtenida resulta ser sub-óptima, el algoritmo debe desechar el proceso realizado e iniciar de nuevo.

Gracias a la múltiple descendencia que poseen los algoritmos genéticos, estos pueden explorar el espacio de soluciones en múltiples direcciones a la vez. Así, si una solución resulta ser un callejón sin salida, el algoritmo genético puede eliminarlo fácilmente y continuar el trabajo por caminos más prometedores, brindando una mayor probabilidad en cada ejecución de encontrar la solución.

La ventaja que ofrece el algoritmo genético al trabajar en paralelo, consiste en que al evaluar la aptitud de una cadena particular. Un algoritmo genético explora al mismo tiempo cada uno de los espacios a los que pertenece dicha cadena. Tras muchas evaluaciones, se irá obteniendo un valor cada vez más preciso de la aptitud media de cada uno de estos espacios, cada uno de los cuales contiene muchos miembros. El algoritmo genético puede dirigirse hacia el espacio con los individuos más aptos y encontrar el mejor de ese grupo. En el contexto de los

algoritmos evolutivos, este proceso se conoce como teorema del esquema, y es la ventaja principal de los algoritmos genéticos sobre otros métodos de resolución de problemas.

Gracias al paralelismo con que trabaja el algoritmo genético, se puede evaluar implícitamente muchos esquemas a la vez, los algoritmos genéticos funcionan particularmente bien resolviendo problemas cuyo espacio de soluciones potenciales es muy extenso. Cuando esto ocurre, se presenta un problema denominado “no lineal”.

Los problemas lineales presentan una aptitud independiente para cada componente. Es así como cualquier mejora en el proceso, da como resultado una mejora en todo el sistema.

La mayor cantidad de problemas que se presentan en la vida real poseen características de no linealidad, donde cambios múltiples que, individualmente, son perjudiciales, en combinación pueden conducir hacia una solución óptima en todo el sistema.

La no linealidad que presentan los algoritmos genéticos produce una explosión combinatoria de posibles soluciones para un determinado problema. El paralelismo implícito que poseen los algoritmos genéticos les permite considerar un enorme número de posibilidades. El resultado de esto se refleja en la capacidad de encontrar con éxito resultados óptimos o muy buenos en un corto periodo de tiempo.

Los algoritmos genéticos se desenvuelven bien en problemas que poseen un paisaje adaptativo complejo, es decir, aquellos problemas en los que la función de aptitud cambia con el tiempo, es discontinua, ruidosa, o tiene muchos óptimos locales.

Una gran cantidad de problemas presentes en la vida real poseen un espacio de soluciones muy grande, casi imposible de explorar exhaustivamente; el reto se convierte entonces en evitar los óptimos locales o las soluciones sub-óptimas, es decir, las soluciones que son mejores que todas las que son similares a ella, pero que no son mejores que otras soluciones distintas situadas en algún otro lugar del espacio de soluciones.

Los algoritmos de búsqueda tradicional pueden quedar atrapados en los óptimos locales. Por otro lado, los algoritmos evolutivos han demostrado su efectividad al escapar de los óptimos locales para así descubrir un óptimo global incluso en paisajes adaptativos que poseen un alto grado de complejidad.

Para conseguir este atributo, los algoritmos genéticos utilizan diferentes técnicas que permiten perfeccionar los resultados esperados:

- Paralelismo.
- Selección.
- Mutación.
- Cruzamiento.

Otra ventaja característica de los algoritmos genéticos es su habilidad para manipular varios parámetros simultáneamente. Muchos problemas de la vida real no pueden definirse en términos de un único valor que hay que minimizar o maximizar, sino que es necesario expresar diferentes términos para múltiples objetivos. Esto generalmente produce un ligue entre varias soluciones: uno sólo puede mejorar a expensas de otro.

Los algoritmos genéticos son perfectos para resolver estos problemas: Gracias a su atributo del paralelismo, se producen múltiples soluciones, casi tan buenas como la solución final esperada. Así se crea una serie de soluciones candidatas, una que optimiza un parámetro y otra que optimiza uno parámetro distinto. Posteriormente un agente puede seleccionar una de esas candidatas para su utilización.

Una solución Paretiana o No Dominada es aquella solución en particular a un problema con múltiples objetivos optimiza un parámetro hasta el punto en el que ese parámetro no puede mejorarse más sin causar una correspondiente pérdida de calidad en algún otro parámetro.

Es necesario analizar un detalle importante cuando se trabaja con algoritmos genéticos: Las condiciones iniciales. Para los métodos tradicionales esto puede ser un problema, pero para los algoritmos genéticos es una ventaja. Los algoritmos genéticos no tienen un conocimiento previo del problema que deben resolver. En lugar de utilizar información específica conocida a priori para guiar cada paso y realizar cambios guiados expresamente hacia la mejora, los algoritmos genéticos realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora.

Ya que sus decisiones están basadas por el principio de la aleatoriedad, todos los posibles caminos de búsqueda están abiertos teóricamente a un algoritmo genético.

En contraste, cualquier estrategia de resolución de problemas que dependa de un conocimiento previo, debe inevitablemente comenzar descartando muchos caminos a priori, perdiendo así cualquier solución novedosa que pueda existir.

Los algoritmos genéticos carecen de ideas preconcebidas basadas en creencias establecidas sobre “cómo deben hacerse las cosas” o sobre lo que “de ninguna manera podría funcionar”. En contraste con métodos tradicionales, los algoritmos genéticos no tienen este problema. De manera similar, cualquier técnica que dependa de conocimiento previo fracasará cuando no esté disponible tal conocimiento, pero, de nuevo, los algoritmos genéticos no se ven afectados negativamente por la ignorancia.

Gracias a sus componentes de paralelismo, cruzamiento y mutación, pueden viajar extensamente por todo un paisaje adaptativo, explorando regiones que algoritmos producidos con inteligencia podrían no haber tenido en cuenta, y revelando potencialmente soluciones asombrosas y lo más importante, creativas.

6.2.3 DESVENTAJAS DE LOS ALGORITMOS GENÉTICOS

La consideración más importante que se debe tener en cuenta al crear un algoritmo genético es la forma de representar el problema. El lenguaje utilizado para especificar toda la familia de soluciones candidatas debe ser robusto; es decir, debe ser capaz de tolerar cambios aleatorios y esto no debe producir constantemente errores fatales o resultados sin sentido.

Para conseguir esto existen 2 diferentes formas:

La primera, utilizada por la mayoría de los algoritmos genéticos, consiste en definir a los individuos como listas de números, que pueden ser binarios, enteros o reales. Así, cada número representa algún aspecto de la solución candidata. Si los individuos son cadenas binarias, un 0 o un 1 podría significar la presencia o ausencia de una determinada característica.

Si son listas de números, estos números podrían representar múltiples características. Por ejemplo,

- El orden de las ciudades visitadas en un recorrido dado.
- La situación espacial de componentes electrónicos.
- Los valores con los que se alimenta a un controlador.
- Los pesos de las conexiones en una red neuronal.
- Entre muchas otras características.

Es así como cobra sentido el operador genético de la mutación, la cual implica cambiar estos números, cambiar bits o sumar o restar valores aleatorios.

Para el caso de los algoritmos genéticos, el propio código del programa no cambia; el código es lo que dirige la emulación y hace un seguimiento de los

individuos, evaluando sus aptitudes y quizá asegurándose que sólo se produzcan valores realistas y posibles para el problema dado.

La segunda opción para la programación de algoritmos genéticos es la utilizada por la programación genética, en la cual el propio código del programa sí cambia.

La programación genética representa a los individuos como árboles de código ejecutables que pueden mutar cambiando o intercambiando sub-árboles. Ambos métodos producen representaciones robustas ante la mutación, y pueden representar muchos tipos diferentes de problemas. Esta técnica también ha tenido un éxito considerable.

La idea de representar a las soluciones candidatas de manera robusta no surge de la representación que brinda la naturaleza misma, porque el método de representación utilizado por la evolución, es decir, el código genético, es inherentemente robusto.

Por lo tanto, cualquier cambio en los genes de un individuo siempre producirá un resultado evidente, y por tanto las mutaciones en la evolución tienen mayor probabilidad de producir una mejora.

Esto entra en contraste con los lenguajes creados por el hombre como el inglés, donde el número de palabras con significado es pequeño, comparado con el número total de formas en las que se pueden combinar las letras del alfabeto, y por tanto, es probable que un cambio aleatorio en una frase en inglés produzca un fonema sin sentido.

Es muy importante tener en cuenta que la forma de representar la función objetivo con el fin de poder alcanzar una mayor aptitud y que está verdaderamente signifique una solución mejor para el problema dado. Si se elige mal una función objetivo o se define de manera inexacta, puede que el algoritmo genético sea incapaz de encontrar una solución al problema, o puede acabar resolviendo el problema equivocado.

En ocasiones este error se malinterpreta como una tendencia de engaño del algoritmos genético, aunque en realidad lo que está sucediendo es que el algoritmo genético está haciendo lo que se le pidió hacer, más no lo sus creadores pretendían que hiciera.

Además de elegir bien la función objetivo, es necesario también elegir atentamente los otros parámetros de un algoritmo genético: Tamaño de la población, el ritmo de cruzamiento y mutación, criterio de selección. Si el tamaño de la población es demasiado pequeño, puede que el algoritmo genético no explore suficientemente el espacio de soluciones para encontrar buenas soluciones consistentemente. Si el ritmo de cambio genético es demasiado alto o el sistema de selección se escoge inadecuadamente, puede alterarse el desarrollo

de esquemas beneficiosos y la población puede entrar en catástrofe de errores, al cambiar demasiado rápido para que la selección llegue a producir convergencia.

Un problema que se presenta comúnmente al utilizar algoritmos genéticos se conoce como convergencia prematura. Si un individuo que es más apto que la mayoría de sus competidores emerge muy pronto en el curso de la ejecución, se puede reproducir tan abundantemente que reduzca la diversidad de la población demasiado pronto, provocando que el algoritmo converja hacia el óptimo local que representa ese individuo, en lugar de rastrear el paisaje adaptativo lo bastante a fondo para encontrar el óptimo global.

Varios investigadores, expertos en el eje temático, aconsejan no utilizar algoritmos genéticos en problemas que pueden resolverse de manera analítica, debido a que los métodos analíticos tradicionales consumen mucho menos tiempo y potencia computacional que los algoritmos genéticos y, a diferencia de los algoritmos genéticos, a menudo los métodos analíticos matemáticos brindan una única solución exacta.

