UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

LICENCIATURA EN INGENIERÍA EN SISTEMAS INTELIGENTES

MANUAL PARA PRÁCTICAS DEL LABORATORIO DE CÓMPUTO

ASIGNATURA:

ALGORITMOS GENÉTICOS

ELABORARÓN:

DRA. DORICELA GUTIÉRREZ CRUZ
M. en C. YAROSLAF AARÓN ALBARRÁN FERNÁNDEZ
DRA. CARMEN LILIANA RODRÍGUEZ PÁEZ

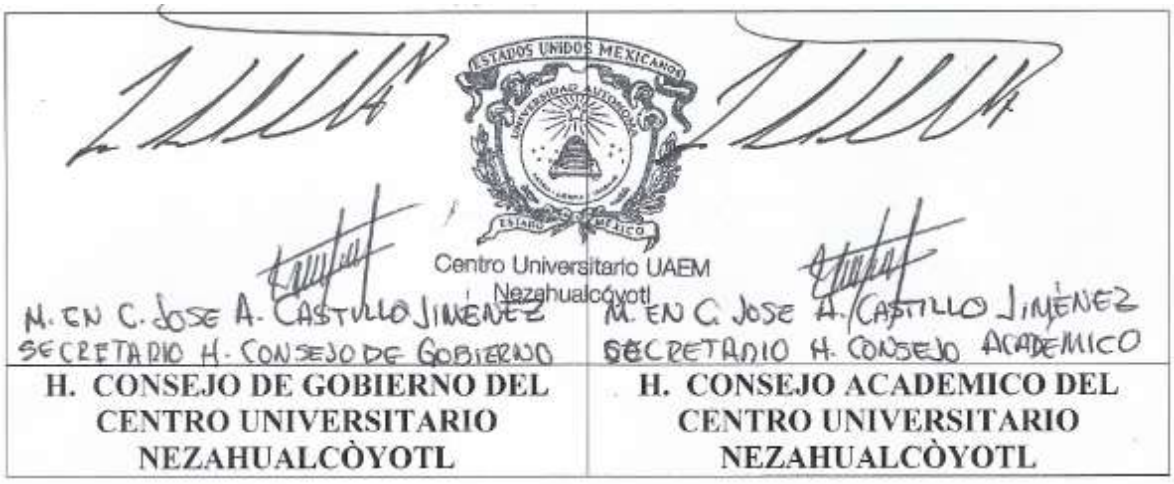
AGOSTO 2019

MANUAL PARA PRÁCTICAS DEL LABORATORIO DE CÓMPUTO PARA LA ASIGNATURA DE ALGORITMOS GENÉTICOS

IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

<table>
<tr><td colspan="8">ESPACIO ACADÉMICO: Centro Universitario Nezahualcóyotl</td></tr>
<tr><td colspan="5">PROGRAMA EDUCATIVO: LICENCIATURA EN INGENIERÍA EN SISTEMAS INTELIGENTES</td><td colspan="3">Área de docencia: HERRAMIENTA PARA LOS SISTEMAS INTELIGENTES</td></tr>
<tr><td colspan="3">Aprobación por los H.H. Consejos Académico y de Gobierno</td><td colspan="2">Fecha:
Julio 2019</td><td colspan="3">Programa elaborado por: Doricela Gutiérrez Cruz, Yaroslaf Aarón Albarrán Fernández</td></tr>
<tr><td colspan="5">Nombre de la Unidad de Aprendizaje: ALGORITMOS GENÉTICOS</td><td colspan="3">Fecha de elaboración:AGOSTO 2019</td></tr>
<tr><td>Clave
L41657</td><td>Horas de teoría
3</td><td>Horas de práctica
2</td><td>Total, de horas
5</td><td>Créditos
8</td><td>Tipo de Unidad de Aprendizaje
Curso</td><td>Carácter de la Unidad de Aprendizaje
Obligatoria</td><td>Núcleo de formación
Integral</td></tr>
<tr><td colspan="2">Prerrequisitos
Programación.</td><td colspan="3">Unidad de Aprendizaje Antecedente
Ninguna</td><td colspan="3">Unidad de Aprendizaje Consecuente
Ninguna</td></tr>
</table>

EL PRESENTE MANUAL DE PRÁCTICAS HA SIDO AVALADO EN EL MES DE AGOSTO DE 2019 POR:



H. CONSEJO DE GOBIERNO DEL CENTRO UNIVERSITARIO NEZAHUALCÒYOTL	H. CONSEJO ACADEMICO DEL CENTRO UNIVERSITARIO NEZAHUALCÒYOTL

ÍNDICE

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

DIRECTORIO INSTITUCIONAL

Dr. en Edu. **Alfredo Barrera Baca**

RECTOR

M. en E.U. y R.
Marco Antonio Luma Pichardo
Secretario de Docencia

Dr. en C.I.
Carlos Eduardo Barrera Díaz
Secretario de Investigación y Estudios Avanzados

M. en C
Jannet Valero Vilchis
Secretaria de Rectoría

Dr. en A.
José Édgar Miranda Ortiz
Secretario de Difusión Cultural

Dra. en Ed.
Sandra Chávez Marín
Secretaria de Extensión y Vinculación

M. en E.
Javier González Martínez
Secretario de Finanzas

M. en Dis.
Juan Miguel Reyes Viurquez
Secretario de Administración

Dr. en C.C.
José Raymundo Marcial Romero
Secretario de Planeación y DESARROLLO Institucional

M. en L.A.
María del Pilar Ampudia García
Secretaria de Cooperación Internacional

Dra. en Dis.
Mónica Marina Mondragón
Secretaría de Cultura Física y Deporte

Dr. en C.S.
Luis Raúl Ortiz Ramírez
Abogado General

M. en R. I.
Jorge Bernáldez García
Secretario Técnico de la Rectoría

Lic. en Com.
Gastón Pedraza Muñoz
Director General de Comunicación Universitaria

M. en A. P.
Guadalupe Ofelia Santamaría González
Directora General de Centros Universitarios y Unidades Académicas Profesionales

M. en D.F.
Jorge Rogelio Zenteno Domínguez
Encargado del Despacho de la Contraloría Universitaria

Lic. En Act.
Angelita Garduño Gómez
Secretaria particular del Rector

M. en A.
José Francisco Mejía Carbajal
Secretario Particular Adjunto del Rector

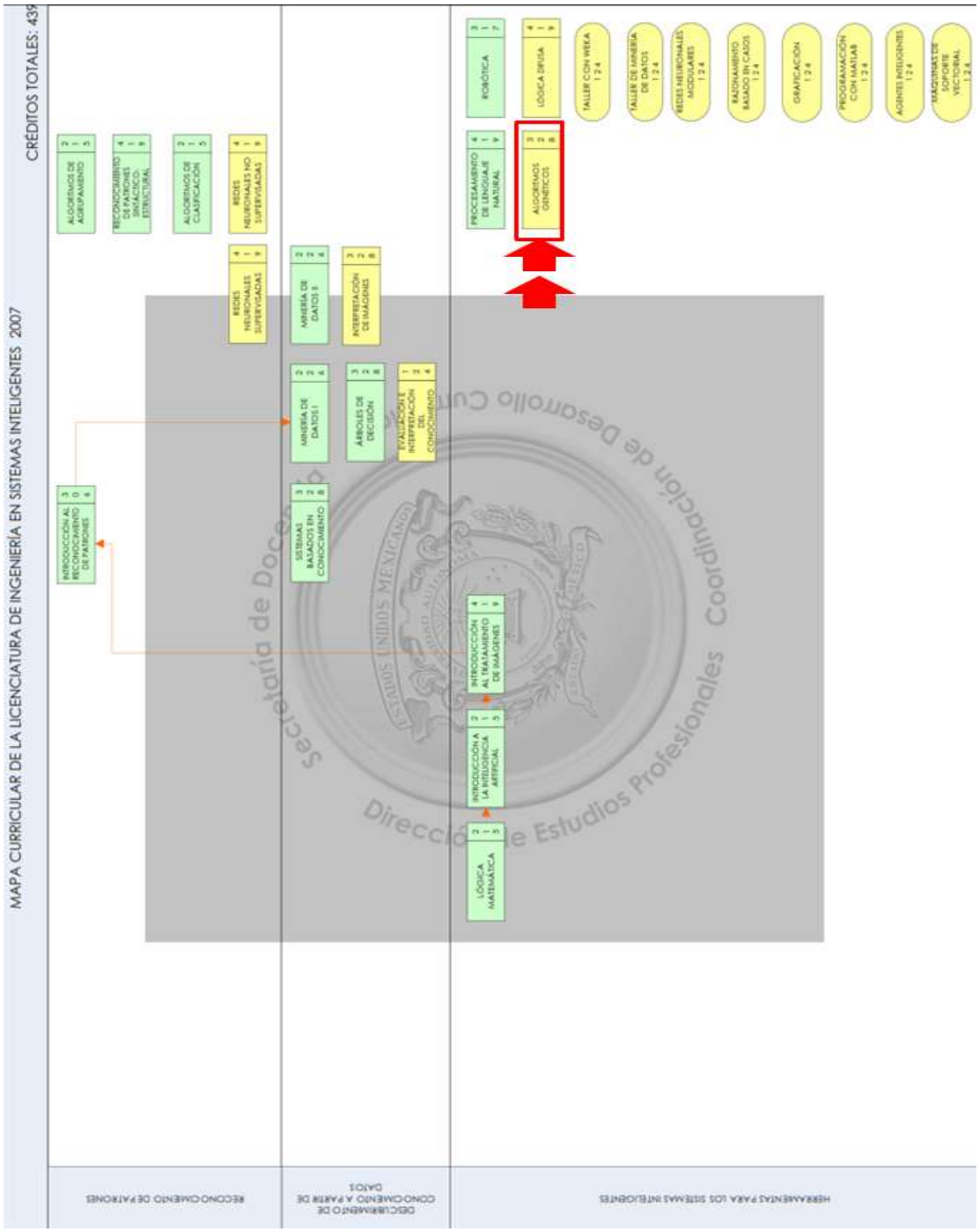
CENTRO UNIVERSITARIO UAEM NEZAHUALCÓYOTL

DIRECTORIO

Maestro en Derecho **Juan Carlos Medina Huicochea**	**ENCARGADO DEL DESPACHO DE LA DIRECCIÓN**
Maestro en Ciencias **José Antonio Castillo Jiménez**	Subdirector Académico
Licenciado en Economía **Ramón Vital Hernández**	Subdirector Administrativo
Doctora en Ciencias Sociales **María Luisa Quintero Soto**	Coordinadora de Investigación y Estudios Avanzados
Licenciado en Administración de Empresas **Víctor Manuel Durán López**	Coordinador de Planeación y DESARROLLO Institucional
Maestro en Ciencias **Cesar Lucio Gutiérrez Ruiz**	Coordinador de la Licenciatura en Comercio Internacional
Maestro en S.F. **Carlos Anaya Hernández**	Coordinador de la Licenciatura en Educación para la Salud
Doctor en Ingeniería de Sistemas **Ricardo Rico Molina**	Coordinador de la Licenciatura en Ingeniería en Sistemas Inteligentes
Maestro en Ciencias **Ricardo Pacheco Ruiz**	Coordinador de la Licenciatura en Ingeniería en Transporte
Maestro en Ciencias de la Computación **Erick Nicolás Cabrera Álvarez**	Coordinador de la Licenciatura en Seguridad Ciudadana Mixta
Maestro en Administración **José Ramon CS. Garcia Ibarra**	Coordinador de la Licenciatura en Seguridad Ciudadana Presencial

Ubicación de la asignatura de Algoritmos Genéticos, dentro del programa de la Lic. en Ing.

en Sistemas Inteligentes.



SECUENCIA DIDÁCTICA

PRÁCTICA 1
INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS Y BASES EVOLUTIVAS

PRÁCTICA 2
CODIFICACIÓN DE LA POBLACIÓN (Decimal - Binario)

PRÁCTICA 3
Decodificación de la Población (Binario - Real)

PRÁCTICA 4
INICIALIZACIÓN DE LA POBLACIÓN (REAL- ADAPTADO)

PRÁCTICA 5
PARÁMETRO DE PRECISIÓN EN LA LONGITUD DE UN CROMOSOMA

PRÁCTICA 6
OPERADOR SELECCIÓN

PRÁCTICA 7
OPERADOR DE CRUCE DE UN PUNTO

PRÁCTICA 8
OPERADOR CRUCE DE DOS PUNTOS

PRÁCTICA 9
OPERADOR CRUCE UNIFORME

PRÁCTICA 10
OPERADOR MUTACIÓN

PRÁCTICA 11
ALGORITMO GENÉTICO SECUENCIAL

PRÁCTICA 1

INTRODUCCIÓNA LOS ALGORITMOS GENÉTICOS Y BASES EVOLUTIVAS

OBJETIVO

- Introducir al alumno en conceptos básicos y terminología utilizada en algoritmos genéticos

INTRODUCCIÓN

La herencia genética consiste en la transmisión del material genético de generación en generación a través de los gametos (espermatozoides u óvulos). La mitad de nuestro material genético lo heredamos de nuestro padre y la otra mitad de nuestra madre. El material genético se organiza en genes, que contienen información para una determinada característica o función. De este modo, tenemos dos copias o alelos de cada gen, una procedente del padre y otra de la madre. La información contenida en los genes puede tener errores que alteran la función del gen denominados mutaciones. Estas mutaciones pueden transmitirse de padres a hijos y desencadenar, en determinadas condiciones, enfermedades genéticas.

Mediante la teoría del origen común, Darwin logró integrar evidencias procedentes de campos tan dispares como la bio-geografía, la paleontología, la anatomía o la embriología. La convergencia de todas estas evidencias demostraba la comunidad de descendencia de todos los organismos vivos y extintos. De este modo, Darwin ofrecía una demostración sistemática de la evolución de las especies, generadas todas a partir de un tronco común, cuya diversidad era fruto de cambios graduales (el gradualismo) y no de catástrofes o de inmovilismos, y también de la selección natural, es decir, de la mejor adaptación al medio.

Una situación análoga se puede dar con el mundo empresarial y económico hoy en día. Podríamos elaborar una teoría en la que el medioambiente estuviera compuesto de empresas de diferentes áreas (individuos de diferentes especies), cuya supervivencia dependiera de la adaptación al medio de cada una de ellas, y que su permanencia en el medio (el mercado) depende más de cambios graduales que no de catástrofes puntuales. Salvando el hecho de que todas las empresas-especies derivan de un tronco común, que también podría ser postulado, nos queda por averiguar con qué mecanismos cuentan dichas empresas para la adaptación a los cambios en el mercado y la búsqueda de la certidumbre. (Consultado en ISDI, 2019)

El origen de las especies expone sus ideas acerca de la evolución y la selección natural. Estas ideas se basaron en gran medida en las observaciones directas que Darwin realizó en sus viajes alrededor del mundo. De 1831 a 1836 fue parte de una expedición de investigación realizada a bordo del barco HMS Beagle, la cual hizo paradas en Sudamérica, Australia y la punta sur de África. En cada parada, Darwin tuvo la oportunidad de estudiar y catalogar las plantas y los animales de la localidad.

En el transcurso de sus viajes, Darwin empezó a observar patrones interesantes en la distribución y las características de los organismos.

El encontró que las islas cercanas en las Galápagos tenían especies similares, pero no idénticas, de pinzones. Más aún, notó que cada especie de pinzón era adecuada a su entorno y su función en este. Por ejemplo, las especies que comían semillas grandes tenían picos grandes y duros, mientras que las que consumían insectos presentaban picos delgados y puntiagudos. Finalmente, observó que los pinzones (y otros animales) de las islas Galápagos eran parecidos a las especies que se encontraban en la parte continental de Ecuador, pero distintas de las del resto del mundo. Lo cual, propuso que las especies cambian con el tiempo, es decir; que las especies nuevas provienen de especies preexistentes y que todas las especies comparten un ancestro común.

Darwin se refirió a este proceso, en el que los grupos de organismos cambian en sus características heredables a lo largo de generaciones, como "descendencia con modificaciones". Hoy en día, lo llamamos evolución.

El concepto de selección natural está basado en varias observaciones fundamentales de acuerdo con Darwin:

- *-Los rasgos a menudo son heredables. En los seres vivos, muchas características son hereditarias o pasan de padres a hijos. (Darwin sabía que esto sucedía, si bien no sabía que los rasgos se heredaban mediante genes).

- *-Se produce más descendencia de la que puede sobrevivir. Los organismos son capaces de generar más descendientes de los que su medio ambiente puede soportar, por lo que existe una competencia por los recursos limitados en cada generación.

- *-La descendencia varía en sus rasgos heredables. La descendencia en cualquier generación tendrá rasgos ligeramente distintos entre sí (color, tamaño, forma, etcétera), y muchas de estas características serán heredables.

Basado en estas sencillas observaciones, Darwin concluyó lo siguiente:

- *-En una población, algunos individuos tendrán rasgos heredables que les ayudarán a sobrevivir y reproducirse (dadas las condiciones del entorno, como los depredadores y las fuentes de alimentos existentes). Los individuos con los rasgos ventajosos dejarán más descendencia en la siguiente generación que sus pares, dado que sus rasgos los hacen más efectivos para la supervivencia y la reproducción.