6.3 DESARROLLO DEL OBJETIVO NÚMERO TRES

El objetivo número tres consiste en proporcionar a los lectores un tutorial básico de programación, en el cual se recopile la información necesaria acerca de los algoritmos genéticos, sus métodos de representación y las técnicas de selección utilizadas para la resolución de problemas. A continuación se presenta su respectivo desarrollo.

6.3.1 ARQUITECTURA GENERAL

El tutorial de Algoritmos Genéticos fue desarrollado bajo el lenguaje de programación PHP, basado en el sistema de gestión de contenidos (CMS).

CMS son las siglas de Content Management System, que se traduce directamente al español como Sistema Gestor de Contenidos. Como su propio nombre lo indica, es un sistema que nos permite gestionar contenidos. En líneas generales, un CMS permitiría administrar contenidos en un medio digital y para el caso particular que nos ocupa, un CMS permitiría gestionar los contenidos de una web.

Dicho de otra forma, un CMS es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente los CMS trabajan contra una base de datos, de modo que el editor simplemente actualiza una base de datos, incluyendo nueva información o editando la existente.

[<http://www.desarrolloweb.com/articulos/que-es-un-cms.html>]

Visitada el 06 de Diciembre de 2008

Para el diseño del tutorial web de Algoritmos Genéticos utilizamos el CMS llamado Joomla.

Joomla es un CMS que permite construir sitios web de una forma sencilla y no se necesita tener conocimientos avanzados de programación web para implementarlo.

Una de los grandes atributos para seleccionar Joomla como CMS para el diseño del tutorial, es la gran cantidad de información y soporte disponible para resolver todo tipo de inquietudes y problemas que se pueden presentar al implementarlo como aplicación web.

Joomla, resulta muy sencillo para los usuarios sin conocimientos técnicos que deseen añadir o editar contenido, subir imágenes, y gestionar los la información a ser incluida en la Internet. Cualquier usuario con conocimientos básicos sobre un procesador de palabras puede aprender fácilmente a gestionar un sitio bajo Joomla.

Mediante un navegador web, se puede publicar fácilmente los artículos o noticias a ser incluidos, como también, gestionar páginas de interés personal, listados de trabajo, imágenes, videos y también crear una cantidad ilimitada de secciones o páginas de contenido en el sitio web.

[<http://ayuda.joomlaspanish.org/content/view/46/31/>]

Visitada el 12 de Diciembre de 2008

El Tutorial fue implementado en la Web en un hosting de pago con un nombre de dominio gratuito.

6.3.2 ALOJAMIENTO WEB O WEBHOSTING

Para la implementación en la web del tutorial, era necesario un espacio mínimo de Alojamiento de 50MB y al menos 1 base de datos MYSQL para la implementación del CMS.

El servicio de alojamiento Web fue contratado a Ideas Web Colombia (www.IdeasWebColombia.com).

El plan elegido a nuestro proveedor de WebHosting fue el llamado "Paris". Este plan suple las necesidades que requiere la implementación del tutorial web. Sus características son listadas a continuación:

Tabla 3. Características del plan contratado

| PLAN | ESPACIO EN DISCO | TRANSFERENCIA DE DATOS | BASES DE DATOS | PRECIO |
|-------|------------------|------------------------|----------------|----------------|
| PARIS | 100MB | 4000MB | 20 | \$60.000 (AÑO) |

Las características del Hardware y Software del servidor contratado son:

Sistema Operativo: Linux
Versiones: PHP 5.2.5, MySQL 5.0.81.
CPU: Dual Dual-core AMD 2216 (Un total de 4 núcleos)
Memoria Ram: 2GB DDR RAM.
Disco Duro: SATA 7200RPM w/ 8MB Cache.
Conexión: 100Mbit.
Localización: USA, Phoenix - Arizona.

6.3.3 NOMBRE DE DOMINIO

El nombre de dominio elegido para el tutorial fue:

<http://www.tutorialalgoritmosgeneticos.co.cc>

Los nombres de dominio .co.cc son de carácter gratuito. Basta con registrarse en su portal web oficial (<http://www.co.cc>) para adquirir un nombre de dominio disponible y así, realizar la configuración respectiva para apuntarlo directamente hacia el servidor web contratado.

6.3.4 CONTENIDO INTERNO

Para ingresar al sitio web se debe ingresar en el navegador la URL **<http://www.tutorialalgoritmosgeneticos.co.cc>**

La estructura del sitio web le permite al usuario una navegación dinámica. Esto quiere decir que el visitante puede saltar en cualquier momento a cualquier lugar

del Tutorial Web de Algoritmos Genéticos para verificar los conceptos teóricos aquí expuestos y realizar las descargas habilitadas, sin la posibilidad de perder el enlace al cual accedió.

Para esto, el sitio web está organizado de forma jerárquica, distribuido por secciones que permiten una navegación más clara y ordenada.

6.3.4.1 SECCIONES PRINCIPALES

Las secciones que se encuentran listadas en la página web son:

- **Inicio:**

Es la página principal del sitio web. Le brinda al usuario una explicación breve del contenido expuesto. Además se delimita el tipo de usuarios que pueden acceder a la información.

También se muestran enlaces a otras secciones de la página, para que el usuario tenga claro el contenido que puede encontrar en el sitio web.

- **Algoritmos Genéticos:**

Esta sección contiene la información teórica relacionada con el eje temático de Algoritmos Genéticos.

Esta sección pretende que el usuario pueda aclarar los conceptos teóricos que relaciona el estudio de algoritmos Genéticos. Para esto, existen sub-categorías internas que delimitan el concepto global de este eje temático, los elementos necesarios para desarrollar un algoritmo de este tipo, los otros tipos de algoritmos relacionados existentes y las aplicaciones que pueden desarrollarse al utilizar algoritmos genéticos.

- **Tutoriales:**

La sección de tutoriales le brinda al usuario la posibilidad de desarrollar 2 diferentes aplicaciones por medio de Algoritmos Genéticos utilizando el Software de Matlab.

Aquí se demuestra paso a paso la realización de 2 aplicaciones: La primera denominada "Generador De Señales" y la segunda llamada "Diseño De Antena Tipo Panel". Para cada una de las aplicaciones, el usuario visualiza paso a paso el código que permite desarrollar la aplicación final.

Al finalizar el tutorial de cada una de las aplicaciones, el usuario comprenderá claramente el funcionamiento de un Algoritmo Genético básico.

- **Descarga de ejemplos:**

Esta sección contiene 4 archivos de descarga:

Los 2 primeros son los archivos código fuente desarrollados en los tutoriales en el entorno de Matlab. Los cuales contienen una población inicial menor que las de las aplicaciones finales desarrolladas.

Los 2 archivos finales restantes son los códigos fuente del “Generador De Señales” y del “Diseño De Antena Tipo Panel”, los cuales poseen una estructura superior debido al gran número de su población inicial.

- **Contáctenos:**

Esta sección posee un formulario de contacto que permite que el usuario envíe información tal como preguntas, dudas, comentarios, sugerencias y cualquier tipo de mensaje relacionado con el contenido expuesto en el sitio web. Esto permite realizar una retroalimentación de la calidad del contenido expuesto.

6.3.4.2 OTRAS SECCIONES

- **Acerca de nosotros:**

Esta sección contiene información personal acerca de los desarrolladores del sitio web, con el fin de lograr una posterior retroalimentación del tema expuesto y así mismo, brindar los créditos meritorios por la realización del proyecto. Incluyendo los créditos que merece la Fundación Universitaria San Martín al ser partícipe del desarrollo de este tutorial.

- **Enlaces de interés:**

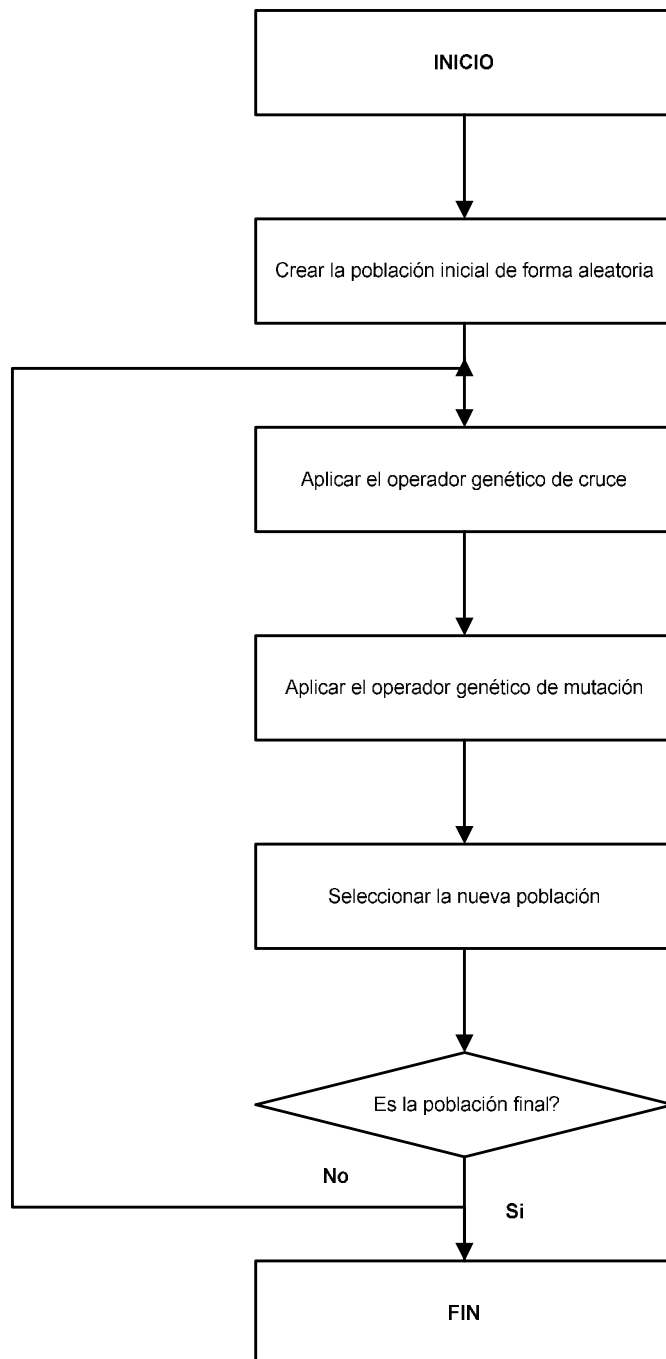
Se enlazan las URL's de los sitios web que permitieron el desarrollo del proyecto TUTORIAL DE ALGORITMOS GENÉTICOS EN EL ENTORNO DE MATLAB, incluyendo cada uno de los sitios web de las herramientas utilizadas como fuentes de información teóricas, plataformas de trabajo, proveedores de Hosting y Dominio, entre otros.

6.4 DESARROLLO DEL OBJETIVO NÚMERO CUATRO

El objetivo número cuatro consiste en implementar un generador de señales con base en algoritmos genéticos en el entorno de Matlab. A continuación se presenta su respectivo desarrollo.

El objetivo del programa es encontrar una señal que posea una potencia promedio definida entre 25 y 26 dB.

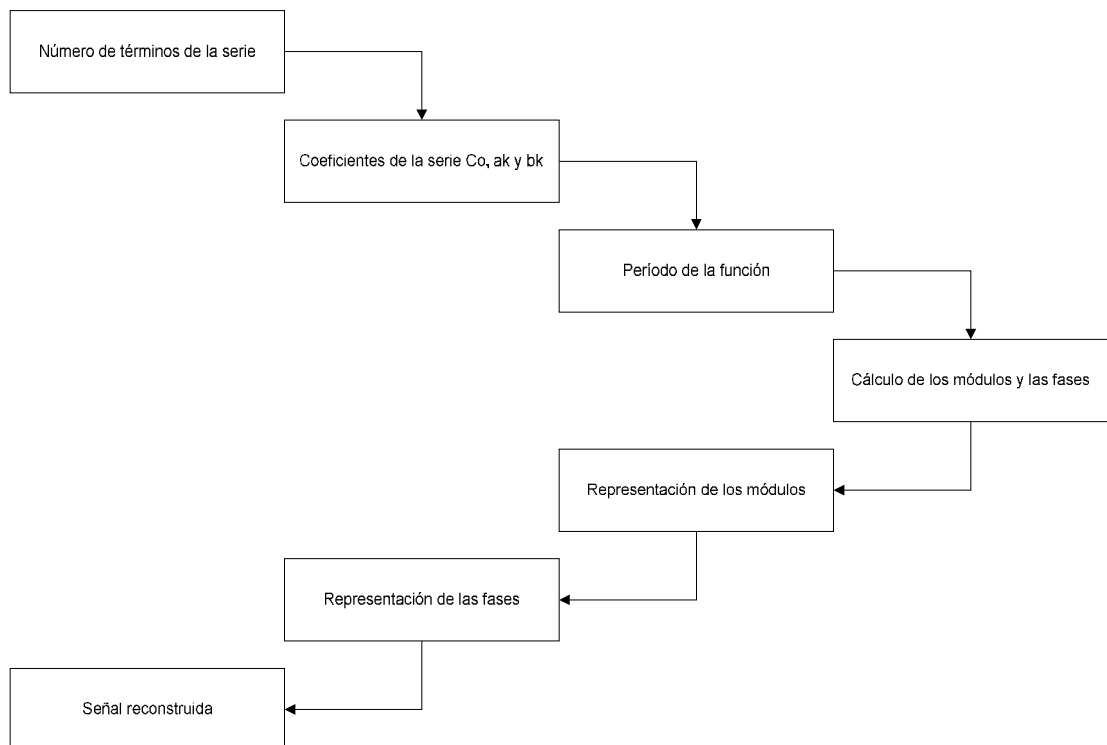
Figura 12. Anatomía del algoritmo genético representado en las aplicaciones.



6.4.1 GENERAR LA POBLACIÓN INICIAL

Las señales aleatorias que conforman la población inicial se implementaron mediante las series de Fourier. Esta población consta de 20 señales generadas mediante una serie de pasos que van a ser descritos a continuación.

Figura 13. Proceso para generar la población inicial



- El número de términos de la serie definido fue 12, es decir que la cantidad de deltas generados que van a ser representados en las gráficas tanto de magnitud como de fase van a ser 12.
- Los coeficientes de la serie son expresiones algebraicas, que van en función de la cantidad de términos definidos. Estos coeficientes son generados aleatoriamente.
- La frecuencia definida para las funciones a ser representadas fue de 4 KHz.
- El número de periodos a representar definido fue de 1.
- El cálculo de los módulos se hizo mediante la fórmula 16. Y el cálculo de la fase mediante la fórmula 17.

Fórmula 16. Magnitud de la serie de Fourier.

$$A = \sqrt{(ak^2) + (bk^2)}$$

Fórmula 17. Fase de la serie de Fourier.

$$Fase = \tan^{-1}\left(\frac{bk}{ak}\right)$$

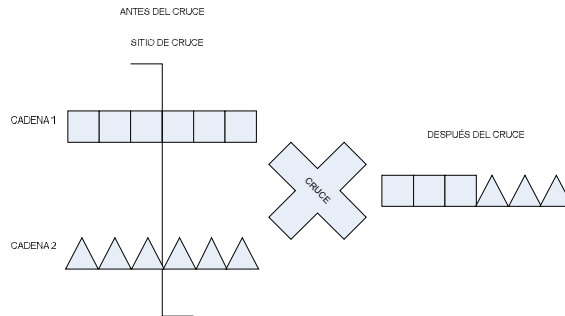
- En la representación de los módulos se grafica la magnitud de las señales. Para demostrar cómo se representan éstos módulos se van a graficar solamente para las dos primeras señales que conforman la población inicial.
- Al igual que la representación de los módulos las fases van a ser graficadas sólo para las dos primeras señales que conforman la población inicial.
- La señal se reconstruye teniendo en cuenta la magnitud y la fase de cada señal. Se realiza una sumatoria de cada uno de los deltas de la magnitud y de la fase.

Los pasos descritos anteriormente se realizan 20 veces, con el fin de representar las 20 señales aleatorias que conforman la población inicial. Estas señales generadas simulan a los padres, quienes conforman la primera generación.

6.4.2 OPERADOR DE CRUCE

El criterio de cruce seleccionado se ilustra en la figura 14.

Figura 14. Gráfica del criterio de cruce utilizado para el generador de señales.



(Annicchiarico, 1996)

El cruce como lo ilustra la figura 14, intercambia información entre la magnitud y la fase de dos señales. La selección de las dos señales que van a ser cruzadas se hace de manera aleatoria, es decir que el cruzamiento puede realizarse entre cualquiera de las 20 señales que conforman la población inicial. La cantidad de señales que resultaran de los cruces van a ser 20. Este número fue definido así, con el fin de que la segunda generación fuera del mismo tamaño que la primera generación.

La tabla 4 muestra cómo se seleccionan las señales que van a realizar el cruce.

Tabla 4. Selección de la señales que van a realizar el operador de cruce.

| q | r | Cruce |
|---------------------------------------|--------------------------------------|--|
| Va desde la señal 1 hasta la señal 20 | Toma valores aleatorios entre 1 y 20 | |
| [Mag. Señal1] [Fase Señal1] | [Mag. Señal5] [Fase Señal5] | [Mag. Señal1-Mag. Señal5] [Fase Señal1-Fase Señal 5] |
| [Mag. Señal2] [Fase Señal2] | [Mag. Señal19] [Fase Señal19] | [Mag. Señal2- Mag. Señal19] [Fase Señal2-Fase Señal 19] |

| | | |
|--------------------------------|----------------------------------|---|
| [Mag. Señal3] [Fase Señal3] | [Mag. Señal10] [Fase Señal10] | [Mag. Señal3- Mag. Señal10] [Fase Señal3-Fase Señal10] |
| [Mag. Señal4] [Fase Señal4] | [Mag. Señal1] [Fase Señal1] | [Mag. Señal4- Mag. Señal1] [Fase Señal4-Fase Señal1] |
| [Mag. Señal5] [Fase Señal5] | [Mag. Señal5] [Fase Señal5] | [Mag. Señal5- Mag. Señal5] [Fase Señal5-Fase Señal5] |

Cuando se realiza el intercambio de la magnitud y de la fase de dos señales, se procede a reconstruir la señal con las nuevas magnitudes y fases resultantes. Esta señal tendrá entonces una mitad con la información de la primera señal y la otra mitad con la información de la segunda señal.

6.4.3 OPERADOR DE MUTACIÓN

El operador de mutación va a ser ejecutado en las 20 señales resultantes del cruce. La mutación consiste en realizarle cierta o ciertas modificaciones en alguna parte a las señales de la segunda generación.

El criterio de mutación seleccionado consiste en agregarle un valor de 6 al delta de la magnitud y de la fase que están en una posición ya definida. Esta posición se definió en 6.

La tabla 5 explica con un ejemplo como es el proceso de mutación en una señal.

Tabla 5. Proceso de mutación en una señal.

| SEÑAL | SEÑAL MUTADA |
|--|--|
| Magnitud = [5 1 7 2 4 5 8 9] Fase = [4 3 7 5 1 2 9 1] | Magnitud = [5 1 7 2 4 11 8 9] Fase =[4 3 7 5 1 8 9 1] |

Cuando la mutación de la magnitud y de la fase de las 20 señales está hecha se procede a reconstruir las señales con las nuevas magnitudes y fases resultantes. Estas señales tendrán entonces un cambio en la posición 6 tanto de la magnitud como de la fase.

6.4.4 SELECCIÓN

El criterio de selección utilizado se basa en la potencia de las señales. Por consiguiente, en primer lugar se calcula la potencia de las 20 señales mutadas.

Este criterio que se definió previamente, consiste en seleccionar una señal que tenga una potencia entre los 25 dB y 26 dB.

Si se encontró la señal que se desea con esta potencia, el algoritmo llegará a su fin; pero por el contrario si no se encontró la señal requerida, el algoritmo volverá a realizar los mismos pasos ya descritos desde el punto de cruce, es decir volverá a cruzar las 20 señales padres, para obtener una nueva generación; aplicará a éstas el criterio de mutación y de selección, y si se obtiene la señal requerida con la potencia deseada, llegará a su fin, pero si no se tiene hará de nuevo el mismo proceso, hasta que el requerimiento del programa se cumpla.

6.5 DESARROLLO DEL OBJETIVO NÚMERO CINCO

El objetivo número cinco consiste en diseñar una antena tipo panel a partir del uso de algoritmos genéticos. A continuación se presenta su respectivo desarrollo.