- *-Debido a que los rasgos ventajosos son heredables y a que los organismos que los portan dejan más descendientes, los rasgos tenderán a volverse más comunes (presentarse en una mayor parte de la población) en la siguiente generación.

- *-En el transcurso de varias generaciones, la población se adaptará a su entorno (ya que los individuos con rasgos ventajosos en ese ambiente tendrán consistentemente un mayor éxito reproductivo que sus pares). (Consultado en Khan Academy, 2019)

Fue precisamente Weismann el primero en plantear esa gran tergiversación a la que se refería Darwin. A este respecto, Gould puso el acento en el concepto de allmacht con el que Weismann concibe la selección natural, un verdadero caso de autosuficiencia científica y causalidad omnipotente: el panseleccionismo (158). Además, para desconsuelo de Darwin, esa gran tergiversación iniciada por Weismann sí perduró mucho tiempo, hasta la misma actualidad.

Hoy los neodarwinistas no hablan de otra cosa que no sea la selección natural. Huxley lo expone de la siguiente forma: "La contribución especial de Darwin al problema de la evolución fue la teoría de la selección natural, pero, a causa del rudimentario estado de los conocimientos en ciertos campos biológicos, Darwin se vio obligado a reforzar su exposición con hipótesis subsidiarias acerca de la herencia, de los efectos del uso y desuso y de modificaciones producidas de modo directo por el ambiente. (Consultado en CEPRID, 2019)

Desde que Darwin formuló su teoría en 1859, se han acumulado muchas evidencias que la sustentan. Pero sólo a comienzo del siglo XX, con el DESARROLLO de la genética y el redescubrimiento de los experimentos del botánico austríaco Gregor Mendel (1822-1884), en donde se pudieron explicar cómo funciona el mecanismo de la herencia: de qué manera se transmiten las características que pasan de una generación a otra; por qué las mismas pueden desaparecer para luego reaparecer en generaciones posteriores, y cómo se originan las variaciones sobre las cuales actúa la selección natural. La combinación de la teoría de Darwin con los principios de la genética de Mendel se conoce como teoría sintética de la evolución. Constituye el fundamento de los trabajos de los biólogos actuales, en sus intentos de explicar los detalles de los mecanismos evolutivos.

Mendel pensaba, que, con el control del tipo de cruzas entre los diferentes individuos, se podría rastrear la herencia de ciertas características durante varias generaciones y, con esto, establecer los principios que explican su herencia o transmisión. A partir de ello eligió deliberadamente características simples con formas claramente perceptibles y no intermedias, por ejemplo, el tipo de la semilla era liso o rugoso, la planta tenía un tallo alto o enano, etc. Haciendo estas cruzas durante varias generaciones, lo cual le permitió y pudo explicar la forma de transmisión de los caracteres.

Sus investigaciones sobre estos patrones de la herencia en las plantas de jardín y lo llevaron a suponer la idea de la herencia de partes. Lo cual significaba, que al estudiar ciertas características como el color de la flor el tamaño del tallo, el tipo de semilla o la forma y textura de ésta, las contribuciones paternas (del padre y de la madre) se expresaban con desigualdad. Si estos rasgos o características de cada planta se heredan como elementos o partes, entonces cada planta recibe un elemento de cada progenitor, uno del padre y otro de la madre. Esta herencia de partes significa que cada progenitor contribuye con un elemento, y por lo tanto que la cría tiene pares de elementos. A estos elementos Mendel los llamó caracteres diferenciantes porque, precisamente, diferenciaban a las plantas entre sí. (Consultado en La Genética, 2019)

DESARROLLO

El alumno desarrollara una red semántica, considerando como ejemplo la que a continuación se muestra la red semántica de acuerdo con la información obtenida a través de la investigación obtenido de los siguientes temas:

*-Herencia genética.
*-Lucha por la supervivencia.
*-Evolución de Darwin.
*-Selección de Weismann.
*-Genética de Mendel.

Para ello se empezó a diseñar dicha red semántica partiendo de la herencia genética. (Ver Figura 1)

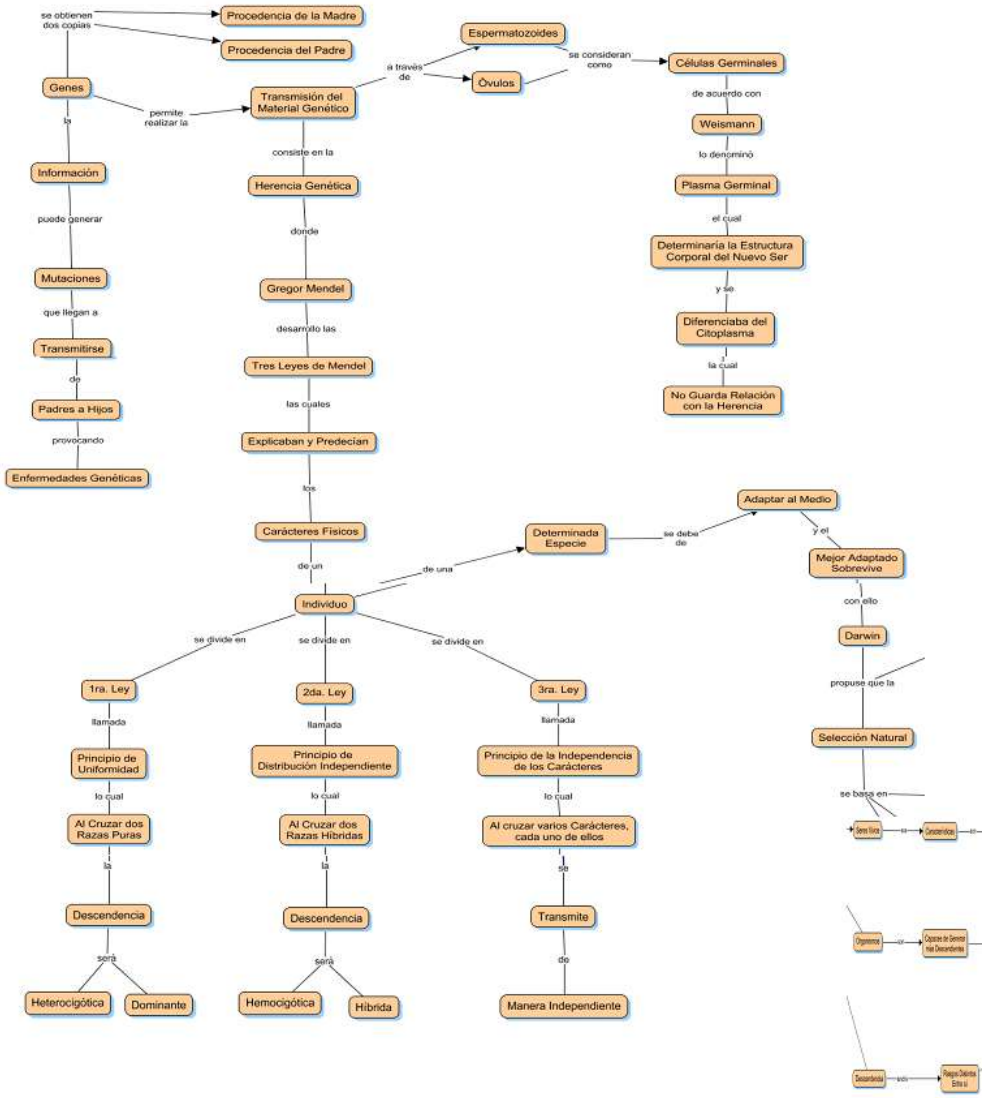


Figura 1. Ejemplo de red semántica sobre principales conceptos de la evolución.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

_____

_____

_____

_____

_____

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones,
   ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

-

BIBLIOGRAFÍA Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.
3. ISDI. La lucha por la supervivencia y la selección natural I/II. [En línea]. [Fecha de consulta: 07 de Agosto de 2019]. Disponible en: https://www.isdi.education/es/isdigital-now/la-lucha-por-la-supervivencia-y-la-seleccion-natural-iii
4. CEPRID. Lysenko: La teoría materialista de la evolución en la URSS (III). . [En línea]. [Fecha de consulta: 07 de agosto de 2019]. Disponible en: https://www.nodo50.org/ceprid/spip.php?article715
5. Khan Academy. Darwin, evolución y selección natural. [En línea]. [Fecha de consulta: 08 de agosto de 2019]. Disponible en: https://es.khanacademy.org/science/biology/her/evolution-and-natural-selection/a/darwin-evolution-natural-selection
6. S/n. La genética: la ciencia de la herencia. [En línea]. [Fecha de consulta: 08 de agosto de 2019]. Disponible en: http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen3/ciencia3/125/htm/sec_3.htm
7. Barrameda. La teoría de la evolución, ayer y hoy. [En línea]. [Fecha de consulta: 09 de agosto de 2019]. Disponible en: https://www.barrameda.com.ar/biologia/la-teoria-de-la-evolucion.htm

PRÁCTICA 2
CODIFICACIÓN DE LA POBLACIÓN (DECIMAL-BINARIO)

OBJETIVO

- El alumno conocerá la codificación binaria que representa la ristra.
- Desarrollará un programa que emule la codificación binaria.

INTRODUCCIÓN

El origen de las especies expone sus ideas acerca de la evolución y la selección natural. Estas ideas se basaron en gran medida en las observaciones directas que Darwin realizó en sus viajes alrededor del mundo [Pérez, 2009].

Basado en estas sencillas observaciones, Darwin concluyó lo siguiente: *"En una población, algunos individuos tendrán rasgos heredables que les ayudarán a sobrevivir y reproducirse (dadas las condiciones del entorno, como los depredadores y las fuentes de alimentos existentes). Los individuos con los rasgos ventajosos dejarán más descendencia en la siguiente generación que sus pares, dado que sus rasgos los hacen más efectivos para la supervivencia y la reproducción"*

El Algoritmo Genético Simple, también denominado Canónico, se necesita una codificación o representación del problema, que resulte adecuada al mismo. Además, se requiere una función de ajuste o adaptación al problema, la cual asigna un número real a cada posible solución codificada. Durante la ejecución del algoritmo, los padres deben ser seleccionados para la reproducción, a continuación, dichos padres seleccionados se cruzarán generando dos hijos, sobre cada uno de los cuales actuará un operador de mutación. El resultado de la combinación de las anteriores funciones será un conjunto de individuos (posibles soluciones al problema), los cuales en la evolución del Algoritmo Genético formarán parte de la siguiente población.

Codificación

Los individuos (posibles soluciones del problema), pueden representarse como un conjunto de parámetros (que denominaremos genes), los cuales agrupados forman una ristra de valores (a menudo referida como cromosoma). Si bien el alfabeto utilizado para representar los individuos no debe necesariamente estar constituido por el [0,1], buena parte de la teoría en la que se fundamentan los Algoritmos Genéticos utiliza dicho alfabeto.

En términos biológicos, el conjunto de parámetros representando un cromosoma particular se denomina fenotipo. El fenotipo contiene la información requerida para construir un organismo, el cual se refiere como genotipo. Los mismos términos se utilizan en el campo de los Algoritmos Genéticos.

La adaptación al problema de un individuo depende de la evaluación del genotipo. Esta última puede inferirse a partir del fenotipo, es decir puede ser computada a partir del cromosoma, usando la función de evaluación. La función de adaptación debe ser diseñada para cada problema de manera específica. Dado un Cromosoma particular, la función de adaptación le asigna un número real, que se supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

Durante la fase reproductiva se seleccionan los individuos de la población para cruzarse y producir descendientes, que constituirán, una vez mutados, la siguiente generación de individuos. La selección de padres se efectúa al azar usando un procedimiento que favorezca a los individuos mejor adaptados, ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada.

Según dicho esquema, los individuos bien adaptados se escogerán probablemente varias veces por generación, mientras que los pobremente adaptados al problema, no se escogerán más que de vez en cuando.

Codificación de las soluciones

Tipos de representación

Existen distintos tipos de codificaciones de las soluciones: binarias, simbólicas, permutacional con repetición y la real. La codificación depende del problema que se quiere resolver. Según Fang (1994) las normas básicas a tener en cuenta para el diseño de una codificación óptima se pueden resumir en:

1. Cada solución del problema tendrá su correspondiente cadena codificada.
2. Para cada cadena codificada producida mediante los operadores genéticos existirá su correspondiente solución decodificada.
3. Codificación y solución se corresponderán una a una, es decir no habrá redundancia.
4. La codificación deberá permitir heredar de padres a hijos las características inherentes a los primeros.

Codificación binaria

El sistema binario, de este modo, emplea sólo dos dígitos o cifras: el cero (0) y el uno (1). En una cifra binaria, cada dígito tiene distinto valor dependiendo de la posición que ocupe. El valor de cada posición es el de una potencia de base 2, elevada a un exponente igual a la posición del dígito menos uno. Se puede observar que, tal y como ocurría con el sistema decimal, la base de la potencia coincide con la cantidad de dígitos utilizados (2) para representar los números.

De acuerdo con estas reglas, el número binario 1011 tiene un valor que se calcula así:

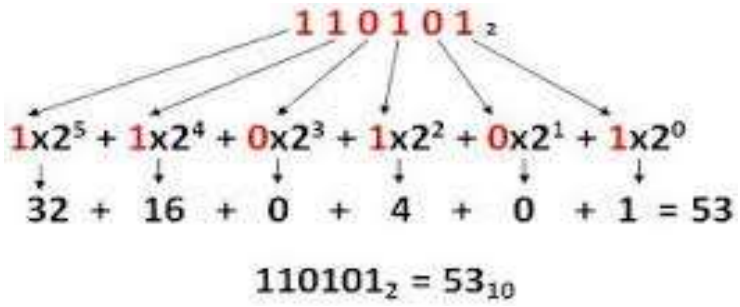
1*23 + 0*22 + 1*21 + 1*20, es decir: 8 + 0 + 2 + 1 = 11

y para expresar que ambas cifras describen la misma cantidad lo escribimos así:

10112 = 1110

Tradicionalmente la codificación binaria ha sido la más utilizada por la facilidad que ofrece el desarrollo de programas informáticos y porque permite la aplicación de sencillos operadores genéticos. Además, la mayor parte de la teoría subyacente en los algoritmos genéticos está desarrollada para la codificación binaria, aunque existen ampliaciones a otros tipos de representaciones

En el AG estándar, los individuos (sus estrategias) se representan mediante una cadena de bits. Entonces, un individuo genético de longitud L consiste en L símbolos 0 y 1. El conjunto completo de todos los posibles individuos genéticos distintos de longitud L será: $S \epsilon \{0,1\}^L$. y su cantidad $|S| \equiv N = 2^L$.



DESARROLLO

El alumno desarrollara un programa que emule la codificación binaria basándose en el diagrama de flujo de la figura 1.

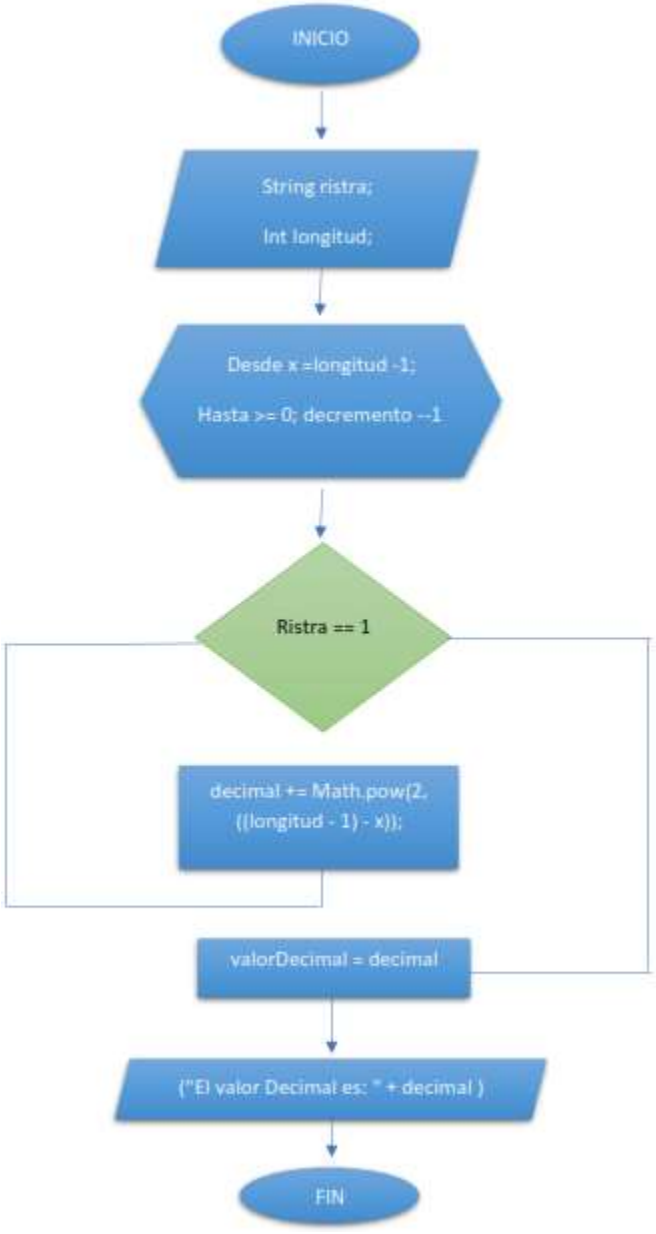


Figura 1. Diagrama de flujo para convertir de decimal a binario

Puede considerar la estructura del programa:

- Código.

```
package decimal.binario;

import java.util.Scanner;

public class DecimalBinario {

  public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int i=0;
        do {
            System.out.print("Introduzca un numero decimal: ");
            i = sc.nextInt();
            System.out.println("El numero en binario es: " +
Integer.toBinaryString(i));
        } while (i==0);
    }
}
```

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, la cual puede ser como la que se muestra a continuación.