El objetivo del programa es encontrar un patrón de radiación que tenga una ganancia promedio entre los 26 y los 27 dB.

6.5.1 GENERAR POBLACIÓN INICIAL

Los patrones de radiación de las antenas generados de manera aleatoria que conforman la población inicial se implementaron mediante la fórmula 18. Esta población consta de 8 patrones de radiación.

Fórmula 18. Directividad

$$D = \frac{(FA)^2}{\frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} (FA)^2 \sin\theta \, d\theta \, d\varphi}$$

Donde FA es el factor de arreglo.

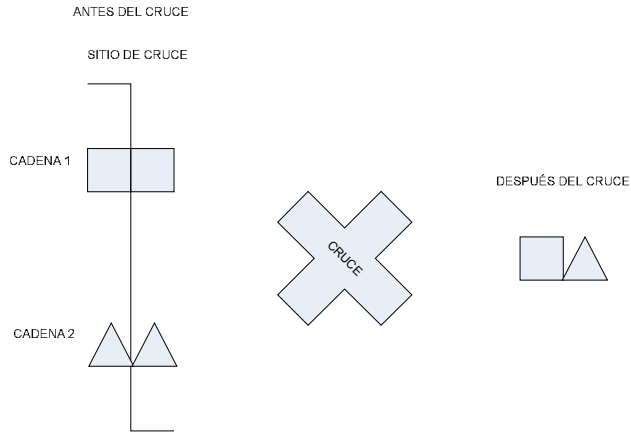
Para generar los patrones de radiación aleatoriamente, se asigna la función *random* de Matlab a las variables d y β , y se grafican con la función *polar*.

6.5.2 OPERADOR CRUCE

El criterio de cruce seleccionado consiste en intercambiar información entre dos patrones de radiación que son seleccionados aleatoriamente, es decir que el cruzamiento puede realizarse entre cualquiera de los 8 patrones de radiación que conforman la población inicial. La cantidad de patrones de radiación que resultaran de los cruces van a ser 8. Este número fue definido así, con el fin de que la segunda generación fuera del mismo tamaño que la primera generación.

La figura15 ilustra el criterio elegido.

Figura 15. Gráfica del criterio de cruce seleccionado para las antenas.



La información intercambiada entre los patrones de radiación es el ángulo de fase y la distancia entre los elementos. La tabla 6 muestra con un ejemplo el criterio de cruce seleccionado.

Tabla 6. Criterio de cruce seleccionado.

| x | y | Patrón de radiación después del cruce |
|--|---|--|
| Va desde el patrón de radiación 1 hasta el 8 | Toma valores aleatorios entre el patrón de radiación 1 al 8 | |
| Antena1: $d1 = 0.6$ $\beta1 = 100$ | Antena 5: $d5 = 0.5$ $\beta5 = 120$ | Cruce: $d1 = 0.6$ $\beta5 = 120$ |
| Antena2: $d2 = 0.75$ $\beta2 = 58$ | Antena 4: $d4 = 0.53$ $\beta4 = 90$ | Cruce: $d2 = 0.75$ $\beta4 = 90$ |

Cuando se realiza el intercambio de la fase y la distancia entre los patrones de radiación de las antenas, se procede a dibujar el nuevo patrón de radiación con las nuevas distancias y fases resultantes. Este patrón de radiación tendrá entonces la distancia de la antena1 y la fase de la antena2.

6.5.3 OPERADOR DE MUTACIÓN

El operador de mutación va a ser ejecutado en los 8 patrones de radiación resultantes del cruce. La mutación consiste en realizarles cierta o ciertas modificaciones en algún parámetro a los patrones de radiación de la segunda generación.

El criterio de mutación seleccionado consiste en agregarle un valor de 0,2 a la distancia entre los elementos del arreglo.

La tabla 7 explica con un ejemplo como es el proceso de mutación en un patrón de radiación.

Tabla 7. Proceso de mutación en un patrón de radiación.

| PATRÓN DE RADIACIÓN | PATRÓN DE RADIACIÓN MUTADO |
|---------------------|------------------------------|
| D1= 0.57 | D1 = 0.57+ 0.2 = 0.72 |

Cuando la mutación de la distancia entre los elementos del arreglo de los 8 patrones de radiación está hecha se procede a graficar los patrones de radiación con las nuevas distancias resultantes.

6.5.4 SELECCIÓN

El criterio de selección utilizado se basa en la ganancia de los patrones de radiación. Por consiguiente, en primer lugar se calcula la ganancia de los 8 patrones de radiación mutados.

Este criterio que se definió previamente, consiste en seleccionar un patrón de radiación que tenga una ganancia promedio entre los 26 y los 27 dB.

Si se encontró el patrón de radiación que se desea con esta ganancia, el algoritmo llegará a su fin; pero por el contrario si no se encontró el patrón de radiación requerido, el algoritmo volverá a realizar los mismos pasos ya descritos desde el punto de cruce, es decir volverá a cruzar los 8 patrones de radiación padres, para obtener una nueva generación; aplicará a éstos el criterio de mutación y de selección, y si se obtiene el patrón de radiación requerido con la ganancia deseada, llegará a su fin, pero si no se tiene hará de nuevo el mismo proceso, hasta que el requerimiento del programa se cumpla.

7. PRUEBAS Y RESULTADOS

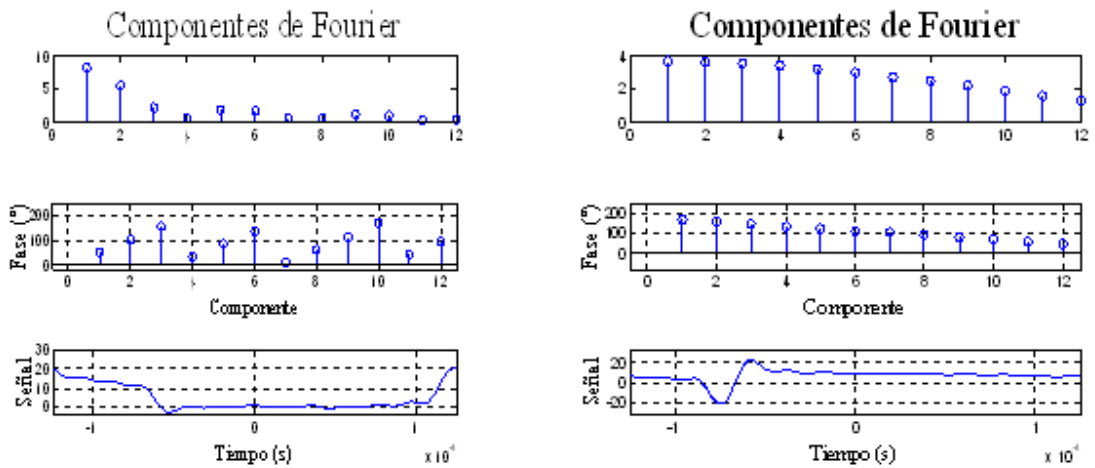
7.1 OBJETIVO NÚMERO UNO.

Implementación de un generador de señales con base en algoritmos genéticos en el entorno de Matlab.

7.1.1 POBLACIÓN INICIAL.

Se ilustran tan sólo dos señales de las 20 que conforman la población inicial generadas aleatoriamente, con sus respectivas gráficas de magnitud y fase.

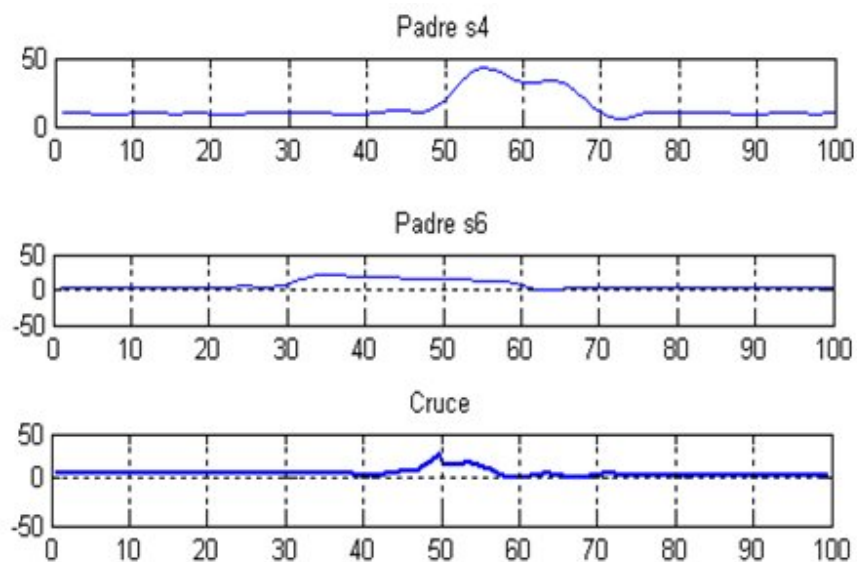
Figura 16. Representación de los módulos y las fases de dos señales.



7.1.2 OPERADOR DE CRUCE.

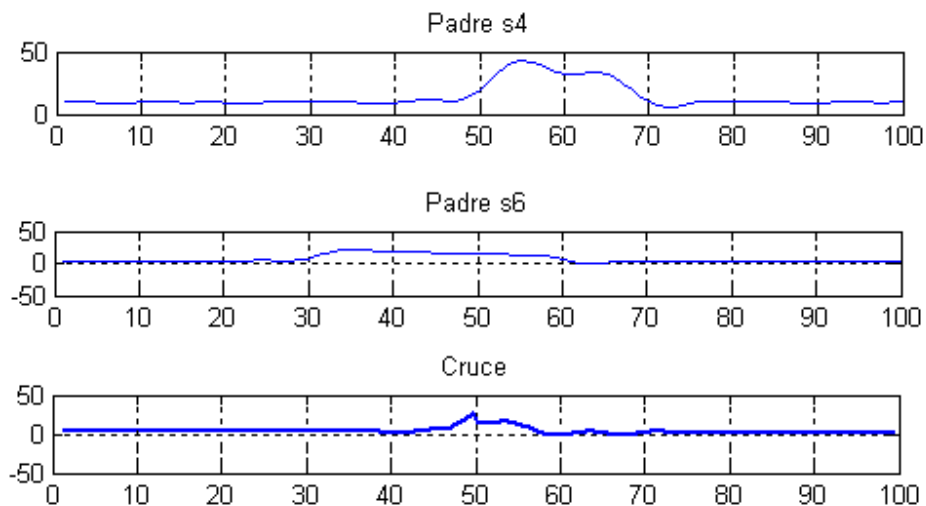
La figura 17 ilustra el cruce entre la señal Padre s2 y la señal padre s19.

Figura 17. Ejemplo del cruce entre dos señales.



La figura 18 ilustra el cruce entre la señal padre s4 y la señal padre s6.

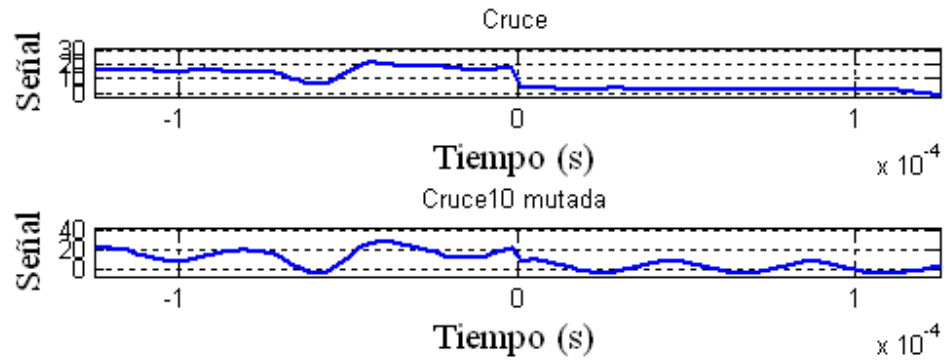
Figura 18. Ejemplo del cruce entre la señal padre s4 y la señal padre s6.



7.1.3 OPERADOR DE MUTACIÓN

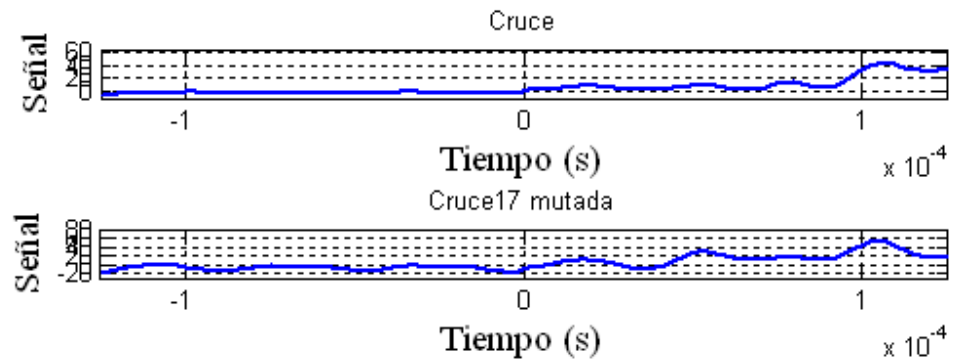
La figura 19 ilustra la mutación realizada a la señal 10.

Figura 19. Gráfica de la mutación realizada a una señal.



La figura 20 ilustra la mutación realizada a la señal 17.

Figura 20. Mutación realizada a la señal 17.

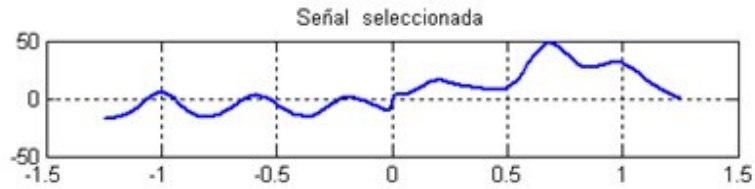


La mutación realizada a las señales consiste en adicionar un valor al delta tanto en magnitud como en fase en una posición definida, lo que origina un aumento en la amplitud de la señal.

7.1.4 SELECCIÓN

La figura 21 muestra la señal que cumple el criterio de selección definido.

Figura 21. Señales seleccionadas que cumplen el criterio de selección definido.



Gracias a la buena definición de los operadores genéticos (cruce, mutación y selección), se encontró la señal que cumple con el criterio que previamente se había definido.

Las tablas 8 a la 22 muestran la cantidad de iteraciones que tuvo que hacer el algoritmo genético, para llegar a la señal que cumpliera el criterio de selección definido.

El proceso hecho por el algoritmo genético para llegar a las señales deseadas, pasó primero por la elección de la población inicial, continuó con el respectivo cruce entre las señales padres o de primera generación, a las cuales se les realizó el proceso de mutación, y finalmente el proceso de selección para definir que señal cumplía con el requerimiento del problema.

Todas las señales seleccionadas en cada ejecución del programa cumplen perfectamente con el criterio de selección de potencia definido.

Tabla 8. Ejecución número 1 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 1 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| F | 25.17 | 4 |

Tabla 9. Ejecución número 2 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 2 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| M | 25.05 | 3 |

Tabla 10. Ejecución número 3 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 3 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| B | 25.78 | 4 |

Tabla 11. Ejecución número 4 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 4 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| H | 25.11 | 4 |

Tabla 12. Ejecución número 5 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 5 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| P | 25.40 | 3 |

Tabla 13. Ejecución número 6 del algoritmo genético para encontrar la señal.

| EJECUCIÓN #6 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| E | 25.26 | 2 |

Tabla 14. Ejecución número 7 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 7 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| O | 25.94 | 3 |

Tabla 15. Ejecución número 8 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 8 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| A | 25.05 | 4 |

Tabla 16. Ejecución número 9 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 9 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| G | 25.05 | 4 |

Tabla 17. Ejecución número 10 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 10 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| D | 25.98 | 3 |

Tabla 18. Ejecución número 11 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 11 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| N | 25.01 | 2 |

Tabla 19. Ejecución número 12 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 12 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| Q | 25.21 | 2 |

Tabla 20. Ejecución número 13 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 13 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| C | 25.52 | 1 |

Tabla 21. Ejecución número 14 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 14 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| F | 25.19 | 2 |

Tabla 22. Ejecución número 15 del algoritmo genético para encontrar la señal.

| EJECUCIÓN # 15 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| E | 25.93 | 2 |

La figura 22 ilustra el número de iteraciones ocurridas por cada ejecución que se le hace al programa. Esta gráfica demuestra que el algoritmo genético encuentra la solución adecuada al problema dado al realizar en su gran mayoría dos ejecuciones. Además, cabe decir que de las 15 ejecuciones realizadas, ninguna fue errada o condujo a soluciones erradas, ya que todas llegaron a la solución requerida, la diferencia fue el número de iteraciones y el tiempo gastado por cada ejecución para encontrar la solución. Por ejemplo, la gráfica 22 ilustra que las ejecuciones 1, 3, 4, 8, y 9 gastaron más tiempo y más iteraciones en encontrar la solución, que el tiempo gastado por la ejecución 13, ya que llegó a ella a la cuarta iteración.

Figura 22. Relación entre el número de ejecuciones y de iteraciones.



Las tablas 23 a la 37 muestran la cantidad de iteraciones que tuvo que hacer el algoritmo genético, para llegar a la señal que cumpliera el segundo criterio de selección elegido, que consiste en encontrar señales que tengan una potencia de 24 a 25 dB.

Tabla 23. Ejecución número 1 para un segundo criterio de selección.

| EJECUCIÓN # 1 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| P | 24.28 | 2 |

Tabla 24. Ejecución número 2 para un segundo criterio de selección.

| EJECUCIÓN # 2 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| R | 24.32 | 4 |

Tabla 25. Ejecución número 3 para un segundo criterio de selección.

| EJECUCIÓN # 3 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| G | 24.51 | 2 |

Tabla 26. Ejecución número 4 para un segundo criterio de selección.

| EJECUCIÓN # 4 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| M | 24.69 | 2 |

Tabla 27. Ejecución número 5 para un segundo criterio de selección.

| EJECUCIÓN # 5 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| I | 24.6 | 3 |

Tabla 28. Ejecución número 6 para un segundo criterio de selección.

| EJECUCIÓN # 6 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| S | 24.36 | 2 |

Tabla 29. Ejecución número 7 para un segundo criterio de selección.

| EJECUCIÓN #7 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| K | 24.28 | 3 |

Tabla 30. Ejecución número 8 para un segundo criterio de selección.

| EJECUCIÓN # 8 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| A | 24.32 | 2 |

Tabla 31. Ejecución número 9 para un segundo criterio de selección.

| EJECUCIÓN # 9 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| N | 24.11 | 2 |

Tabla 32. Ejecución número 10 para un segundo criterio de selección.

| EJECUCIÓN # 10 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| S | 24.32 | 1 |

Tabla 33. Ejecución número 11 para un segundo criterio de selección.

| EJECUCIÓN # 11 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| H | 24.18 | 3 |

Tabla 34. Ejecución número 12 para un segundo criterio de selección.

| EJECUCIÓN # 12 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| Q | 24.62 | 2 |

Tabla 35. Ejecución número 13 para un segundo criterio de selección.

| EJECUCIÓN # 13 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| A | 24.74 | 4 |

Tabla 36. Ejecución número 14 para un segundo criterio de selección.