Figura 2. Pantalla de ejecución del programa que convierte de decimal a binario

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.
5. Pérez, V. (2009). Cuando Charles Darwin publicó el origen de las especies (1859) [En línea]. 04 de Noviembre de 2009. Disponible en:
6. https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-686X2009000200006 [Consultado: 06 de Agosto de 2019].
7. Khan Academy. Darwin, evolución y selección natural. [En línea]. [Fecha de consulta: 06 de Agosto de 2019]. Disponible en: https://es.khanacademy.org/science/biology/her/evolution-and-natural-selection/a/darwin-evolution-natural-selection

PRÁCTICA 3
DECODIFICACIÓN DE LA POBLACIÓN (BINARIO- REAL)

OBJETIVO

- El alumno conocerá la decodificación del valor real que representa la ristra.
- Desarrollará un programa que emule la decodificación binaria a el valor real.

INTRODUCCIÓN

El origen de las especies expone sus ideas acerca de la evolución y la selección natural. Estas ideas se basaron en gran medida en las observaciones directas que Darwin realizó en sus viajes alrededor del mundo [Pérez, 2009].

Debido a que los rasgos ventajosos son heredables y a que los organismos que los portan dejan más descendientes, los rasgos tenderán a volverse más comunes (presentarse en una mayor parte de la población) en la siguiente generación.

En el transcurso de varias generaciones, la población se adaptará a su entorno (ya que los individuos con rasgos ventajosos en ese ambiente tendrán consistentemente un mayor éxito reproductivo que sus pares).

Mendel pensaba, que, con el control del tipo de cruzas entre los diferentes individuos, se podría rastrear la herencia de ciertas características durante varias generaciones y, con esto, establecer los principios que explican su herencia o transmisión. A partir de ello eligió deliberadamente características simples con formas claramente perceptibles y no intermedias, por ejemplo, el tipo de la semilla era liso o rugoso, la planta tenía un tallo alto o enano, etc. Haciendo estas cruzas durante varias generaciones, lo cual le permitió y pudo explicar la forma de transmisión de los caracteres.

En sí para esta segunda parte se realizará un programa en Java con IDE NetBeans para poder aplicar la función de decodificación, es decir obtener su máximo y el mínimo de cada individuo, tras haber convertir los números binarios [0 y 1] a decimales, esto permitirá ir ordenando los individuos de mayor a menor tras el transcurso del avance en la programación.

De acuerdo a los valores obtenidos de manera decimal se empezará a realizar un algoritmo genético, incluyendo pasos y propiedades que (Holland, 1975) menciona para la realización de cruces con respecto al algoritmo genético, el cual es un método adaptativo que puede usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por (Darwin, 1859) y tomando parte de sus leyes de (Mendel, 1865) conocidas como reglas básicas sobre la transmisión por herencia genética de las características de los organismos de padres a sus hijos. Por imitación de este proceso, dichos algoritmos genéticos son capaces de ir creando soluciones para problemas del mundo real.

La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de estas.

Decodificación del cromosoma

Para determinar el número real que la cadena binaria de caracteres representa. Es importante tener en cuenta que la cadena no representa un número real, sino que este número binario etiqueta un número dentro del intervalo inicialmente fijado.

Decodificación de un individuo, es decir, buscar el valor REAL que corresponde a la cadena binaria «c» de longitud «l»:

$$x = x_{min} + Vcad_{bin} * \frac{x_{max} - x_{min}}{2^l - 1} \text{.....(a)}$$

Ejemplo:
Se tiene una cadena binaria de longitud [5], expresada de esta manera: 01011, y se está trabajando en el intervalo [1,2], obtenga el valor real.

Solución: el valor entero de la cadena **01011** es 11, al aplicar y sustituir en [a], se tiene:

$$x = 1 + 11 * \frac{2 - 1}{2^l - 1}$$
$$x = 1.354838$$

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

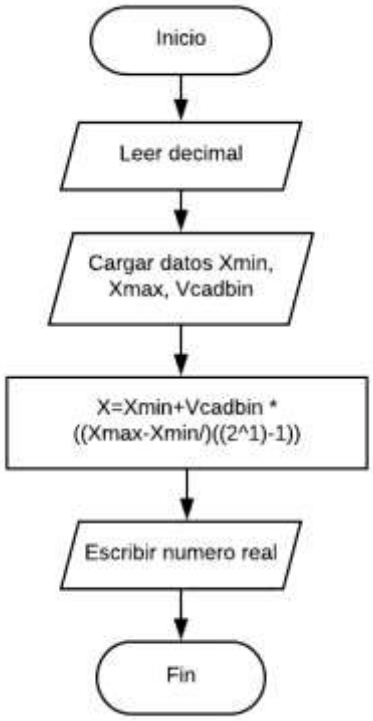


Figura 1. Diagrama de flujo para convertir el valor binario a valor real

Diseñar un programa que retome el valor binario del número decimal (programa anterior) para obtener el valor "real" que sea congruente con la fórmula:

$$x = x_{min} + Vcad_{bin} * \frac{x_{max} - x_{min}}{2^{l} - 1} \text{.....(a)}$$

Se sugiere considerar el seudocódigo y código, que a continuación se muestra:

Seudocódigo:

```
Proceso Decimal_Real
      Escribir 'Ingresar Decimal: ';
      Leer Decimal;
      Escribir 'Ingresar Xmin';
      Leer Xminl;
      Escribir 'Ingresar Xmax: ';
      Leer Decimal;

      Si Decimal == true entonces
```

```
      Hacer X= Xmin+Vcadbin * ((Xmax-Xmin/)((2^1)-1));
      Imprimir X;
      Fin Si
FinProceso
```

CÓDIGO:

```
BinaryToDecimal btd = new BinaryToDecimal();
int valueDecimal = btd.convertBinaryToDecimal(binary, binary.length - 1);
Decimal.setText(Integer.toString((int) valueDecimal));

double a = (Double.parseDouble(Decimal.getText()));
double b = ((Double.parseDouble(max.getText()) - Double.parseDouble(min.getText()))
        / (Math.pow(2, Double.parseDouble(ran.getText())) - 1));
double r = Double.parseDouble(min.getText()) + a * b;
real.setText(Double.toString(r));
double as = Double.parseDouble(real.getText());
```

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente.

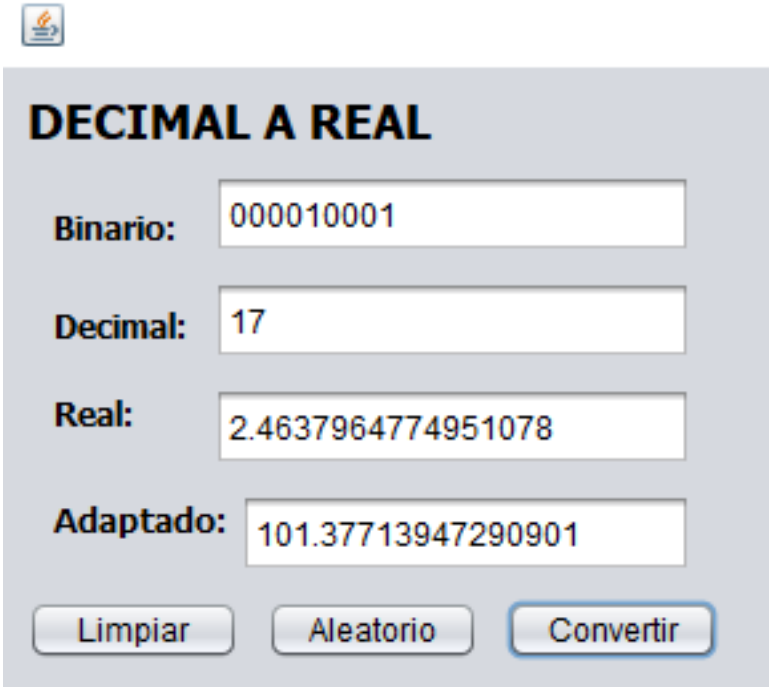


Figura 2. Ejecución programa

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

_____

_____

_____

_____

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.
5. Pérez, V. (2009). Cuando Charles Darwin publicó el origen de las especies (1859) [En línea]. 04 de Noviembre de 2009. Disponible en:
6. https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-686X2009000200006 [Consultado: 06 de Agosto de 2019].
7. Khan Academy. Darwin, evolución y selección natural. [En línea]. [Fecha de consulta: 06 de Agosto de 2019]. Disponible en: https://es.khanacademy.org/science/biology/her/evolution-and-natural-selection/a/darwin-evolution-natural-selection

PRÁCTICA 4
INICIALIZACIÓN DE LA POBLACIÓN (REAL- ADAPTADO)

OBJETIVO

- El alumno conocerá el valor real que representa el fenotipo la ristra.
- Desarrollará un programa que emule el valor real del cromosoma.

INTRODUCCIÓN

Los algoritmos genéticos constituyen una técnica de búsqueda fundamentada en el proceso de evolución natural en la cual los individuos más adaptados tienen mayores probabilidades de sobrevivir y de transferir su material genético a las siguientes generaciones. La idea fundamental de los algoritmos genéticos consiste en encontrar una solución aceptable a un problema por medio del mejoramiento de un conjunto de individuos, cuya función de evaluación corresponde a una solución del problema. Esta optimización se realiza mediante procesos selectivos y de intercambio de información genética (Ruge & Alvis, 2009).

Algunos de las propiedades de los algoritmos Genéticos Holland son las siguientes:

1. Que exista un espacio de soluciones posibles llamado espacio de soluciones o espacio de búsqueda en este sentido se sabe que la solución exacta del problema es un individuo que vive o se encuentra dentro de ese conjunto.

2. Que sea posible representar las posibles soluciones mediante esquemas decodificación (binario-decimal-real-adaptado).

3. Que de alguna forma exista artefactos de estos códigos capaces de decodificarlos y representarlos con cierta clase de individuos o agentes (programa emulando el proceso).

4. Los individuos se colocan a se liberan en un espacio de búsqueda. El ambiente del espacio de búsqueda impone restricciones sobre estas dichas restricciones implican una función de aptitud que evalúa a cada individuo entendiendo este como una posible solución del problema de optimización (Que satisfaga el valor máximo de la función.

En términos biológicos, el conjunto de parámetros representando un cromosoma particular se denomina fenotipo. El fenotipo contiene la información requerida para construir un organismo, el cual se refiere como genotipo. Los mismos términos se utilizan en el campo de los Algoritmos Genéticos. La adaptación al problema de un individuo depende de la evaluación del genotipo. Esta última ´ puede inferirse a partir del fenotipo, es decir puede ser computada a partir del cromosoma, usando la función de evaluación. La función de adaptación debe ser diseñada ˜ para cada problema de manera específica. Dado un

cromosoma particular, la función de adaptación le asigna un numero ´ real, que se supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

Durante la fase reproductiva se seleccionan los individuos de la población para cruzarse y producir descendientes, que constituirán, una vez mutados, la siguiente generación de individuos.

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

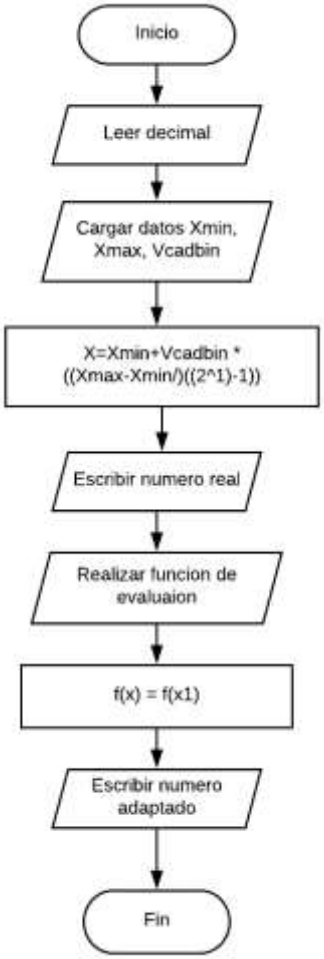


Figura 1. Diagrama de flujo para convertir el valor real al valor de adaptación.

Dependiendo de la fórmula que sea, sustituyendo el valor de x, se tendrán dos valores, el mínimo y el máximo.

Teniendo:

$$F = 8cosx$$

Sustituyendo los valores en dicha fórmula, quedaría de la siguiente manera:

$$F = 8\cos(0) = 8$$
$$F = 8\cos(45) = 4.20$$

Se sugiere considerar el seudocódigo y código, que a continuación se muestra:

Seudocódigo:

```
Proceso Decimal_Real
      Escribir 'Ingresar Decimal: ';
      Leer Decimal;
      Escribir 'Ingresar Xmin';
      Leer Xminl;
      Escribir 'Ingresar Xmax: ';
      Leer Decimal;

      Si Decimal == true entonces
      Hacer X= Xmin+Vcadbin * ((Xmax-Xmin/)((2^1)-1));
      Imprimir X;
      Realizar f(x) = f(x1)
      Imprimir numero_adaptado;
      FinSi
FinProceso
```

Fragmento de código que implementa el método de conversión de real a adaptado y lo imprime en salida estándar.

```
public class main {
    public static void main(String[] args) {
        OperacionesGeneticas OG=new OperacionesGeneticas();
        String Binari="01010101";
        float Decimal=OG.Binario_Entero(Binari);
        float Real=OG.Real(0, 87, 17,8);
        System.out.println("Real"+Real+"Adaptado->"+OG.formula(Real));
    }
}
```

Ejecución del código

```
1.0
2.0
4.0
8.0
16.0
32.0
64.0
128.0
Real->5.8 Adaptado->118.75546
BUILD SUCCESSFUL (total time: 1 second)
```

Figura 1. Ejecución programa

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.
5. Pérez, V. (2009). Cuando Charles Darwin publicó el origen de las especies (1859) [En línea]. 04 de Noviembre de 2009. Disponible en:
6. https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-686X2009000200006 [Consultado: 06 de Agosto de 2019].
7. Khan Academy. Darwin, evolución y selección natural. [En línea]. [Fecha de consulta: 06 de Agosto de 2019]. Disponible en: https://es.khanacademy.org/science/biology/her/evolution-and-natural-selection/a/darwin-evolution-natural-selection

PRÁCTICA 5
PARÁMETRO DE PRECISIÓN EN LA LONGITUD DE UN CROMOSOMA

OBJETIVO

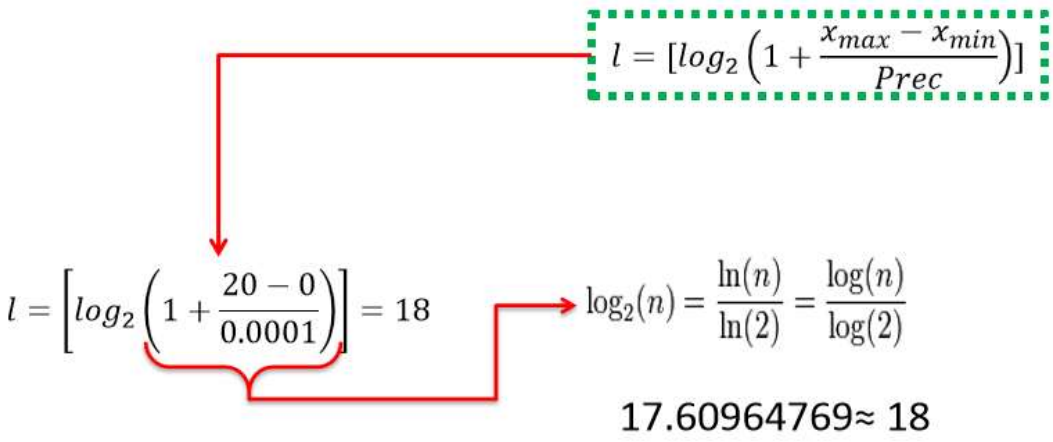
- El alumno conocerá el funcionamiento del parámetro de presión para obtener la longitud del cromosoma o ristra
- Desarrollará un programa que emule el funcionamiento del parámetro de presión para obtener la longitud del cromosoma o ristra

INTRODUCCIÓN

La *precisión* indica la reproducibilidad de los resultados y puede definirse como la concordancia entre los valores de dos o más medidas obtenidas de la misma manera y para la misma muestra.

El parámetro precisión [Prec] indica el número de posiciones decimales en las cuales nos interesa que el resultado de esta correspondencia sea exacto, condiciona la longitud de la cadena binaria de nuestro algoritmo.

la formula dada para el cálculo de la longitud por medio de la precisión:



«a mayor precisión requerida => más larga la cadena»

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

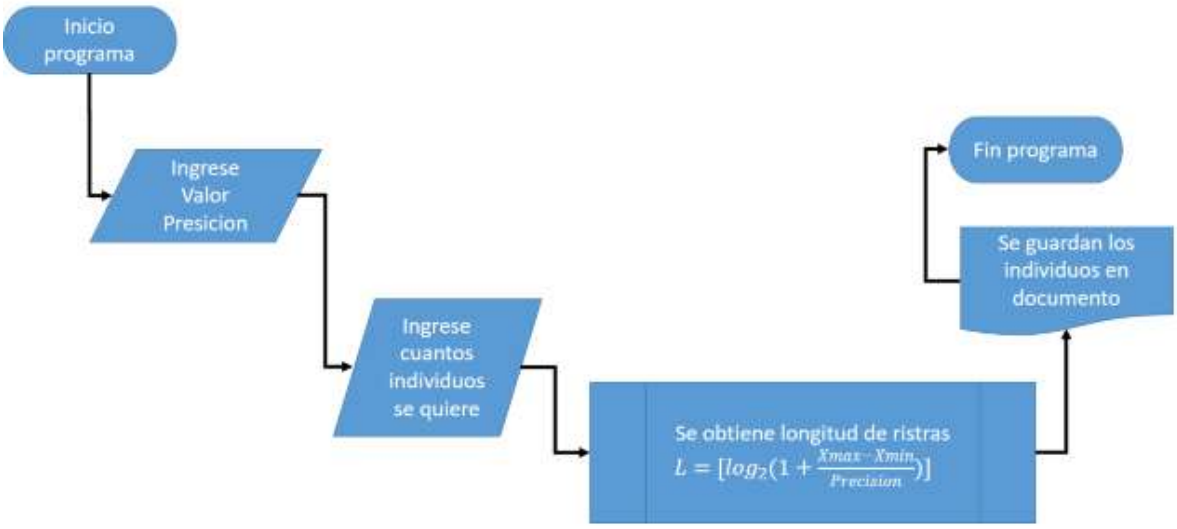


Figura 1. Diagrama de flujo para definir el la longitud de la ristra.

Se sugiere considerar el fragmento de código, que a continuación se muestra:

```
public class Num_aleat {

    public int FPrecision(double prec, double max, double min){
        double precision = 0, n;
        n = ( 1 + ((max - min) / (prec)));
        precision = Math.log10(n) / Math.log10(2);
        return (int) Math.round(precision);
    }
```

Paso1.-Tener los valores de Xmax, Xmin y Prec.

Xmax=20,Xmin=0,Prec=0.0001

Paso2. Una vez realizado el acomodo de la formula solo nos quedará sustituir los valores obtenidos anteriormente, y de esta manera obtendremos la longitud.

log2(n)=log(20-0)/log(0.0001):17.6 redondeando tenemos 18.

Paso3.-El último paso será tomarlo como parámetro para poder crear los individuos que requiera el usuario.

```
Output - ActividadesSeduca_AG (run)
run:
Ingrese el valor maximo
20
Ingrese el valor minimo
0
Ingrese la precisión
0.0001
Ingrese el número de individuos
10
La longitud de la ristra es: 18.0
1.- 110111100010011001
2.- 010110000100101001
3.- 111011000100101111
4.- 011001011110101110
5.- 101100011100111111
6.- 001100101010011110
7.- 010100011101111111
8.- 011101001101111010
9.- 111011011101000010
10.- 011110011010010010
```

Los resultados pueden ser correspondientes a estos:

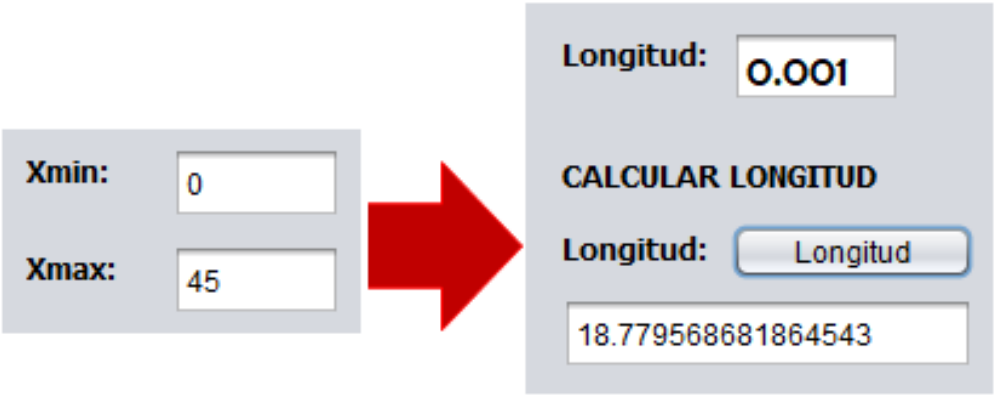


Figura 2. Ejecución del programa

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

_____

_____

_____

_____

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.
5. Pérez, V. (2009). Cuando Charles Darwin publicó el origen de las especies (1859) [En línea]. 04 de Noviembre de 2009. Disponible en:
6. https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-686X2009000200006 [Consultado: 06 de Agosto de 2019].
7. Khan Academy. Darwin, evolución y selección natural. [En línea]. [Fecha de consulta: 06 de Agosto de 2019]. Disponible en: https://es.khanacademy.org/science/biology/her/evolution-and-natural-selection/a/darwin-evolution-natural-selection

PRÁCTICA 6

OPERADOR DE SELECCIÓN

OBJETIVO

- El alumno conocerá el funcionamiento del operador de selección
- Desarrollará un programa que emule el funcionamiento del operador de selección

INTRODUCCIÓN

La selección de padres se efectúa al azar usando un procedimiento que favorezca a los individuos mejor adaptados, ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada. Según´ dicho esquema, los individuos bien adaptados se escogerán probablemente varias veces por generación, mientras que los pobremente adaptados al problema, no se escogerán más que de vez en cuando. Una vez seleccionados dos padres, sus cromosomas se combinan, utilizando habitualmente los operadores de cruce y mutación.

En la naturaleza, la selección sirve para determinar qué individuos sobreviven y se reproducen y cuales desaparecerán por una incorrecta adaptación al entorno. Este proceso es imitado mediante los métodos de selección. El proceso de selección es el que ha despertado un mayor interés a juzgar por la gran cantidad de estudios y métodos desarrollados.

Los métodos de selección, basándose en la calidad de las soluciones (adaptación de los individuos), seleccionan a los mejores individuos (soluciones) para que se reproduzcan, formado parte de la población de padres. Por el contrario, los peores individuos (soluciones) no serán seleccionados y desaparecerán de la búsqueda. Este proceso permite que una misma solución pase varias copias a la población de padres. Se trata, por tanto, de un muestreo sin sustitución.

Un concepto de gran importancia es la presión selectiva, que indica el grado con el que las mejores soluciones son favorecidas para formar parte de la población padre. A mayor presión selectiva, más se favorecerá a los mejores individuos y, por lo tanto, tendrán un mayor número de copias en la población padre. Esta presión dirige al algoritmo hacia poblaciones cada vez mejor adaptadas al problema.

La ratio de convergencia de un algoritmo genético está en gran parte determinado por la magnitud de la presión selectiva. Con una alta presión selectiva se obtiene una mayor ratio de convergencia, con el inconveniente de aumentar la probabilidad de convergencia prematura hacia un subóptimo. Y, por el contrario, si la presión es demasiado baja, la ratio de convergencia disminuirá y el algoritmo tardará más tiempo en encontrar la solución óptima, si es que la encuentra. En el caso más extremo, donde la presión selectiva es nula,

es decir, todos los individuos tienen la misma probabilidad de sobrevivir, se producirá una búsqueda aleatoria por todo el espacio de soluciones.

Los métodos existentes en la actualidad se dividen en tres, los basados en la calidad, en el orden y los que realizan la selección mediante torneos o competiciones. A continuación, y basándose en los trabajos de Goldberg (1989*)* y Michalewicz (1995), se describirán los principales métodos de selección:

Métodos proporcionales a la calidad.

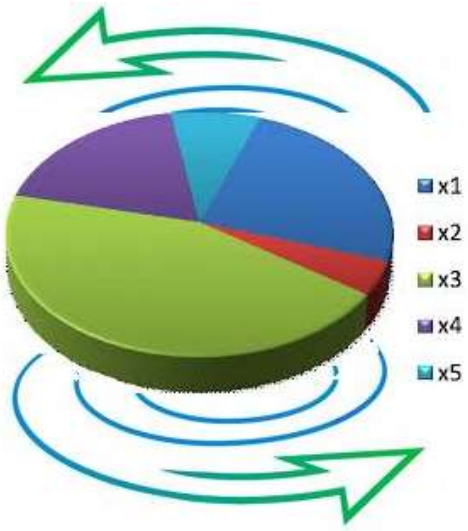
En este caso, la selección de padres se realiza teniendo en cuenta la magnitud de la calidad de las soluciones respecto de la del resto de la población. Por esto, superindividuos, es decir, los que tienen una calidad bastante superior a la del resto de la población, rápidamente coparán los puestos de la población de padres con un gran número de copias. El resultado es una pérdida de la diversidad en la población y una prematura convergencia. Dentro de esta categoría cabe destacar:

Selección proporcional o método de la ruleta, desarrollado en un principio por Holland en 1975, y ampliamente analizado por otros autores como Brindle en su Tesis Doctoral en 1981 y Goldberg (1989). La probabilidad de selección p , para el individuo i está dada por la siguiente ecuación:

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$$

siendo *fi* la calidad del individuo *i*, y *n* el tamaño de la población.

La representación gráfica de este sistema es una ruleta dividida en n áreas cada una de ellas proporcional a la calidad de cada individuo de la población. La selección consistirá rotar n veces la ruleta y la zona donde caiga el marcador indicará el individuo seleccionado para reproducirse.



De esta forma, los individuos con mayor calidad relativa tendrán mayores probabilidades de ser seleccionados para la población de padres, y además más de una vez. Por lo que se presenta el problema de superindividuos y, por lo tanto, de convergencia prematura.

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

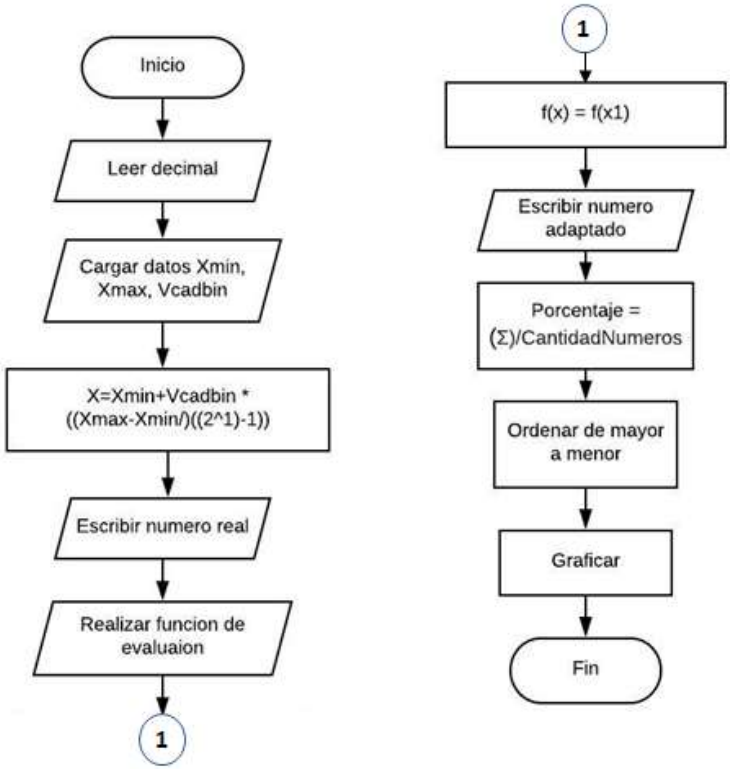


Figura 1. Diagrama de flujo para el operador de selección

Se sugiere considerar el seudocódigo y código, que a continuación se muestra:

Seudocódigo:

```
Seudocódigo:
Proceso Decimal_Real
      Escribir 'Ingresar Decimal: ';
      Leer Decimal;
      Escribir 'Ingresar Xmin';
      Leer Xminl;
      Escribir 'Ingresar Xmax: ';
      Leer Decimal;
      Si Decimal == true entonces
      Hacer X= Xmin+Vcadbin * ((Xmax-Xmin/)((2^1)-1));
      Imprimir X;
      Realizar f(x) = f(x1)
      Imprimir numero_adaptado;
      Si numero_adaptado == true entonces
      Hacer (X)/CantidadNumeros
      Ordenar Mayor a menor
      Imprimir Grafica
```

```
      FinSi
FinProceso
```

Fragmento de código
programa que a continuación se presenta

Código

```
package ruleta;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class Ruleta {

    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);
        int min = 1;
        int max = 15;
        int d, l, n, c, rand, rand2, rand3, auxd1, auxd2;
        c = 0;
        String pa1, pa12, pa2, pa22, h1, h2;
        double auxr1, auxr2, auxa1, auxa2;
        List<Integer> indicer = new ArrayList<>();
        System.out.print("Cuantos numeros se ingresaran ");
        n = sc.nextInt();
        String[] i = new String[n];
        double[] r = new double[n];
        double[] a = new double[n];
        String[] i2 = new String[n];
        double[] r2 = new double[n];
        double[] a2 = new double[n];
        do {
            System.out.print((c + 1) + ".- Introduce un numero
binario ");
            i[c] = sc.next();
            l = i[c].length();
            d = Integer.parseInt(i[c], 2);
            r[c] = min + d * ((max - min) / (Math.pow(2, l) - 1));
            a[c] = 3 * Math.sin(Math.toRadians(r[c])) + 2 * r[c] + 1;
            c += 1;
        } while (c != n);
        for (int m = 1; m <= n; m++) {
            indicer.add(m);
        }
        Collections.shuffle(indicer);
        for (int x = 0; x < n; x++) {
            i2[x] = i[indicer.get(x) - 1];


             rand = ThreadLocalRandom.current().nextInt(1, l - 1);
             do {
                 rand2 = ThreadLocalRandom.current().nextInt(0, n);
                 rand3 = ThreadLocalRandom.current().nextInt(0, n);
             } while (rand2 == rand3);
             pa1 = i2[rand2].substring(rand, l);
             pa12 = i2[rand2].substring(0, rand);
             pa2 = i2[rand3].substring(rand, l);
             pa22 = i2[rand3].substring(0, rand);
             h1 = pa22.concat(pa1);
             h2 = pa12.concat(pa2);
             auxd1 = Integer.parseInt(h1, 2);
             auxd2 = Integer.parseInt(h2, 2);
             auxr1 = min + auxd1 * ((max - min) / (Math.pow(2, l) -
 1));
```

Los resultados pueden ser correspondientes a estos:

Datos iniciales

Id	Binario	Decimal	Real	Adaptado	Porcentaje
0	000000000	0	0.0	8.0	89.25297226653116
1	000100101	37	3.258317025440313	-7.9455635323540825	88.64564519939503
2	110110011	435	38.30724070450098	6.565747974119963	73.25156523039529
3	001110001	113	9.951076320939334	-6.91737965958985	77.17458686430496
4	000101000	40	3.522504892367906	-7.42660704943837	82.85584412724344
5	100110001	305	26.85909980430528	-1.2394846651997518	13.828461305983012

Figura 2. Sin ruleta

Con proporción ruleta

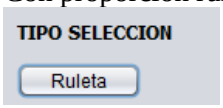
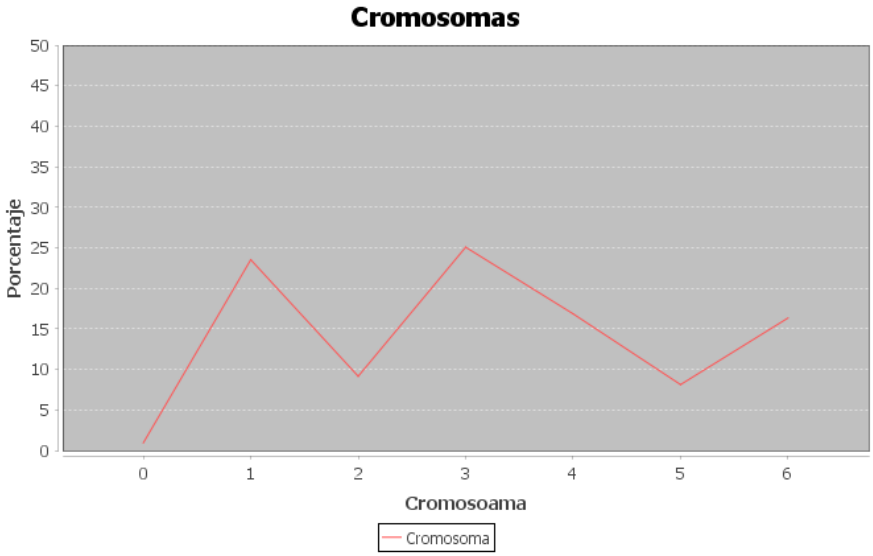


Figura 3. Botón para ruleta

Se muestra de la siguiente manera:

Id	Binario	Decimal	Real	Adaptado	Porcentaje
0	000000000	0.0	0.0	8.0	89.25297226653116
1	000100101	37.0	3.258317025440313	-7.9455635323540825	88.64564519939503
2	000101000	40.0	3.522504892367906	-7.42660704943837	82.85584412724344
3	001110001	113.0	9.951076320939334	-6.91737965958985	77.17458686430496
4	110110011	435.0	38.30724070450098	6.565747974119963	73.25156523039529
5	100110001	305.0	26.85909980430528	-1.2394846651997518	13.828461305983012

Figura 3. Ejecución del programa



Grafica 1. Grafica de datos

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

_____

_____

_____

_____

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

BIBLIOGRAFÍA Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.
5. Pérez, V. (2009). Cuando Charles Darwin publicó el origen de las especies (1859) [En línea]. 04 de Noviembre de 2009. Disponible en:
6. https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-686X2009000200006 [Consultado: 06 de Agosto de 2019].
7. Khan Academy. Darwin, evolución y selección natural. [En línea]. [Fecha de consulta: 06 de Agosto de 2019]. Disponible en: https://es.khanacademy.org/science/biology/her/evolution-and-natural-selection/a/darwin-evolution-natural-selection

PRÁCTICA 7

OPERADOR DE CRUCE DE UN PUNTO

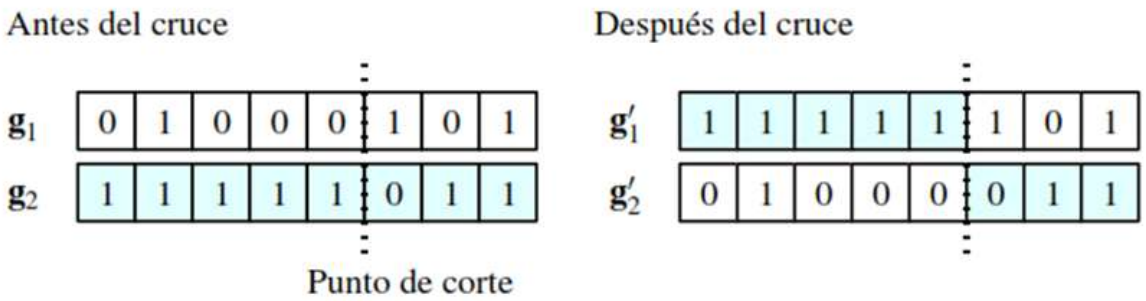
OBJETIVO

- El alumno conocerá el funcionamiento del operador de cruce de un punto
- Desarrollará un programa que emule el operador de cruce de un punto.