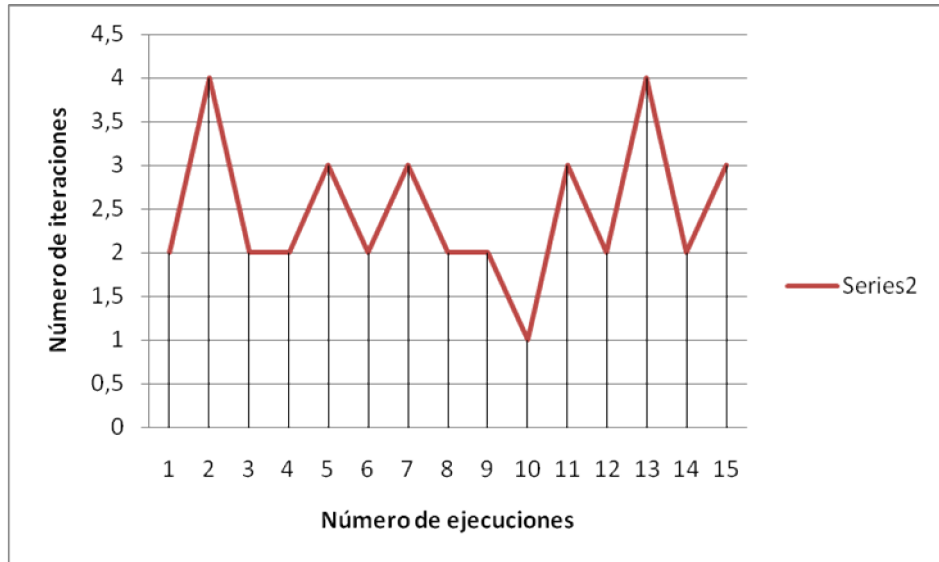
| EJECUCIÓN # 14 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| J | 24.48 | 2 |

Tabla 37. Ejecución número 15 para un segundo criterio de selección.

| EJECUCIÓN # 15 | | |
|--------------------|--|-----------------------|
| SEÑAL SELECCIONADA | POTENCIA DE LA SEÑAL SELECCIONADA (dB) | NÚMERO DE ITERACIONES |
| C | 24.51 | 3 |

La figura 23 ilustra el número de iteraciones ocurridas por cada ejecución que se le hace al programa. Esta gráfica demuestra que el algoritmo genético encuentra la solución adecuada al problema dado al realizar en su gran mayoría dos ejecuciones. Además, cabe decir que de las 15 ejecuciones realizadas, ninguna fue errada o condujo a soluciones erradas, ya que todas llegaron a la solución requerida, la diferencia fue el número de iteraciones y el tiempo gastado por cada ejecución para encontrar la solución. Por ejemplo, la gráfica 23 ilustra que las ejecuciones 2 y 13 gastaron más tiempo y más iteraciones en encontrar la solución, que el tiempo gastado por la ejecución 10, ya que llegó a ella a la primera iteración.

Figura 23. Gráfica del Número de ejecuciones vs. Número de iteraciones.



7.2 OBJETIVO NÚMERO DOS

Diseño de una antena tipo panel a partir del uso de algoritmos genéticos.

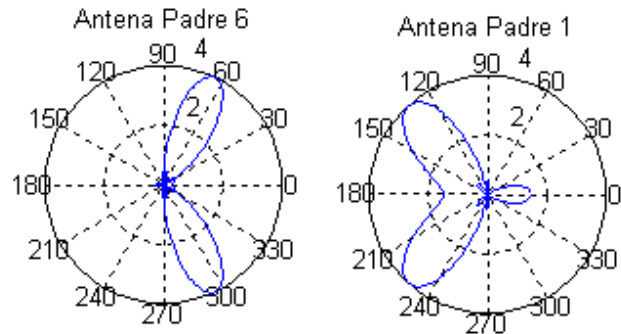
7.2.1 ARREGLO DE ANTENAS.

Un arreglo de antenas es un conjunto de antenas simples, generalmente iguales y orientadas en la misma dirección, las cuales son acomodadas en una disposición física determinada, relativamente cercana unas respecto a otras, y además cada antena es manejada por un mismo sistema de separación (combinador) de señal. Son capaces de concentrar la radiación en direcciones deseadas.

7.2.2 POBLACIÓN INICIAL

Se ilustran tan sólo dos patrones de radiación de los 8 que conforman la población inicial.

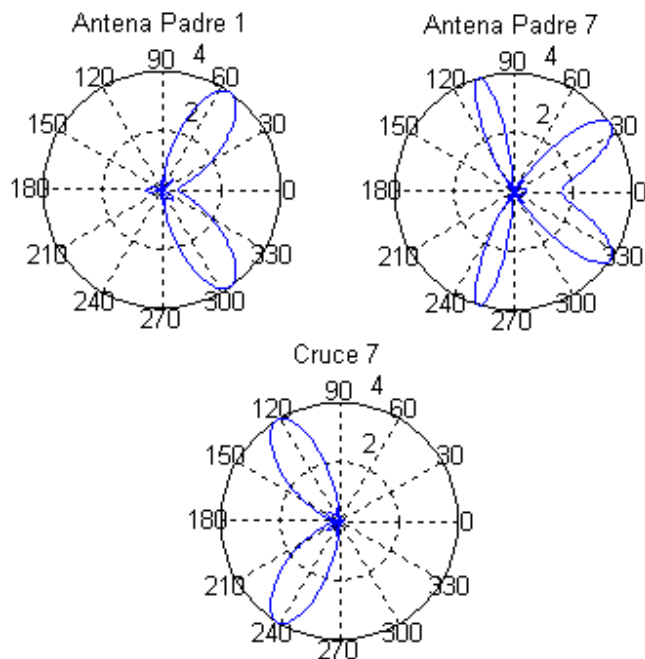
Figura 24. Patrones de radiación de dos antenas padres.



7.2.3 OPERADOR DE CRUCE

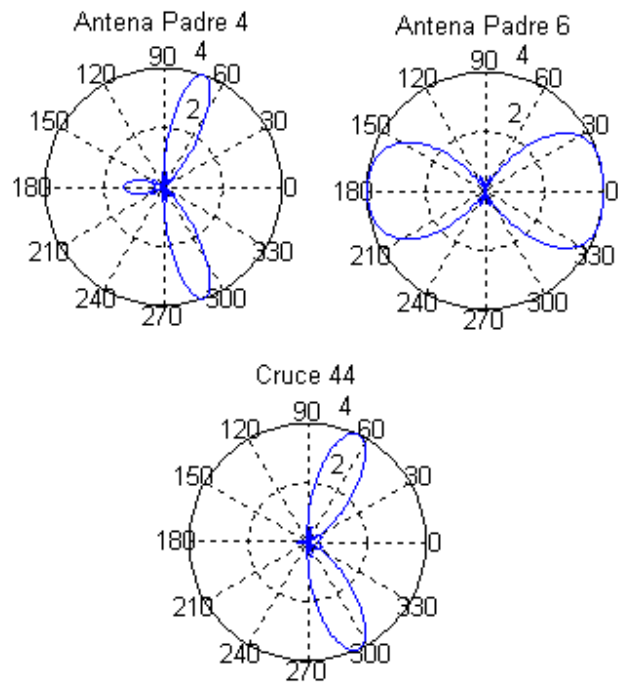
La figura 25 ilustra el cruce realizado entre la Antena Padre 1 con la Antena Padre 7. El patrón resultante de este cruce tiene la misma distancia que la de la Antena Padre 1 pero con la fase de la Antena Padre 7.

Figura 25. Representación de dos antenas padres y su respectivo cruce.



La figura 26 ilustra el cruce realizado entre la Antena Padre 6 con la Antena Padre 4. El patrón resultante de este cruce tiene la misma distancia que la de la Antena Padre 6 pero con la fase de la Antena Padre 4.

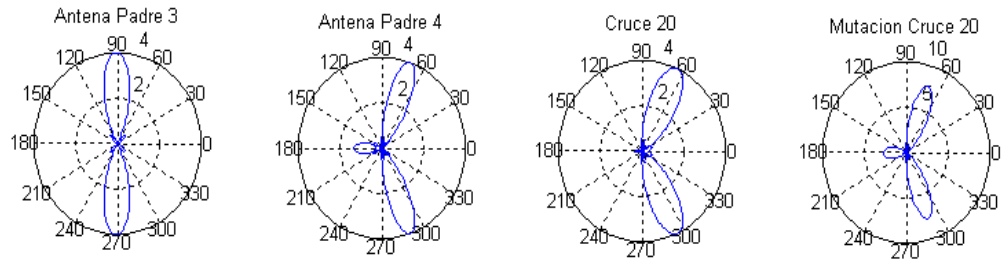
Figura 26. Ejemplo de cruce entre dos antenas.



7.2.4 OPERADOR DE MUTACIÓN

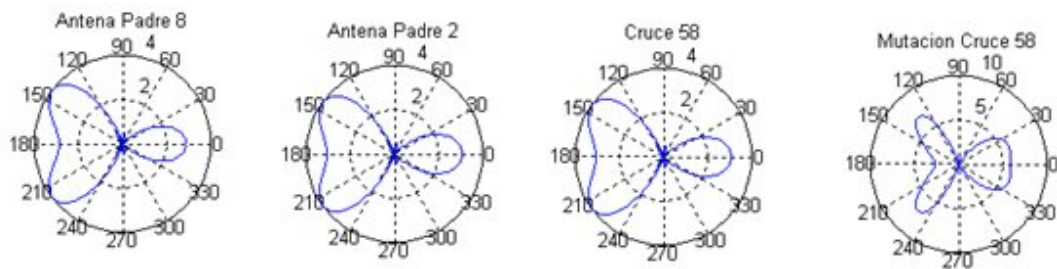
La figura 27 ilustra la mutación realizada al patrón de radiación del cruce 20. Este cruce resulta del intercambio de información entre la Antena Padre 3 y la Antena Padre 4.

Figura 27. Proceso que atraviesa una antena en un algoritmo genético.



La figura 28 ilustra la mutación realizada al patrón de radiación del cruce 58. Este cruce resulta del intercambio de información entre la Antena Padre 8 y la Antena Padre 2.

Figura 28. Gráfica del proceso que hace una antena en un algoritmo genético.

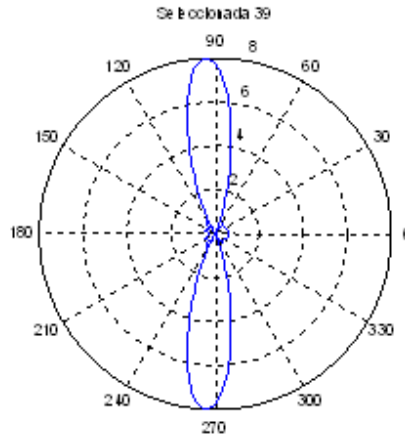


La mutación realizada a los patrones de radiación consiste en adicionar un valor a la distancia entre los elementos del arreglo.

7.2.5 SELECCIÓN

La figura 29 ilustra un patrón de radiación que cumple el criterio de selección definido.

Figura 29. Patrones de radiación de las antenas seleccionadas.



Gracias a la buena definición de los operadores genéticos (cruce, mutación y selección), se encontró el patrón de radiación que cumplía con el criterio que previamente se había definido.

Las tablas 38 a 52 muestran el proceso hecho por el algoritmo genético para el criterio de selección ya mencionado, para llegar a la antena deseada. Por ejemplo, para llegar a la antena de selección 15, mostrada por la tabla 38, las antenas cruzadas fueron la 8 y la 3 respectivamente quienes poseen una ganancia, por cierto, muy parecida la una de la otra tanto en su parte real como en su parte imaginaria. El cruce entre estas dos antenas también denominadas antenas padres produjo la antena llamada cruce 15, quién a su vez tiene también una ganancia con valores reales e imaginarios cercanos a los de las antenas de la primera generación. A este cruce se le realizó una mutación que obtenía la ganancia que iba a ser evaluada con el criterio de selección, ésta ganancia cumplía perfectamente con los criterios de selección definidos, por lo que fue seleccionada como la mejor solución al requerimiento del programa.

Tabla 38. Ejecución número 1 del algoritmo genético para encontrar la antena.

| ANTENA 1 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 1 | PADRE 8 | PADRE 3 | CRUCE 15 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.35 | 13.48 | 15.17 | 26.58 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 39. Ejecución número 2 del algoritmo genético para encontrar la antena.

| ANTENA 2 | | | | |
|------------------------------|----------------|----------------|-----------------|--------------------------------|
| EJECUCIÓN # 2 | PADRE 7 | PADRE 5 | CRUCE 53 | MUTACIÓN Y SELECCIÓN 53 |
| Ganancia en dB | 13.51 | 13.5157 | 15.2288 | 26.2468 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 40. Ejecución número 3 del algoritmo genético para encontrar la antena.

| ANTENA 3 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 3 | PADRE 5 | PADRE 5 | CRUCE 37 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5157 | 13.5157 | 15.2290 | 26.2576 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 41. Ejecución número 4 del algoritmo genético para encontrar la antena.

| ANTENA 4 | | | | |
|------------------------------|----------------|----------------|----------------|-----------------------------|
| EJECUCIÓN # 4 | PADRE 1 | PADRE 7 | CRUCE 7 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5194 | 13.51 | 15.0914 | 26.9886 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 42. Ejecución número 5 del algoritmo genético para encontrar la antena.

| ANTENA 5 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 5 | PADRE 2 | PADRE 4 | CRUCE 12 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5228 | 13.5030 | 15.5438 | 26.9720 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 43. Ejecución número 6 del algoritmo genético para encontrar la antena.

| ANTENA 6 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 6 | PADRE 4 | PADRE 7 | CRUCE 31 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5030 | 13.51 | 15.6114 | 26.0263 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 44. Ejecución número 7 del algoritmo genético para encontrar la antena.

| ANTENA 7 | | | | |
|-----------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 7 | PADRE 7 | PADRE 4 | CRUCE 52 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.51 | 13.5030 | 15.6671 | 26.1320 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 45. Ejecución número 8 del algoritmo genético para encontrar la antena.

| ANTENA 8 | | | | |
|-----------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 8 | PADRE 8 | PADRE 4 | CRUCE 60 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5244 | 13.5030 | 15.3242 | 26.9962 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 46. Ejecución número 9 del algoritmo genético para encontrar la antena.

| ANTENA 9 | | | | |
|-----------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 9 | PADRE 3 | PADRE 6 | CRUCE 22 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5186 | 13.5229 | 15.1196 | 26.0383 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 47. Ejecución número 10 del algoritmo para encontrar la antena.

| ANTENA 10 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 10 | PADRE 6 | PADRE 1 | CRUCE 41 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5229 | 13.5194 | 15.5918 | 26.9830 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 48. Ejecución número 11 del algoritmo para encontrar la antena.

| ANTENA 11 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 11 | PADRE 4 | PADRE 4 | CRUCE 28 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5030 | 13.5030 | 15.2789 | 26.3578 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 49. Ejecución número 12 del algoritmo para encontrar la antena.

| ANTENA 12 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 12 | PADRE 4 | PADRE 1 | CRUCE 25 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5030 | 13.5194 | 15.3926 | 26.5850 |
| NÚMERO DE ITERACIONES | | | | 3 |

Tabla 50. Ejecución número 13 del algoritmo para encontrar la antena.

| ANTENA 13 | | | | |
|---------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 13 | PADRE 2 | PADRE 6 | CRUCE 14 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5228 | 13.5229 | 15.5417 | 26.8835 |
| NÚMERO DE ITERACIONES | | | | 3 |

Tabla 51. Ejecución número 14 del algoritmo para encontrar la antena.

| ANTENA 14 | | | | |
|---------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 14 | PADRE 6 | PADRE 6 | CRUCE 46 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5229 | 13.5229 | 15.3125 | 26.6082 |
| NÚMERO DE ITERACIONES | | | | 4 |

Tabla 52. Ejecución número 15 del algoritmo para encontrar la antena.

| ANTENA 15 | | | | |
|---------------------------------|----------------|----------------|----------------|-----------------------------|
| EJECUCIÓN # 15 | PADRE 1 | PADRE 3 | CRUCE 3 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 13.5194 | 13.5186 | 15.4996 | 26.7989 |
| NÚMERO DE ITERACIONES | | | | 1 |

Todas las antenas seleccionadas cumplen perfectamente con el criterio de selección definido previamente, por lo que el algoritmo genético resulta ser una herramienta de programación eficiente que cumple exitosamente con los requerimientos a los que sea sometido, ya que puede realizar la cantidad de iteraciones necesarias hasta que encuentre la solución.

La figura 30 ilustra el número de iteraciones ocurridas por cada ejecución que se le hace al programa. Esta gráfica demuestra que el algoritmo genético cuenta con una gran probabilidad de éxito con tan sólo una ejecución realizada, puesto que de las 15 ejecuciones hechas, 8 tuvieron éxito con tan sólo una iteración. Además, cabe decir que de las 15 ejecuciones realizadas, ninguna tuvo una mala respuesta, ya que todas las ejecuciones llegaron a la solución requerida, la diferencia fue el número de iteraciones y el tiempo gastado por cada ejecución para encontrar la solución. Por ejemplo, la gráfica 30 ilustra que la ejecución 14 gastó más tiempo y más iteraciones en encontrar la solución, ya que llegó a ella a la cuarta iteración.

Figura 30. Número de iteraciones vs. Número de ejecuciones.



Las tablas 53 a la 67 muestran el proceso hecho por el algoritmo genético, para un segundo criterio de solución expuesto el cual se basa en encontrar una antena con una ganancia (en dB) que esté entre los 23 y 24 dB. Por ejemplo, para llegar a la antena de selección 17, mostrada por la tabla 53, las antenas cruzadas fueron la 3 y la 1 respectivamente quienes poseen una ganancia, por cierta muy parecida. El cruce entre estas dos antenas también denominadas antenas padres produjo la antena llamada cruce 15, quién a su vez tiene también una ganancia con valores cercanos a los de las antenas de la primera generación. A este cruce se le realizó

una mutación que obtenía la ganancia que iba a ser evaluada con el criterio de selección, ésta ganancia cumplía perfectamente con los criterios de selección definidos, por lo que fue seleccionada como la mejor solución al requerimiento del programa.

Tabla 53. Ejecución número 1 para un segundo criterio de selección.

| ANTENA 1 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 1 | PADRE 3 | PADRE 1 | CRUCE 17 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.3314 | 12.3142 | 15.3145 | 23.0714 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 54. Ejecución número 2 para un segundo criterio de selección.

| ANTENA 2 | | | | |
|------------------------------|----------------|----------------|----------------|-----------------------------|
| EJECUCIÓN # 2 | PADRE 1 | PADRE 4 | CRUCE 4 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.3781 | 12.6216 | 15.8426 | 23.1253 |
| NÚMERO DE ITERACIONES | | | | 4 |

Tabla 55. Ejecución número 3 para un segundo criterio de selección.

| ANTENA 3 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 3 | PADRE 4 | PADRE 4 | CRUCE 28 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.6874 | 12.4128 | 15.1742 | 23.7492 |
| NÚMERO DE ITERACIONES | | | | 3 |

Tabla 56. Ejecución número 4 para un segundo criterio de selección.

| ANTENA 4 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 4 | PADRE 5 | PADRE 5 | CRUCE 37 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.0785 | 12.6120 | 15.9126 | 23.0102 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 57. Ejecución número 5 para un segundo criterio de selección.

| ANTENA 5 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 5 | PADRE 5 | PADRE 5 | CRUCE 37 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.6791 | 12.3158 | 15.1260 | 23.0284 |
| NÚMERO DE ITERACIONES | | | | 4 |

Tabla 58. Ejecución número 6 para un segundo criterio de selección.

| ANTENA 6 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 6 | PADRE 5 | PADRE 3 | CRUCE 35 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.1867 | 12.9217 | 15.1928 | 23.8326 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 59. Ejecución número 7 para un segundo criterio de selección.

| ANTENA 7 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 7 | PADRE 8 | PADRE 3 | CRUCE 59 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.0183 | 12.0684 | 15.6317 | 23.8215 |
| NÚMERO DE ITERACIONES | | | | 4 |

Tabla 60. Ejecución número 8 para un segundo criterio de selección.

| ANTENA 8 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 8 | PADRE 2 | PADRE 5 | CRUCE 13 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.7315 | 12.8517 | 15.9478 | 23.8124 |
| NÚMERO DE ITERACIONES | | | | 3 |

Tabla 61. Ejecución número 9 para un segundo criterio de selección.

| ANTENA 9 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 9 | PADRE 5 | PADRE 6 | CRUCE 38 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.6809 | 12.3891 | 15.5512 | 23.6689 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 62. Ejecución número 10 para un segundo criterio de selección.

| ANTENA 10 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 10 | PADRE 8 | PADRE 6 | CRUCE 62 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.1368 | 12.7591 | 15.1053 | 23.6289 |
| NÚMERO DE ITERACIONES | | | | 4 |

Tabla 63. Ejecución número 11 para un segundo criterio de selección.

| ANTENA 11 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 11 | PADRE 3 | PADRE 8 | CRUCE 24 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.6478 | 12.0679 | 15.9310 | 23.3321 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 64. Ejecución número 12 para un segundo criterio de selección.

| ANTENA 12 | | | | |
|------------------------------|----------------|----------------|----------------|-----------------------------|
| EJECUCIÓN # 12 | PADRE 1 | PADRE 3 | CRUCE 3 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.6982 | 12.6691 | 15.0485 | 23.9215 |
| NÚMERO DE ITERACIONES | | | | 1 |

Tabla 65. Ejecución número 13 para un segundo criterio de selección.

| ANTENA 13 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 13 | PADRE 3 | PADRE 1 | CRUCE 17 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.8919 | 12.6215 | 15.9447 | 23.9217 |
| NÚMERO DE ITERACIONES | | | | 2 |

Tabla 66. Ejecución número 14 para un segundo criterio de selección.