INTRODUCCIÓN

Durante la fase reproductiva se seleccionan los individuos de la población para cruzarse y producir descendientes, que constituirán, una vez mutados, la siguiente generación de individuos. La selección de padres se efectúa´ al azar usando un procedimiento que favorezca a los individuos mejor adaptados, ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada. Según dicho esquema, los individuos bien adaptados se escogerán probablemente varias veces por generación, mientras que los pobremente adaptados al problema, no se escogerán más que de vez en cuando. Una vez seleccionados dos padres, sus cromosomas se combinan, utilizando habitualmente los operadores de cruce y mutación. Las formas básicas de dichos operadores se describen a continuación. El operador de cruce coge dos padres seleccionados y corta sus ristras de cromosomas en una posición escogida al azar, para producir dos subristras iniciales y dos subristras finales. Después se intercambian las subristras finales, produciéndose dos nuevos cromosomas completos

Existen diferentes tipos de Operadores Genéticos, uno de ellos es el CRUCE DE UN PUNTO, que juega el papel más importante dentro de los operadores genéticos, ya que se permite el intercambio de características de una generación a la siguiente. Se desarrolla probabilísticamente hablando, se le asigna una probabilidad (p) para que se pueda ejecutar. Su valor depende del tipo de problema, de la representación utilizada, y de muchos otros factores. Este cruce es operado a partir de una pareja de genotipos seleccionados g1 y g2 y seleccionado un punto de corte aleatorio, se generan dos descendientes h1 y h2 intercambiando parte de su información.

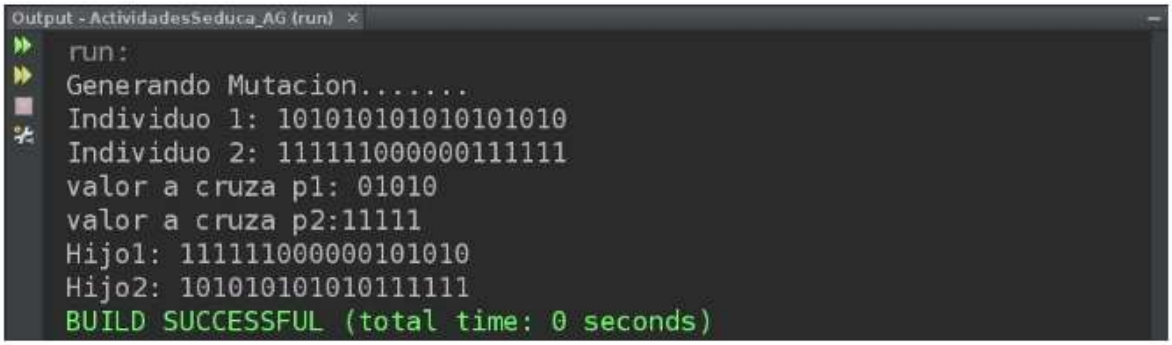


DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

Figura 1. Diagrama de flujo para el operador de cruce de un punto

Se obtendrá una salida estándar controlada de esta manera:



INICIO.

- Se declaran las variables.
- MIENTRAS que la bandera sea verdadera:
- Se solicitan los valores binarios.
- Se almacenan en un vector
- Pregunta si desea ingresar mas valores:
- SI no termina los ciclos anteriores
- Comienza a imprimier los valores binarios almacenados en el vector
- SI NO regresa al ciclo while para agregar un valor nuevo y se repite.
- Se seleccionan dos individuos al azar para ser los padres.
- Se clacula el punto de corte de manera aleatorea.
- Se hace el Cruce de un Punto.
- Se muestra cuales fueron los hijos obtenidos.
- Se calcula: Decimal->Real->Funcion Adaptativa de ambos hijos.
- Se almacena el valor de la funcion adaptativa de cada hijo.
- Se comparan los valores obtenidos contra los de los padres.
- Se remplaza el padre por el hijo mas fuerte.
- Se repite varias veces hasta que se cumple el umbral.
- Se ingresan los individuos a un random para elegir el individua a mutar.
- Se calcula la mutacion.
- Se imprime el resultado de la mutacion y los cruces. FIN.

```
27 -    end
28      %_______
29      % extrae aleatoriamente del vector N los numeros binario
30 -    b=round((i-1)*rand(1,2)+1); extraccion=N(b);%numeros de extraccion ose
31      %los numeros binarios extraidos aleatroriamente se tranforman en binar
32 -    A=dec2bin(extraccion)
33      %
34 -    P=A(1,:)
35 -    str_P = num2str(P)
36 -    O=A(2,:)
37 -    str_O= num2str(O)
38 -    puntoC=input('\n Ingrese el punto de corte para esta iteracion:\n');
39 -    numf=longitud-puntoC;
40 -      for k = 1:numf
41 -      w1=str_P(k)~=str_O(k)
42 -       str_P(k)= str_O(k)
43 -        w3=str_O(k)~=str_P(k)
44 -       str_O(k)= str_P(k)
45 -      end
```

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente.

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones,
   ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press

2. - PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley

3.- Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.

4.- Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.

PRÁCTICA 8
OPERADOR DE CRUCE DE DOS PUNTOS

OBJETIVO

- El alumno conocerá el funcionamiento del operador de cruce de dos puntos.
- Desarrollará un programa que emule el operador de cruce de dos puntos.

INTRODUCCIÓN

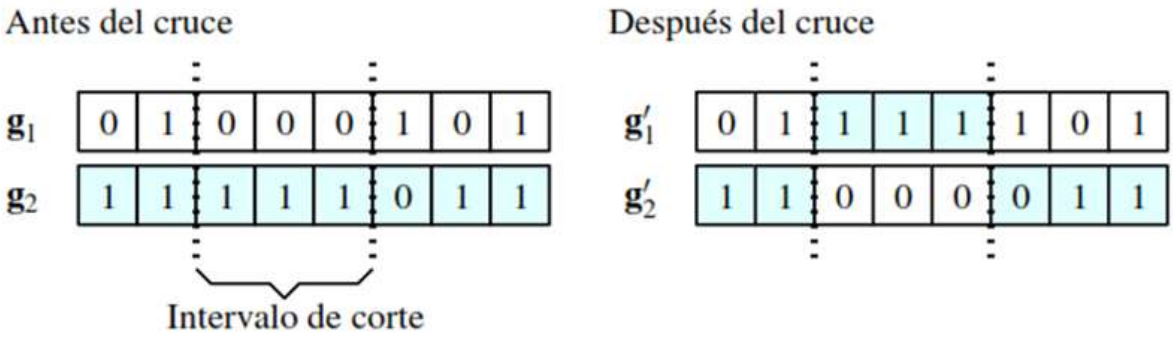
Tras parametrizar el problema en una serie de variables, se codifican en un cromosoma. Todos los operadores utilizados por un algoritmo genético se aplicarán sobre estos cromosomas, o sobre poblaciones de ellos. En el algoritmo genético va implícito el método para resolver el problema. Hay que tener en cuenta que un algoritmo genético es independiente del problema, lo cual lo hace un algoritmo robusto, al resultar útil en cualquier ámbito de acción, pero a la vez débil, pues no está especializado en ninguno. Las soluciones codificadas en un cromosoma compiten para ver cuál constituye la mejor solución (aunque no necesariamente la mejor de todas las soluciones posibles).

El ambiente, constituido por las otras soluciones, ejercerá una presión selectiva sobre la población, de forma que sólo los mejor adaptados (aquellos que resuelvan mejor el problema) sobrevivan o leguen su material genético a las siguientes generaciones, igual que en la evolución de las especies. La diversidad genética se introduce mediante mutaciones y reproducción sexual. Por lo tanto, un algoritmo genético consiste en hallar de qué parámetros depende el problema, codificarlos en un cromosoma, y aplicar los métodos de la evolución: selección y reproducción sexual con intercambio de información y mutaciones que generen diversidad.

Una vez se realiza la selección de los cromosomas se procede a realizar la reproducción o cruce entre dos de estos cromosomas Más concretamente, el crossover consiste en el intercambio de material genético entre dos cromosomas. El objetivo del cruce es conseguir que el descendiente mejore la aptitud de sus padres. Para aplicar el cruce habrá que seleccionar con anterioridad dos individuos de la población con una de las diversas técnicas de selección que hemos mencionado en el punto anterior. Además, esta selección puede elegir el mismo padre para un descendiente. Esto no es ningún problema pues se asegura la perpetuación del cromosoma más dominante, pero si este cruce se produjese con mucha frecuencia podría acarrear consecuencias adversas en caso de que ese cromosoma dominante presente algunos genes no deseados.

Crossover de dos Puntos

Se trata de la misma filosofía que en el caso anterior peo en este caso los padres se cortan por dos puntos. Se copiará al descendiente los genes de un cromosoma progenitor desde el principio hasta el primer punto de cruce, los genes del otro progenitor desde el primer punto de cruce hasta el segundo y del segundo punto de cruce hasta el final se copiará del otro progenitor.



La finalidad que tiene un algoritmo genético es optimizar funciones, de acuerdo con la selección, cruce y mutación los cuales tienen la finalidad de identificar valores repetidos y de qué manera se podrá optimizar.

El cruce de dos puntos consiste en dos puntos de cortes, los cuales son generados de manera aleatoria, donde una vez generados los puntos de cortes se realiza un intercambio de ellos generando así nuevos individuos.

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

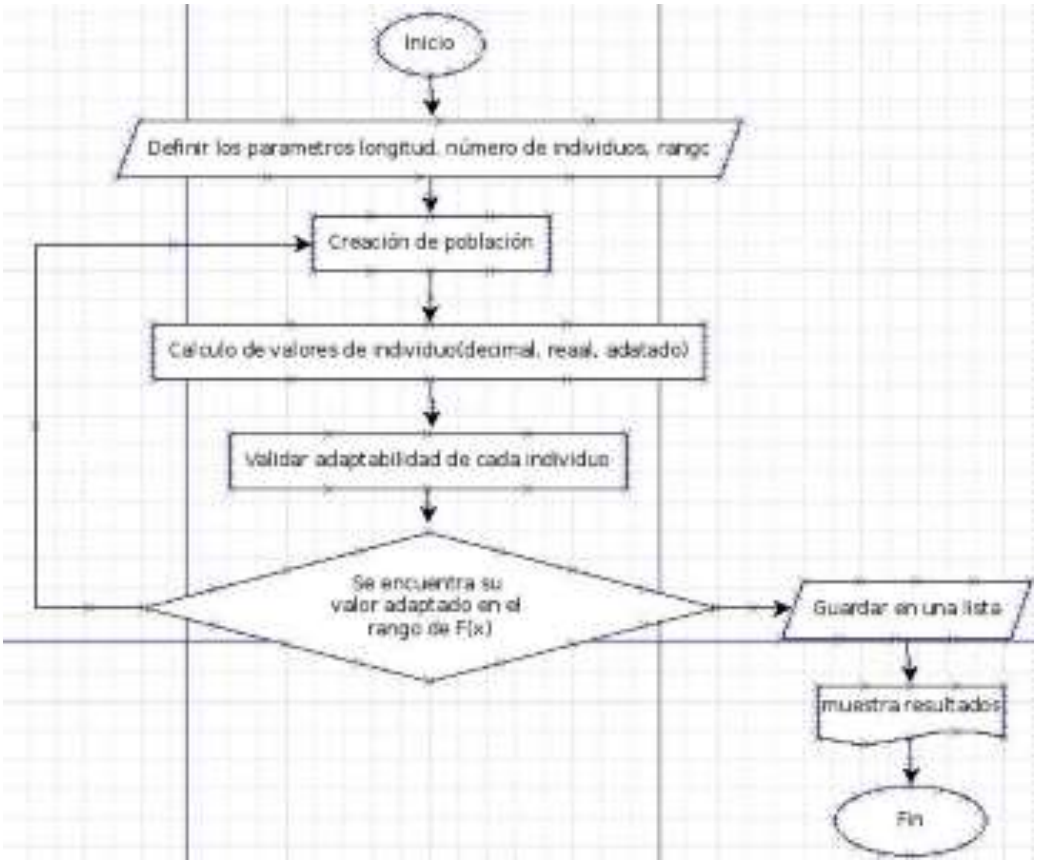


Figura 1. Diagrama de flujo para el operador de cruce de dos puntos.

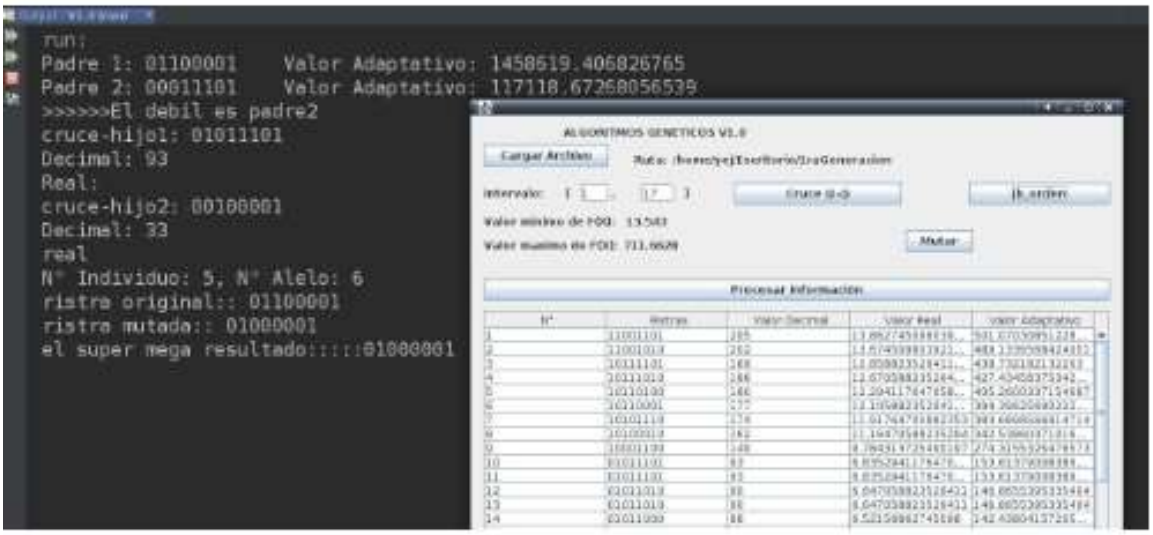
Código:

```
//cruces
for(int i=0;i<cadenas.length;i++){

    c1=cadenas[i].substring(0,3);
    p1.add(c1);
    c2=cadenas[i].substring(3,7);
    p2.add(c2);

}
```

```
         Todos los sujetos
01010110
11100011
00100101
10100100
11110001
00110100
10111111
00101001
Mascara = 00110101
Padre uno 11100011
Hijo uno 00100001
Padre dos 00101001
Hijo dos 00100001
Padre dos 00101001 es sustituido por hijo dos 00100001
         Todos los sujetos
01010110
11100011
00100101
10100100
11110001
00110100
10111111
00100001
```



Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente.

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

_____

_____

_____

_____

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.

PRÁCTICA 8

OPERADOR DE CRUCE UNIFORME

OBJETIVO

- El alumno conocerá el funcionamiento del operador de cruce uniforme
- Desarrollará un programa que emule el funcionamiento de este operador

INTRODUCCIÓN

El operador cruce es el que juega el papel más importante dentro de los operadores genéticos, ya que permite el intercambio de características de una generación a la siguiente; es decir, el que hace evolucionar a las especies. Su realización es probabilística, se le asigna una probabilidad (*p*) para que se pueda ejecutar. Su valor depende del tipo de problema, de la representación utilizada, y de muchos factores más. Inicialmente el valor dado fue del 60% (Goldberg, 1989), pero han surgido muchos estudios de cómo adaptar estas probabilidades al estado del proceso genético (Michalewicz, 1995).

Es importante considerar que el operador cruce realiza dos trabajos, los denominados mecanismo e idea (Jones, 1995). La *idea* del cruce se basa en el hecho de que cada uno de los padres seleccionados tiene una probabilidad mayor que la media de la generación actual, de ofrecer un mejor material genético. La razón principal de mantener un conjunto de soluciones y utilizar el cruce es formar cada vez mejores individuos y combinarlos a su vez para conseguir otros nuevos aún mejores: cumplir el teorema de los bloques constitutivos. Si la idea del cruce así definida no se desarrolla en el algoritmo elegido, entonces no merece la pena gastar recursos en mantener una población de soluciones.

El *mecanismo* del cruce es el proceso por el que se lleva a cabo la idea. Es decir, la forma del intercambio, la aplicación literal de los métodos para la implementación de cada operador cruce que serán explicados a continuación.

Es importante separar estos dos conceptos del cruce ya que, si un algoritmo realiza sólo el mecanismo, el resultado final es tal que no importa entonces que los padres seleccionados sean los mejores, al no aprovecharse sus buenas características. Se puede hablar de macromutación de los padres: cambio de la mayor parte del material genético de cada uno de los padres por otro que, en principio, no les aporta beneficios, tal y como pasa en las mutaciones.

Existe otro método menos estudiado de producir hijos, pero en cuyo proceso no están implicados directamente dos padres. Fue desarrollado inicialmente por Syswerda en 1993 y modificado por Eshelman y Schaffer (1993). Consiste en la creación de los hijos teniendo en

cuenta toda la información que proporciona el total de individuos de la generación actual. Para crear los nuevos individuos, se observa la frecuencia de los valores existentes en cada posición, y por medio de dicha frecuencia se van construyendo los hijos de forma estocástica.

Todos los operadores utilizados por un algoritmo genético se aplicarán sobre estos cromosomas, o sobre poblaciones de ellos. En el algoritmo genético va implícito el método para resolver el problema. Hay que tener en cuenta que un algoritmo genético es independiente del problema, lo cual lo hace un algoritmo robusto, al resultar útil en cualquier ámbito de acción, pero a la vez débil, pues no está especializado en ninguno. Las soluciones codificadas en un cromosoma compiten para ver cuál constituye la mejor solución (aunque no necesariamente la mejor de todas las soluciones posibles).

El ambiente, constituido por las otras soluciones, ejercerá una presión selectiva sobre la población, de forma que sólo los mejor adaptados (aquellos que resuelvan mejor el problema) sobrevivan o leguen su material genético a las siguientes generaciones, igual que en la evolución de las especies. La diversidad genética se introduce mediante mutaciones y reproducción sexual. Por lo tanto, un algoritmo genético consiste en hallar de qué parámetros depende el problema, codificarlos en un cromosoma, y aplicar los métodos de la evolución: selección y reproducción sexual con intercambio de información y mutaciones que generen diversidad.

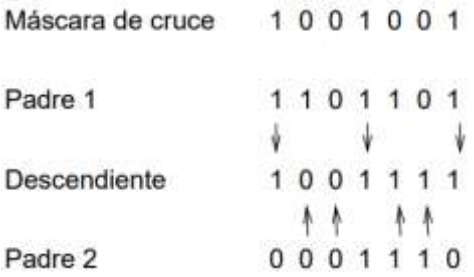
Una vez se realiza la selección de los cromosomas se procede a realizar la reproducción o cruce entre dos de estos cromosomas Más concretamente, el cruce consiste en el intercambio de material genético entre dos cromosomas. El objetivo del cruce es conseguir que el descendiente mejore la aptitud de sus padres. Para aplicar el cruce habrá que seleccionar con anterioridad dos individuos de la población con una de las diversas técnicas de selección que hemos mencionado en el punto anterior. Además, esta selección puede elegir el mismo padre para un descendiente. Esto no es ningún problema pues se asegura la perpetuación del cromosoma más dominante, pero si este cruce se produjese con mucha frecuencia podría acarrear consecuencias adversas en caso de que ese cromosoma dominante presente algunos genes no deseados.

Operador cruce uniforme.

Es la generalización de los operadores anteriores. Cada gen tiene una probabilidad del 50% de ser intercambiado por el correspondiente del otro padre, utilizándose para ello una máscara (Figura 5). Fue desarrollado por primera vez por Ackeley en 1987, y posteriormente Syswerda (1989) realizó un estudio comparativo entre este operador y los operadores 1-punto y 2puntos. La conclusión Fue que el cruce uniforme funciona mejor que los otros dos en estudio.

El cruce uniforme es una técnica completamente diferente de las vistas hasta el momento. Cada gen de la descendencia tiene las mismas probabilidades de pertenecer a uno u otro padre. Aunque se puede implementar de muy diversas formas, la técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen situado en esa posición en uno de los descendientes se copia del primer padre. Si por el contrario hay un 0 el gen se copia del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

- Cruce uniforme: para cada gen de la cadena del descendiente existe una probabilidad de que el gen pertenezca al padre, y otra de que pertenezca a la madre.



Operador cruce uniforme

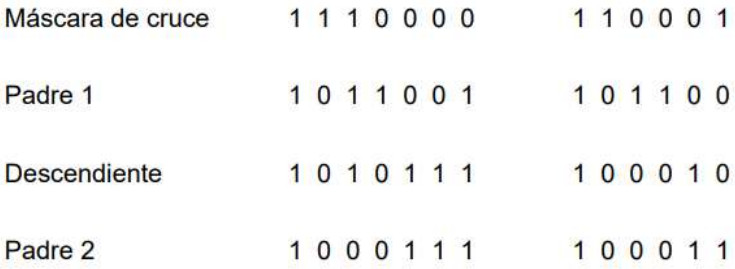


Figura 1. “Máscaras de cruce” para los operadores de cruce basados en 1 punto y en 2 puntos

En la literatura, el término operador de cruce uniforme se relaciona con la obtención de la “máscara de cruce” uniforme, en el sentido de que cualquiera de los elementos del alfabeto tenga asociada la misma probabilidad. Hablando en términos de la teoría de la probabilidad la máscara de cruce está compuesta por una muestra aleatoria de tamaño˜ λ extraída de una distribución de probabilidad de Bernouilli de parámetro 1/2.

Cada gen del descendiente se obtiene de cualquiera de los padres de forma aleatoria. Una opción es generar un número aleatorio. Si este número supera un cierto umbral se elegirá un padre determinado y si no lo supera se elige al otro. Otra opción es seleccionar una máscara. En caso de que el bit correspondiente a la máscara esté a 1, se copia el gen de un progenitor y en caso de que esté a 0 se copia el gen del otro progenitor.

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo que a continuación se muestra:

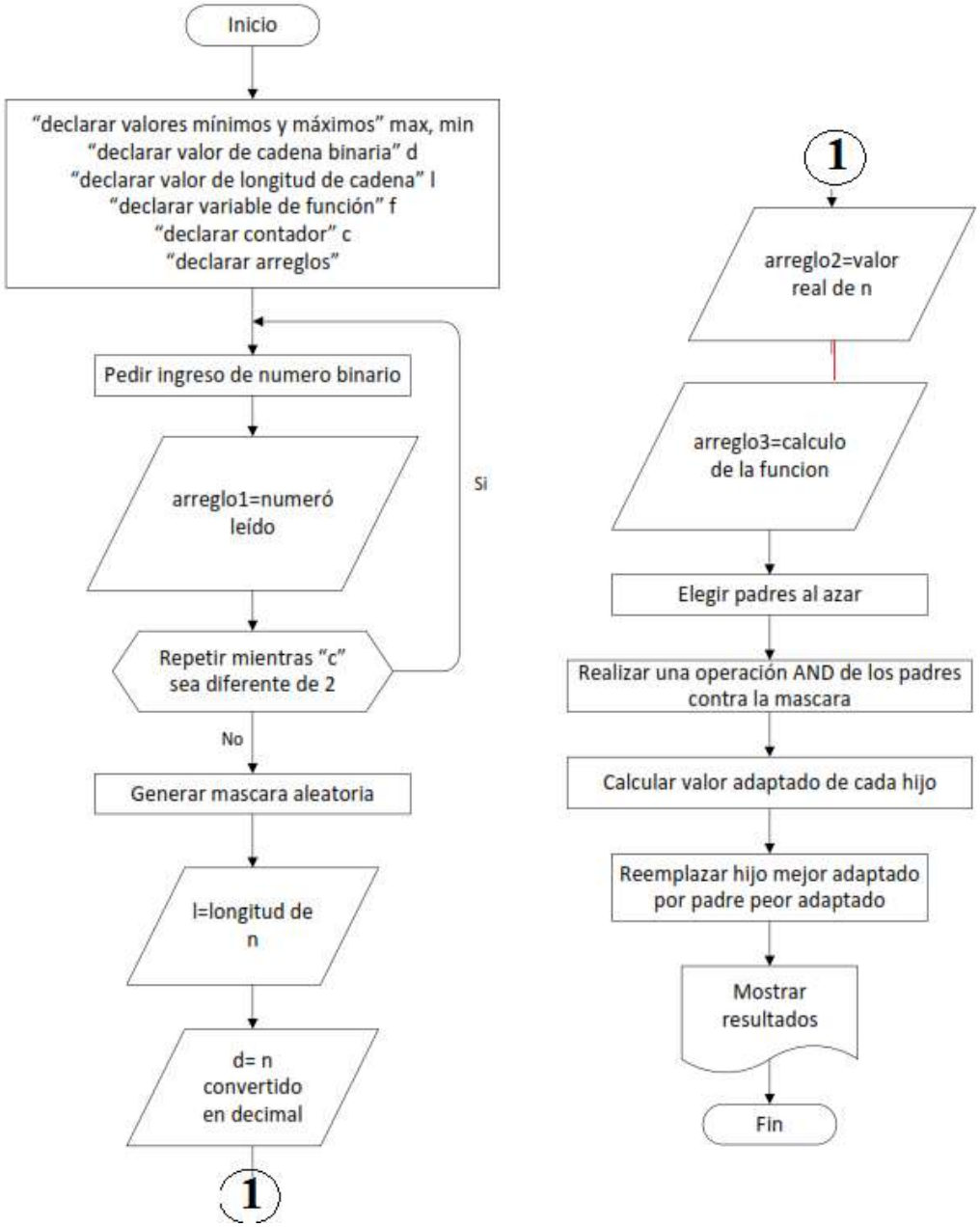


Figura 2. Diagrama de flujo para el operador de cruce uniforme.

Puede considerar el programa que a continuación se presenta

Código.

```
package cruceu;

import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class CruceU {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int c = 0;
        int min = 1;
        int max = 15;
        String[] i = new String[2];
        double[] r = new double[2];
        double[] a = new double[2];
        int l, d, auxd1, auxd2, mascara, rand3;
        String pa1, pa2, mas = null;
        double auxr1, auxr, auxa1, auxa2;
        StringBuilder masc;
        masc = new StringBuilder();
        StringBuilder h1 = new StringBuilder();
        StringBuilder h2 = new StringBuilder();
        do {
            System.out.print((c + 1) + ".- Introduce un numero binario ");
            i[c] = sc.next();
            l = i[c].length();
            d = Integer.parseInt(i[c], 2);
            r[c] = min + d * ((max - min) / (Math.pow(2, l) - 1));
            a[c] = 3 * Math.sin(Math.toRadians(r[c])) + 2 * r[c] + 1;
            c++;
        } while (c != 2);
        c = 0;
        while (c < l) {
            char car = '1';
            masc.append(car);
            c++;
        }
        mascara = Integer.parseInt(masc.toString(), 2);
        do {
            rand3 = ThreadLocalRandom.current().nextInt(1, mascara);
            if (Integer.toBinaryString(rand3).length() < l) {
                c = l;
```

```
}
mascara = Integer.parseInt(masc.toString(), 2);
do {
    rand3 = ThreadLocalRandom.current().nextInt(1, mascara);
    if (Integer.toBinaryString(rand3).length() < l) {
        c = l;
        masc.delete(0, masc.length());
        while (c > Integer.toBinaryString(rand3).length()) {
            char car = '0';
            masc.append(car);
            c--;
        }
        masc.append(Integer.toBinaryString(rand3));
        mas = masc.toString();
    }

  } while (mas == null);

    pa1 = i[0];
    pa2 = i[1];
    System.out.println("Mascara " + mas);
    for (int x = 0; x < l; x++) {
        if (pa1.substring(x, x + 1).equals("1")) {
            if (mas.substring(x, x + 1).equals("1")) {
                h1.append('1');
            } else {
                h1.append('0');
            }
        } else {
            h1.append('0');
        }
        if (pa2.substring(x, x + 1).equals("1")) {
            if (mas.substring(x, x + 1).equals("1")) {
                h2.append('1');
            } else {
                h2.append('0');
            }
        } else {
            h2.append('0');
        }

      if (pa2.substring(x, x + 1).equals("1")) {
          if (mas.substring(x, x + 1).equals("1")) {
              h2.append('1');
          } else {
              h2.append('0');
          }
      } else {
          h2.append('0');
```

En la figura 3 se muestra la población con la que se va a trabajar:

Id	Binario	Decimal	Real	Adaptado	Porcentaje
0	010010100	148	13.033268101761252	7.143752643340369	30.86400699788733
1	111111110	510	44.9119373776908	4.7849787556296235	20.67311470200053
2	001010111	87	7.6614481409001955	1.5307694971224735	6.61356612275544
3	000000101	5	0.44031311154598823	7.236946018458695	31.266641457031174
4	101100010	354	31.174168297455967	7.767348286191435	33.558201666392506
5	110010010	402	35.40117416829745	-5.317893750456621	22.975530946066982

Figura 3. Datos iniciales

En la figura 4 se muestra el porcentaje calculado de cada uno de los números binarios del comienzo. Como se puede ver aún no se ha realizado la ruleta.

Id	Binario	Decimal	Real	Adaptado	Porcentaje
0	010010100	148	13.033268101761252	7.143752643340369	30.86400699788733
1	111111110	510	44.9119373776908	4.7849787556296235	20.67311470200053
2	001010111	87	7.6614481409001955	1.5307694971224735	6.61356612275544
3	000000101	5	0.44031311154598823	7.236946018458695	31.266641457031174
4	101100010	354	31.174168297455967	7.767348286191435	33.558201666392506
5	110010010	402	35.40117416829745	-5.317893750456621	22.975530946066982

Figura 4. Datos con cálculos

En la figura 5 se muestran los datos ordenados debido a la selección ruleta:

Id	Binario	Decimal	Real	Adaptado	Porcentaje
0	101100010	354.0	31.174168297455967	7.767348286191435	33.558201666392506
1	000000101	5.0	0.44031311154598823	7.236946018458695	31.266641457031174
2	010010100	148.0	13.033268101761252	7.143752643340369	30.86400699788733
3	110010010	402.0	35.40117416829745	-5.317893750456621	22.975530946066982
4	111111110	510.0	44.9119373776908	4.7849787556296235	20.67311470200053
5	001010111	87.0	7.6614481409001955	1.5307694971224735	6.61356612275544

Figura 5. Población con ruleta

En la figura 6 se muestra la otra ventana, la cual es donde se realizarán los cruces, en la parte superior izquierda se puede ingresar la máscara o con el botón de lado derecho se puede crear aleatoriamente. Ingresando la máscara, las especificaciones de los padres y al tener la población ya ordenada, se prosigue a hacer la mutación, esto eligiendo dos de los individuos que se presentan en la población inicial, siendo estos el primer y segundo padre, los hijos dependen mucho la instrucción que se les dé, ya que pueden cambiarse dependiendo si encuentras un 0 o un 1.

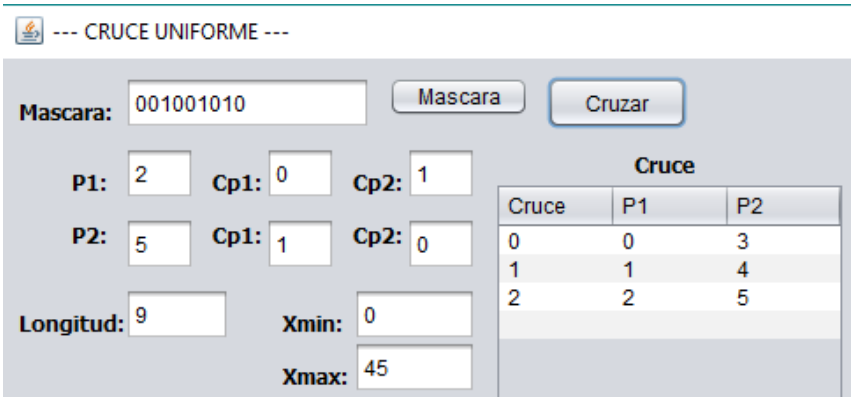


Figura 6. Cruce uniforme

En la figura 7 se muestran los resultados de algunas de los cruces que se realizaron:

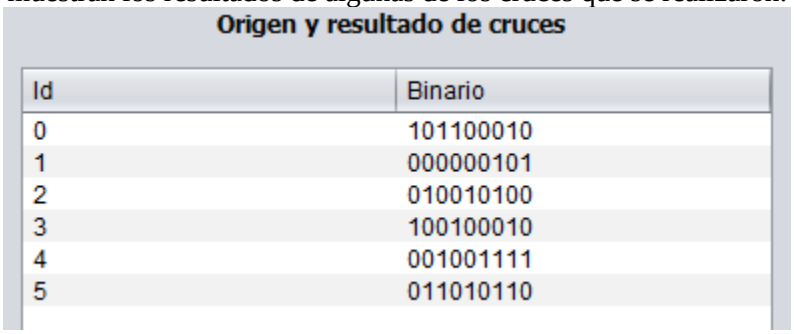


Figura 7. Realizando cruces

En la figura 8 se muestran los hijos ganadores y los padres que se van a sustituir.