| ANTENA 14 | | | | |
|------------------------------|----------------|----------------|-----------------|-----------------------------|
| EJECUCIÓN # 14 | PADRE 3 | PADRE 6 | CRUCE 22 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.9317 | 12.3368 | 15.9845 | 23.9974 |
| NÚMERO DE ITERACIONES | | | | 3 |

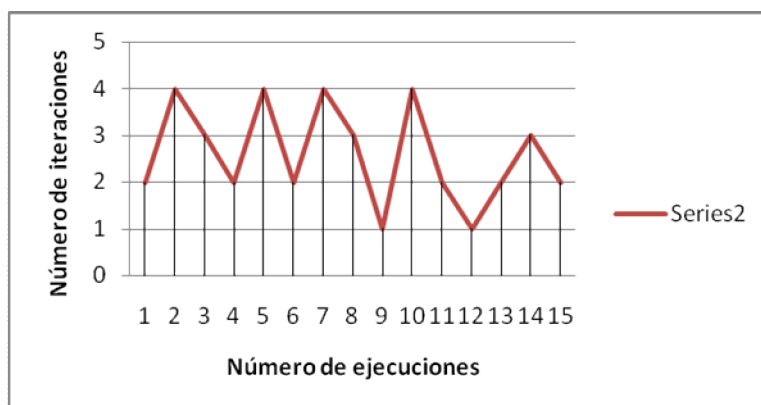
Tabla 67. Ejecución número 15 para un segundo criterio de selección.

| ANTENA 15 | | | | |
|-----------------------|---------|---------|----------|----------------------|
| EJECUCIÓN # 15 | PADRE 7 | PADRE 7 | CRUCE 55 | MUTACIÓN Y SELECCIÓN |
| Ganancia en dB | 12.8567 | 12.9811 | 15.0178 | 23.0908 |
| NÚMERO DE ITERACIONES | | | | 2 |

Todas las antenas seleccionadas cumplen perfectamente con el segundo criterio de prueba de selección definido previamente, por lo que el algoritmo genético resulta ser una herramienta de programación eficiente que cumple exitosamente con los requerimientos a los que sea sometido, ya que puede realizar la cantidad de iteraciones necesarias hasta que encuentre la solución.

La figura 31 ilustra el número de iteraciones ocurridas por cada ejecución que se le hace al programa. Es importante decir que de las 30 ejecuciones realizadas, ninguna tuvo una mala respuesta, ya que todas las ejecuciones llegaron a la solución requerida, la diferencia fue el número de iteraciones y el tiempo gastado por cada ejecución para encontrar la solución. Por ejemplo, la gráfica 31 ilustra que las ejecuciones 2, 5, 7 y 10 gastaron más tiempo y más iteraciones en encontrar la solución, ya que llegaron a ella a la cuarta iteración.

Figura 31. Número de iteraciones vs. Número de ejecuciones.



8. CONCLUSIONES

Tras realizar el análisis comparativo entre la programación habitual y los algoritmos genéticos, se identificaron correctamente las ventajas y desventajas que ofrece la utilización de los algoritmos genéticos, entre ellas, la capacidad de trabajo en forma paralela, la utilización de variables continuas y discretas, la opción de resolver un problema sin poseer un conocimiento específico sobre un tema, la posibilidad de brindar no solo una única solución a un problema, sino un listado de parámetros óptimos para resolverlo, entre otros.

Es necesaria la utilización de un lenguaje de programación robusto que sea tolerable a los continuos cambios aleatorios que ejecutan cada una de las instrucciones debido a los operadores genéticos aplicados con el fin de que no se produzcan errores fatales constantemente y también para evitar resultados sin sentido.

Los problemas denominados como “no lineales”, es decir, aquellos donde cambios múltiples que, individualmente, son perjudiciales, en combinación pueden conducir hacia una solución óptima en todo el sistema, son los ideales para abordar por medio de algoritmos genéticos, gracias a la facultad de paralelismo implícito que estos poseen para encontrar un enorme número de posibilidades.

Una buena definición de los operadores genéticos permite la ejecución óptima de del programa realizado. De no ser así, se deben esperar resultados que no demuestran el verdadero potencial que puede llegar a tener la solución de problemas utilizando algoritmos genéticos.

No se debe utilizar algoritmos genéticos para la solución de problemas que pueden resolverse de manera analítica, debido a que los métodos analíticos tradicionales consumen mucho menos tiempo y potencia computacional que los algoritmos genéticos.

En el caso de la utilización de algoritmos genéticos para encontrar un parámetro de optimización de un arreglo de antenas, debido al trabajo de los mismos por medio de métodos heurísticos, es necesaria la intervención humana para afinar algunos parámetros que están involucrados con ellas, sin embargo los resultados obtenidos fueron muy prometedores y es por eso se invita a seguir un estudio posterior para así desarrollar un sinnúmero de aplicaciones basadas en algoritmos genéticos.

La organización del tutorial Web facilita, en gran proporción, la fácil comprensión de los conceptos que requiere el estudio básico necesario para trabajar con algoritmos genéticos.

9. RECOMENDACIONES

Para un posterior estudio basado en este documento se recomienda profundizar en temas tales como:

- Utilizar el tutorial Web implementado para comprender con facilidad la diferente terminología que implica el estudio de algoritmos genéticos.
- Definición de diferentes criterios para los operadores genéticos de cruce, mutación y selección.
- Optimización del código escrito en Matlab para la reducción del tiempo de máquina gastado en la ejecución de cada una de las aplicaciones y así mismo facilitar la implementación de posteriores estudios.
- Definir nuevos criterios de selección para las aplicaciones propuestas en este documento.
- Profundizar el tema de algoritmos genéticos enfocado al procesamiento digital de señales para realizar posteriores aplicaciones.

GLOSARIO

ALELO: Forma alternativa de un gen, por ejemplo, un hipotético gen c podría presentar tres variantes en una población, los alelos c, c y c1. Cada alelo representa una secuencia de ADN, siendo ligeramente diferentes unas de otras. La combinación de los alelos lo que determina el fenotipo.

COMPETICIÓN: Interacción que tiene lugar entre dos o más organismos, poblaciones o especies que comparten algo, cuando algo escasea. La competición es un importante motor de evolución; las plantas, por ejemplo, se tornan más altas para competir por la luz, y los animales desarrollan mejores sistemas para conseguir alimento (Complutense Oxford, 2004).

CRÍA: Voz que se utiliza para referirse a la selección y desarrollo de algunas variedades de individuos con características especiales (Complutense Oxford, 2004).

CROMOSOMA: Estructura que se compone de ADN asociado con proteínas y que lleva la información genética.

DESCENDIENTES: Evolución de grupos diferentes en una dirección similar; (2) adquisición independiente de rasgos parecidos en dos especies emparentadas.

FENOTIPO: Características que pueden verse en un organismo. Están determinadas por sus genes, a través de las relaciones de dominancia entre los alelos, así como por la interacción de los genes con el medio ambiente en que se encuentra el organismo. (Complutense Oxford, 2004).

GEN: La unidad básica de la herencia, por la que los caracteres hereditarios se transmiten de padres a hijos.

GENÉTICA: La parte de la biología que estudia la variación heredable y su base física, el ADN o material hereditario; (2) de un organismo, la base física de sus caracteres hereditarios, esto es, la secuencia y ordenamiento de sus genes (Complutense Oxford, 2004).

GENOTIPO: Composición genética de un organismo, es decir, la combinación de alelos que posee. La constitución genética exacta de un organismo (Complutense Oxford, 2004).

INDIVIDUO: Cada ser organizado, sea animal o vegetal, respecto de la especie a que pertenece (española)

MUTACIÓN: La mutación es una alteración o cambio en la información genética de un ser vivo y que, por lo tanto, va a producir un cambio de características, que se presenta súbita y espontáneamente, y que se puede transmitir o heredar a la descendencia (Complutense Oxford, 2004).

OPTIMIZAR: Buscar la mejor manera de realizar una actividad (española)

REPLICACIÓN: Situación en la que un cromosoma, segmento cromosómico o gen se encuentran en un número superior al número normal de copias.

SELECCIÓN: Proceso por el que uno o más de los factores que actúan sobre una población provocan una mortalidad diferencial, de forma que se favorece la transmisión de ciertos rasgos. en principio, más provechosos, a las siguientes generaciones (Complutense Oxford, 2004).

BIBLIOGRAFÍA

- Alba, E. Análisis y Diseño de Algoritmos Genéticos Paralelos Distribuidos. Retrieved 15 de Marzo, 2009, from <http://neo.lcc.uma.es/tesis/PhD-Alba99.pdf>
- Annicchiarico, W. (1996). *Algoritmos de optimización estructural basados en simulación genética* (Illustrated, CDCH UCV ed.).
- Balanis, C. A. (2005). *Antenna Theory, Analysis and Design* (Tercera ed.): Wiley-Interscience
- BUTZ, M. V. (2002). *Anticipatory Learning Classifier Systems* (Vol. 1 Edición). Comlutense Oxford, O. U. (2004). *Diccionario de biología* (illustrated ed.). española, R. a. Diccionario de la lengua española. Vigésima segunda. Retrieved 21 de marzo, 2009, from http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=INDIVIDUO
- Etienne, J. (2001). *Bioquímica genética, biología molecular*.
- Ferreira, C. (2006). *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence* (2 ed.).
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machina*.
- Haupt, R. L. (2007). *Genetic Algorithms in Electromagnetics*.
- Kuri, Á. (2000). *Centro de Investigación en Computación*: Instituto Politécnico Nacional.
- Louden, K. (2004). *Lenguajes de Programación, Principios y Práctica*. (Segunda ed.).
- Mitchel, M. (1998). *An Introduction to Genetic Algorithms (Complex Adaptive Systems)* (2nda ed.).
- Sánchez, M. d. C. (2006). *La evolución*.
- Sumathi, S. H., T.; Surekha, P. (2008). *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*. (illustrated ed.): Springer.

ANEXOS

ANEXO A.

El código en Matlab implementado para diseñar el generador de señales por medio de algoritmos genéticos, se encuentra en la página de internet www.tutorialalgoritmosgeneticos.co.cc en la sección de Descargas.

ANEXO B.

El código en Matlab implementado para diseñar la antena tipo panel a través de algoritmos genéticos, se encuentra en la página de internet www.tutorialalgoritmosgeneticos.co.cc en la sección de Descargas.