Id	Remplaza a padre	Hijo ganador	Binario	Decimal	Real	Adaptado
0	3	1	100100010	290	25.538160469667318	7.351
1	4	1	001001111	79	6.956947162426614	6.252
2	5	1	011010110	214	18.845401174168295	8

Figura 8. Padres ganadores y a sustituir

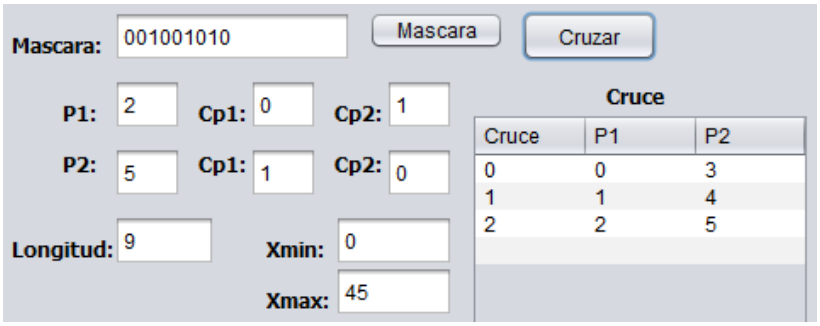


Figura 1. Procedimiento para cruce uniforme

Origen y resultado de cruces

Id	Binario
0	101100010
1	000000101
2	010010100
3	100100010
4	001001111
5	011010110

Figura 2. Origen de cruces

Los resultados de los cruces se muestran en la figura 11:

Id	Remplaza a padre	Hijo ganador	Binario	Decimal	Real	Adaptado
0	3	1	100100010	290	25.538160469667318	7.351
1	4	1	001001111	79	6.956947162426614	6.252
2	5	1	011010110	214	18.845401174168295	8

Figura 3. Resultados

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente.

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.

PRÁCTICA 10
OPERADOR DE MUTACIÓN

OBJETIVO

- El alumno conocerá el funcionamiento del operador de mutación
- Desarrollará un programa que emule el funcionamiento de este operador

INTRODUCCIÓN

Tras el cruce, tiene lugar la mutación. Si nos referimos en términos de evolución, la mutación se manifiesta de forma extraordinaria, nada común. Las mutaciones suelen en promedio ser beneficiosas pues contribuyen a la diversidad genética de la especie. Además, previenen a las soluciones de la población de verse limitadas por un óptimo local. Por lo tanto, la mutación consiste en modificar ciertos genes de forma aleatoria atendiendo a la probabilidad de mutación establecida con anterioridad. La mutación depende de la codificación y de la reproducción. Si se abusa de la mutación podemos caer en el uso del algoritmo genético como una simple búsqueda aleatoria. Por lo tanto, antes de aumentar las mutaciones, conviene estudiar otras soluciones que aporten diversidad a la población como podría ser el aumento del tamaño de la población o garantizar la aleatoriedad de la población inicial. Para el caso de una codificación binaria, la mutación consiste simplemente en la inversión del gen mutado que corresponderá con un bit. En el caso de una codificación numérica, la mutación podría consistir en sustituir un número por otro o intercambiar un número por otro que está en otra posición del cromosoma. En el caso de codificación por valor directo en el que por ejemplo usemos números reales, la mutación puede consistir simplemente en modificar el valor en unos decimales. Por último, en una codificación en árbol, la mutación podría radicar en el cambio de operador, de un número o incluso en la mutación de una rama entera
Mientras el operador cruce combina aquellas buenas características de los individuos llevando a cabo la explotación de las áreas beneficiosas detectadas por la selección, el operador mutación incluye la diversidad en la búsqueda facilitando la exploración de áreas aún no tratadas. Por lo tanto, se puede decir que la principal característica de este operador es permitir que todos los individuos del espacio de búsqueda tengan una probabilidad de ser explorados mayor de cero.

Además, su funcionamiento permite recuperar características de los individuos que por medio de la selección y el cruce podrían haber perdido, cayendo en el olvido y que, aun siendo buenas características, sería imposible recuperar sin la acción del operador mutación.

Mutación estándar. Es el operador por excelencia utilizado en la codificación binaria. La probabilidad de aplicación (*pm*) muy baja, para no incluir demasiada diversidad en la búsqueda y perder de esta forma la dirección de la misma. Según los estudios realizados por Holland, la probabilidad aceptable sería la comprendida entre 0.1 y 1%. Esta probabilidad se aplicará bit a bit, a diferencia de los operadores cruce que se refieren a cada pareja o incluso a otros operadores mutación, generalmente los aplicados sobre codificaciones no binarias,

donde se aplicará sobre cada cadena o individuo. El proceso es sencillo y consiste en mutar el bit correspondiente según la probabilidad de mutación asignada (Figura 1).

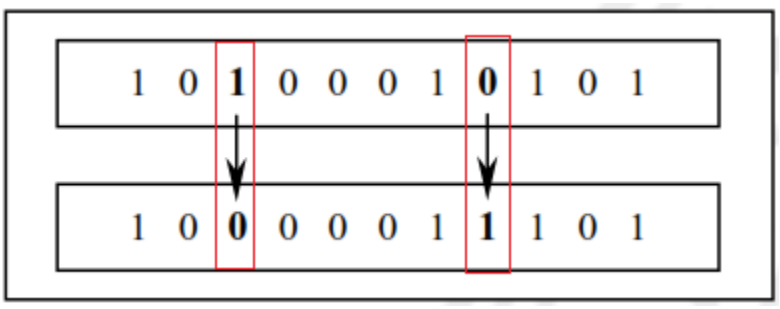


Figura 1. Operador mutación estándar

Los operadores de mutación producen pequeños cambios en los individuos de acuerdo con una probabilidad . Este operador ayuda a prevenir caer en óptimos locales y a extender el espacio de búsqueda del algoritmo.

La mutación es el operador básico de alteración. Se aplica a individuos solos, realizando una pequeña modificación en alguno de sus genes o en el conjunto, permitiendo explorar nuevas zonas del espacio de búsqueda En (Ochoa; Harvey; Buxton, 2000), se realiza un estudio en el que demuestra la importancia de las proporciones de mutación y de selección en la eficiencia del algoritmo. La probabilidad con la que estos cambios se producen suele ser baja, para evitar que el AG realice una búsqueda aleatoria. Sin embargo, en ocasiones puede ser necesario aumentar esta probabilidad para recuperar la diversidad de la población. En algunas ocasiones se puede utilizar un operador de regeneración para evitar problemas de convergencia prematura.

Algunas de las razones que pueden motivar a incorporar son:

- *Desbloqueo del algoritmo*. Si el algoritmo se bloqueó en un mínimo parcial, una mutación puede sacarlo al incorporar nuevos fenotipos de otras zonas del espacio.
- *Acabar con poblaciones degeneradas*. Puede ocurrir que, bien por haber un cuasi-mínimo, bien porque en pasos iniciales apareció un individuo demasiado bueno que acabó con la diversidad genética, la población tenga los mismos fenotipos. Si se ha llegado a una población degenerada, es preciso que las mutaciones introduzcan nuevos genomas.
- *Incrementar el número de saltos evolutivos*. Los saltos evolutivos -aparición de un fenotipo especialmente valioso, o, dicho de otra forma, salida de un mínimo local- son muy poco probables en un genético *puro* para un problema genérico. La mutación permite explorar nuevos subespacios de soluciones, por lo que, si el subespacio es bueno en términos de adaptación, se producirá un salto evolutivo después de la mutación que se expande de forma exponencial por la población.

- *Enriquecer la diversidad genética*. Es un caso más *suave* que el de una población degenerada -por ejemplo, que la población tenga una diversidad genética pobre-, la mutación es un mecanismo de prevención de las poblaciones degeneradas.
-

Sin embargo, si la tasa de mutación es excesivamente alta tendremos la ya conocida deriva genética. Una estrategia muy empleada es una tasa de mutación alta al inicio del algoritmo, para aumentar la diversidad genética, y una tasa de mutación baja al final del algoritmo, para conseguir que converge.

DESARROLLO

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente, puede considerar el diagrama de flujo y segmentos de código que a continuación se muestran:

Diagrama de flujo:

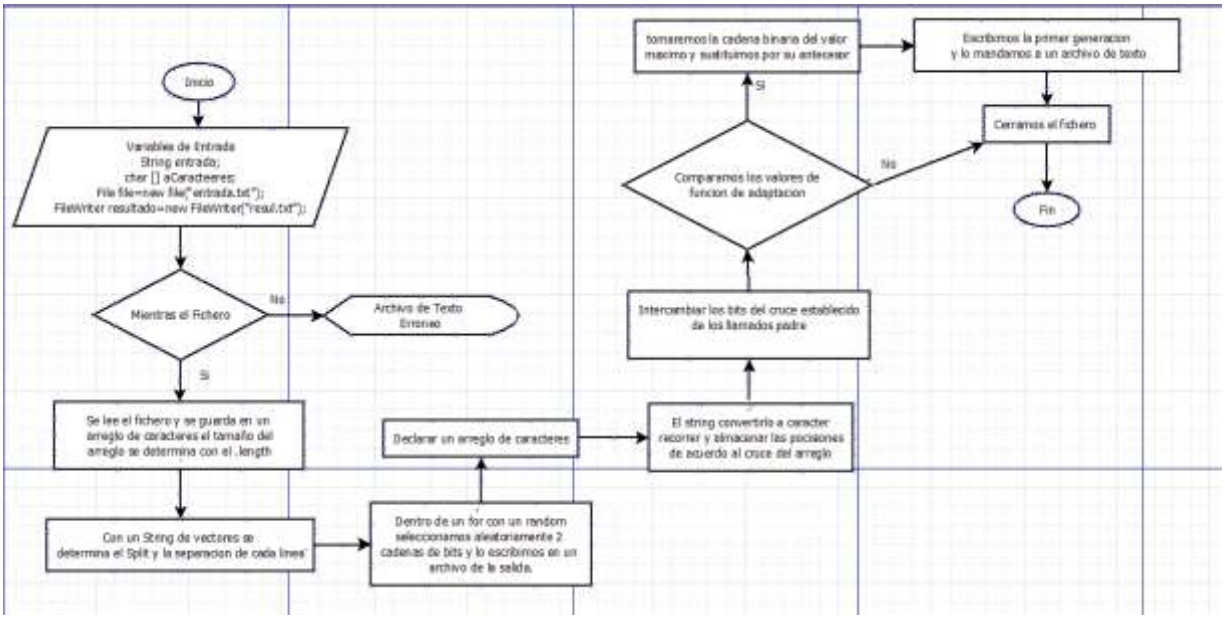


Figura 2. Diagrama de flujo para el operador de mutación.

Código:

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    String array1[]= new String[4], array2[]= new String[4],
           padres[]= new String[2];
    String array[]= new String[4];
    int N= lista.size(), padre1=0, padre2=0, decimal1, decimal2;
    int corte= (int)(Math.random()*Tmax + 1);

    while(padre1 == padre2){
        padre1= (int)(Math.random()*N + 1);
        padre2= (int)(Math.random()*N + 1);
    }

    String cad1= lista.get(padre1-1), cad2= lista.get(padre2-1);

    padres= alg.cruce(cad1, cad2, corte);

    data(padre1, padre2, corte);

    cad1= padres[0];
    cad2= padres[1];

    array1[0]= cad1;
    array1[1]= ConversionDec(cad1)+"";
    array1[2]= alg.Real(Double.parseDouble(array1[1]), min, max, Tmax)+"";
    array1[3]= alg.Funcion(Double.parseDouble(array1[2]))+"";

    array2[0]= cad2;
    array2[1]= ConversionDec(cad2)+"";
    array2[2]= alg.Real(Double.parseDouble(array2[1]), min, max, Tmax)+"";
```

```
array2[0]= cad2;
array2[1]= ConversionDec(cad2)+"";
array2[2]= alg.Real(Double.parseDouble(array2[1]), min, max, Tmax)+"";
array2[3]= alg.Funcion(Double.parseDouble(array2[2]))+"";

modelo2.addRow(array1);
modelo2.addRow(array2);

if(Double.parseDouble(array1[3])> Double.parseDouble(array2[3]))  lista.set(padre1-1, cad1);
else    lista.set(padre2-1, cad2);

String espacio[]= {"****","****","****","****"};
modelo.addRow(espacio);
modelo2.addRow(espacio);
for(String cad: lista){
    array[0]= cad;
    array[1]= ConversionDec(cad)+"";
    array[2]= alg.Real(Double.parseDouble(array[1]), min, max, Tmax)+"";
    array[3]= alg.Funcion(Double.parseDouble(array[2]))+"";
    modelo.addRow(array);
}
```

Ejecución:

Para ejemplificar la mutación, primero debemos crear una población, en este caso va a ser de 6 individuos

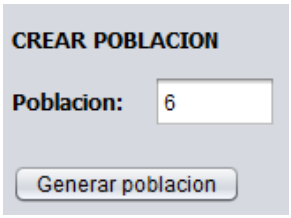


Figura 31. Creación de números binarios

Creando la población quedaría de esta manera:

Id	Binario	Decimal	Real	Adaptado	Porcentaje
0	010010100	148	13.033268101761252	7.143752643340369	30.86400699788733
1	111111110	510	44.9119373776908	4.7849787556296235	20.67311470200053
2	001010111	87	7.6614481409001955	1.5307694971224735	6.61356612275544
3	000000101	5	0.44031311154598823	7.236946018458695	31.266641457031174
4	101100010	354	31.174168297455967	7.767348286191435	33.558201666392506
5	110010010	402	35.40117416829745	-5.317893750456621	22.975530946066982

Figura 4. Resultados de los datos

Realizando algunos cruces:

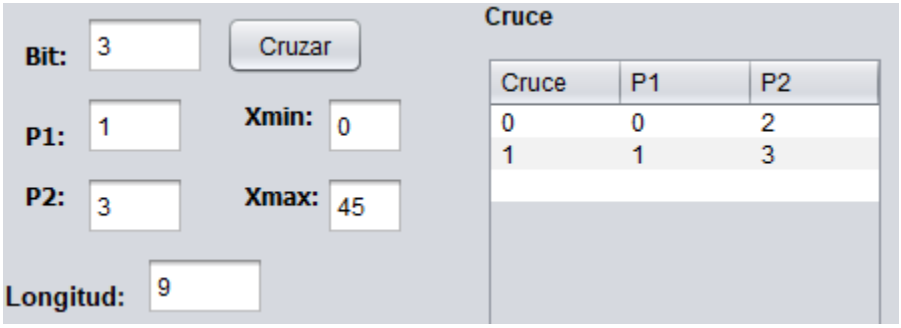


Figura 5. Individuos padres para cruces

Los resultados de los cruces quedarían de esta manera:

Id	Remplaza a padre	Hijo ganador	Binario	Decimal	Real	Adaptado
0	2	1	010010111	151	13.297455968688844	5.956
1	3	2	000000110	6	0.5283757338551859	6.909

Figura 6. Resultados cruces

Los resultados de la nueva población son:

Figura 7. Cruces

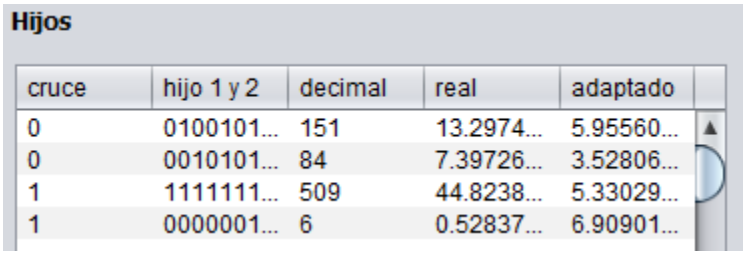


Figura 8. Cruces e hijos

Los datos para la mutación serian:

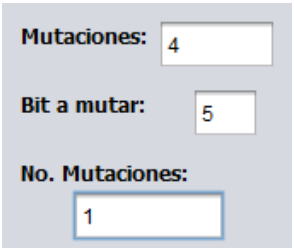


Figura 8. Datos para mutación

Realizando la mutación:

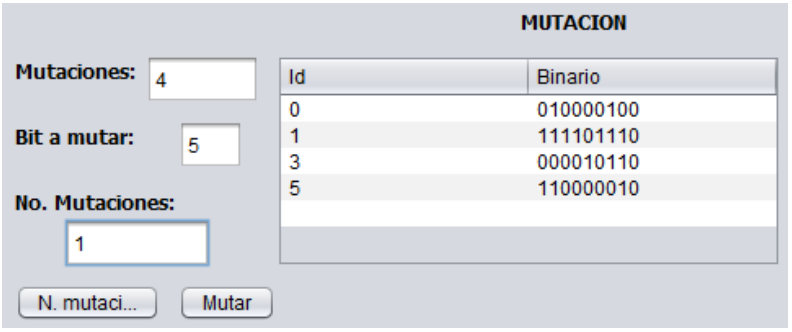


Figura 9. Resultados para mutación

Y la población final, con la mutación, queda de la siguiente manera:

Id	Binario
0	010000100
1	111101110
2	010010111
3	000010110
4	101100010
5	110000010

Figura 10. Mutación

Una parte importante para realizar la mutación es:

```
StringBuilder builder = new StringBuilder();
Random aleatorio = new Random(System.currentTimeMillis());
int pl = aleatorio.nextInt(Integer.parseInt(poblacion.getText()));
int mj= Integer.parseInt(muta.getText());
for(int i=0; i<Integer.parseInt(nC.getText());i++){
ud = ud;
DefaultTableModel m = (DefaultTableModel) Muta.getModel();
DefaultTableModel orri = (DefaultTableModel) Ori.getModel();
int avel = ran(pl);
String origen = orri.getValueAt(avel, 1).toString();
String[] parts = origen.split("");
String[] aux = new String[parts.length];
String aa = "c", bb = "u";
```

Figura 21. Código para hacer la mutación

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente.

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Ochoa, G;I. Harvey, and H. Buxton. Optimal mutation rates and selection pressure in genetic algorithms. In Proceedings of the 2000 Genetic And Evolutionary Computation Conference, pages 315-322. Morgan Kaufmann, 2000.

PRÁCTICA 11

ALGORITMO GENÉTICO SECUENCIAL

OBJETIVO

Que el alumno genere un programa que agrupe el proceso del algoritmo genético secuencial

INTRODUCCIÓN

Los algoritmos evolutivos imitan el proceso de evolución natural, el principal mecanismo que guía la aparición de estructuras orgánicas complejas y bien adaptadas. De forma muy simplificada, la evolución es el resultado de las relaciones entre la creación de nueva información genética y los procesos de evaluación + selección.

Cada individuo en una población se ve afectado por el resto (compitiendo por recursos, emparejándose para procrear, huyendo de los depredadores, ...) y también por el entorno (disponibilidad de comida, clima, ...). Los individuos mejor adaptados son los que tienen mayores posibilidades de vivir más tiempo y reproducirse, generando así una progenie con su información genética (posiblemente modificada). En el transcurso de la evolución se generan poblaciones sucesivas con información genética de los individuos cuya adecuación es superior a la de la media. La naturaleza no determinista de la reproducción provoca una creación permanente de información genética nueva, y por tanto la aparición de distintos individuos.

Los *algoritmos genéticos* han sido muy importantes dentro de las técnicas evolutivas. tradicionalmente han utilizado codificaciones independientes al problema, como son las cadenas binarias, sin embargo, muchas de las aplicaciones recientes se han centrado en otras representaciones tales como grafos, listas ordenadas, números reales, etc. Para la formación de los descendientes la reproducción es sexual, es decir, hay intercambio del material genético de los padres mediante los operadores genéticos, cruce y mutación. En este caso, el operador mutación se considera un operador de segundo orden frente al operador cruce, como principal diferencia con las técnicas evolutivas anteriores.

Se puede decir que los algoritmos genéticos son métodos metaheurísticos poco convencionales. Esto es debido a que trabajan con un conjunto de soluciones al mismo tiempo, imitando los procesos que se llevan a cabo en la evolución natural de las especies. Esta es la principal diferencia con otras metaheurísticas tales como la búsqueda tabú, el recocido simulado, etc. donde una única solución es modificada hasta alcanzar un cierto criterio de parada.

En los últimos años se están desarrollando las técnicas de *programación genética*. En este caso la representación es más compleja el ser estructuras jerarquizadas y de tamaño variable (Koza, 1993). La población está formada por cientos o miles de dichas estructuras que son unidos por operadores apropiados pretendiéndose encontrar aquella combinación mejor adaptada a un determinado problema.

Los algoritmos genéticos es el campo de los algoritmos evolutivos que ha tenido un mayor DESARROLLO. Prueba de ello es el gran número de publicaciones existentes en revistas de gran prestigio como, por ejemplo, *Management Science*, *Operational Research*, etc., y por la creación de grandes conferencias internacionales bianuales: *International Conference on Genetic Algorithms*, *Parallel Problem Solving from Nature*, *Foundations of Genetic Algorithms*.

La Figura 1. Muestra el proceso del Algoritmo Genético Canónico Secuencial, simplificado con las partes más importantes de un algoritmo genético clásico: selección, reproducción y sustitución. Aparte de estos procesos, que pertenecen al núcleo principal y exclusivo de los algoritmos genéticos, existen otros, de igual importancia, que sirven de apoyo a los anteriores: descripción del problema a resolver, codificación de sus soluciones e inicialización. Selección y codificación son los puntos críticos de un algoritmo genético.

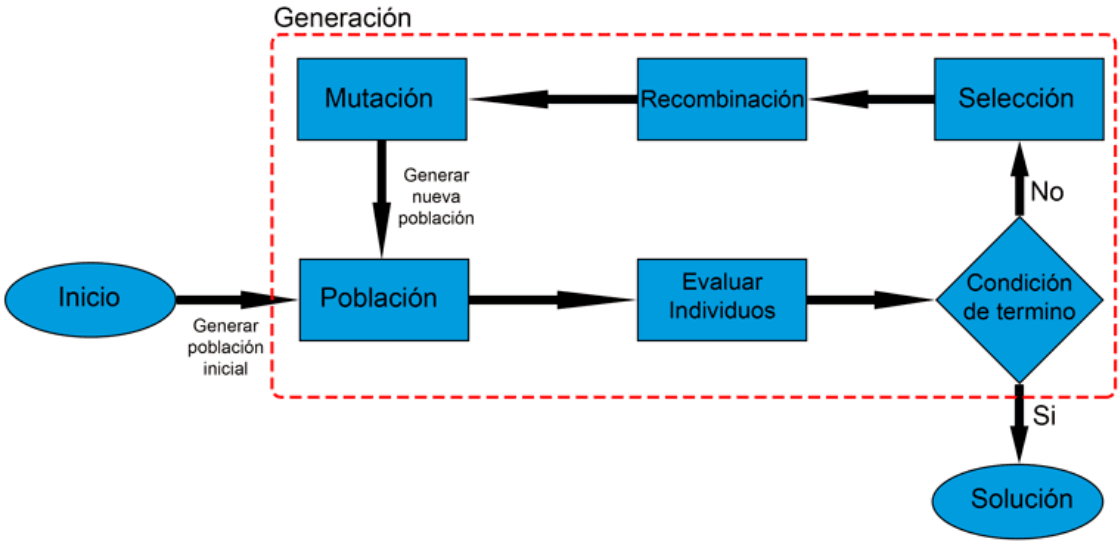


Figura 1. Diagrama del Algoritmo Genético Canónico Secuencial.

DESARROLLO

Realizar un programa que emule el proceso de un algoritmo genético secuencial, considerando el diagrama de flujo, el seudocódigo, así como las salidas de pantalla y graficas que a continuación se presentan, las cuales le servirán de base para diseñar el propio.

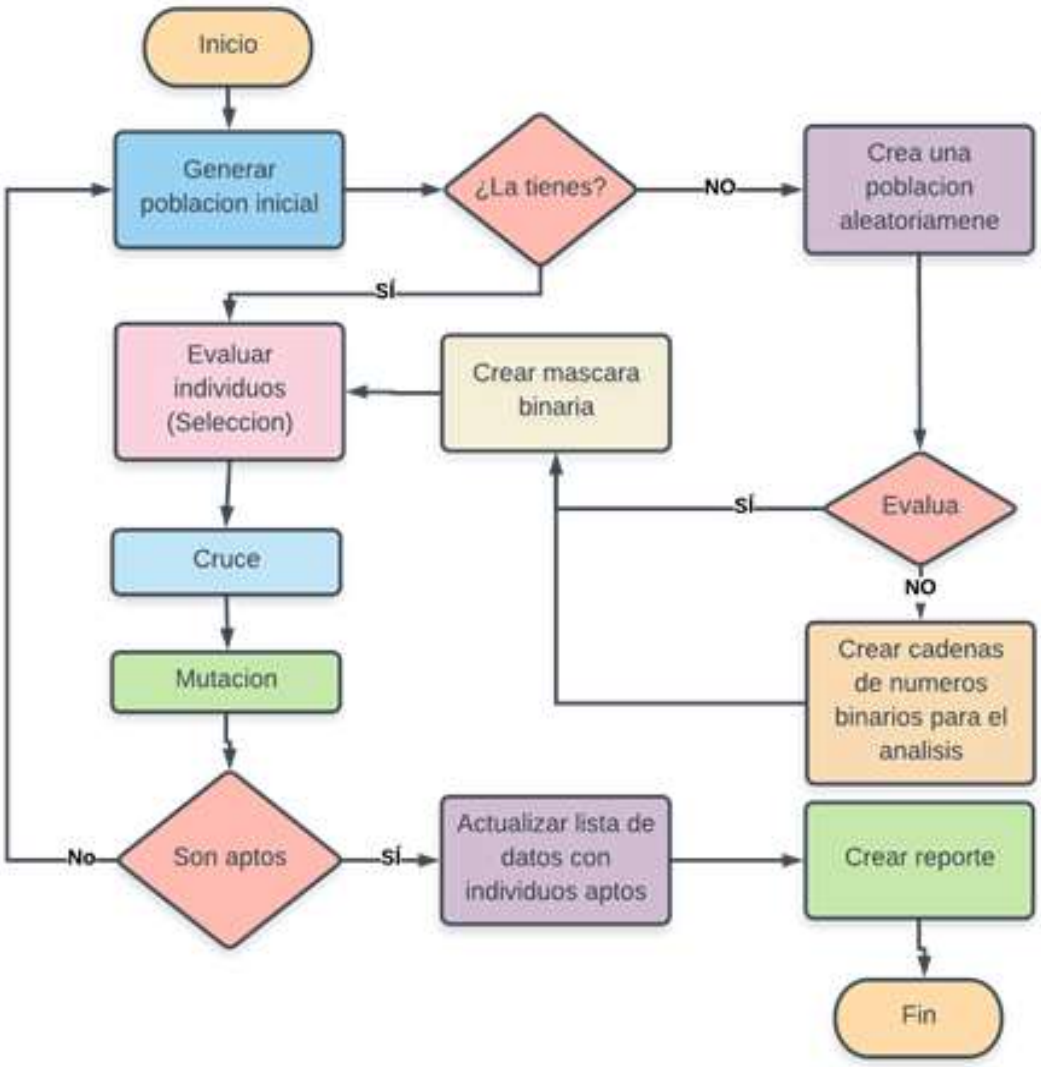


Figura 2. Diagrama se flujo para la ejecución de un AGS.

Seudocódigo.

```
INICIO.
   - Ingresa el usuario el valor
   - Se guarda en la variable llamada "binario".
   - Se declaran vectores "ristra" y "ristraAux".
   - MIENTRAS el valor stream >> aux + Se llena el vector con cada
     componente del #binario.
   - HASTA que se termine de recorre el vector se realizara: +Se
     invierte el valor binario para su manipulacion correcta.
   - SI la el valor indivudual de la ristra
   - -1 es igual a uno: + La variable decimal sera igual a decimal mas
     num.
   - SI NO : + num es igual a num *2.
   - Se almacena el valor decimal.
   - Lee el valor decimal, el mínimo el máximo y el rango.
   - Se guarda en la variable llamada "x-min", "x-max".
   - Se hace la conversión usando la formula obtenida en clase.
   - Se almacena el valor obtenido.
   - Se lee el valor obtenido.
   - Se aplica la formula vista en clase.
   - Se imprimen os 3 resultados obtenidos.
   - FIN.
```

Para la función:

$$F(x) = 2x + 5$$

Con un intervalo de:	[1, 32]
Individuos	50
Cruce de un punto	40
Cruce de dos puntos	40
Cruce uniforme	40
Convergencia	80%
Precisión de:	0.0001
¿Individuos aptos?	4
¿Cruces?	4
Mutación	5%
Individuos mutados	1
F(32)	69
Numero cruces	(50*100)/80

Selección por torneo [20]

DATOS INICIALES

Generando datos iniciales:

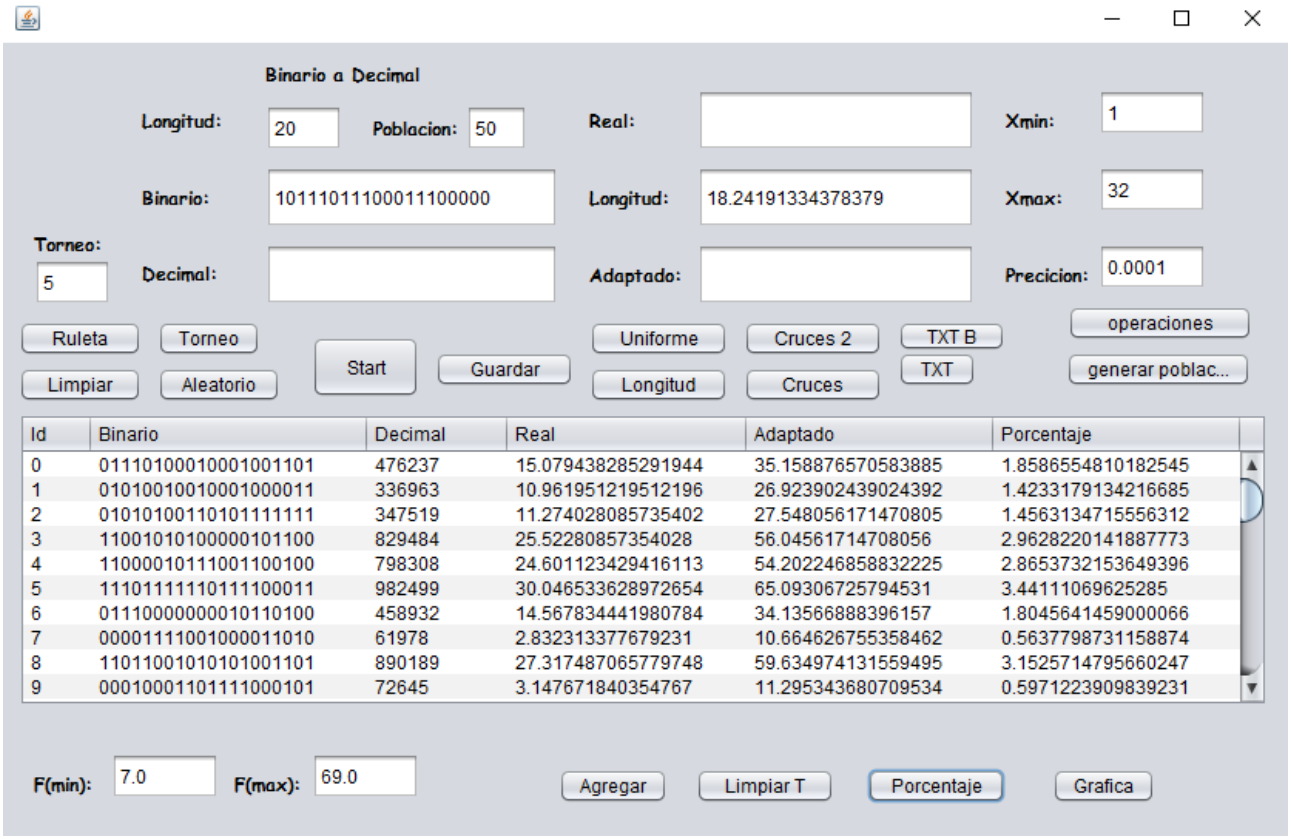


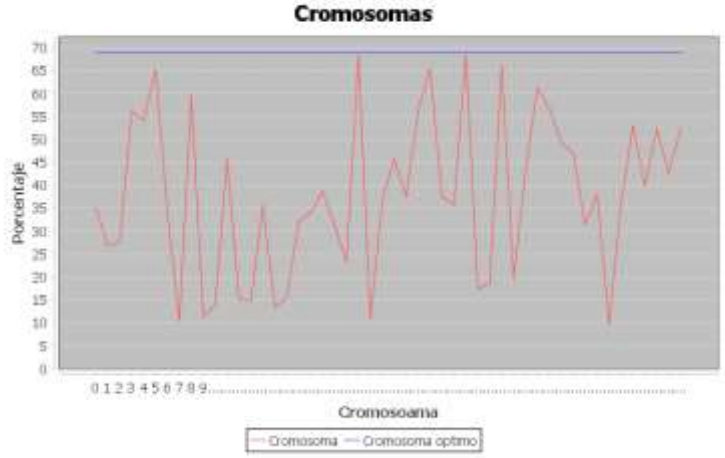
Figura 4. Datos iniciales

Datos iniciales

0	01110100010001001101	476237	15.079438285291944	35.158876570583885	1.8586554810182545
1	01010010010001000011	336963	10.961951219512196	26.923902439024392	1.4233179134216685
2	01010100110101111111	347519	11.274028085735402	27.548056171470805	1.4563134715556312
3	11001010100000101100	829484	25.52280857354028	56.04561714708056	2.9628220141887773
4	11000010111001100100	798308	24.601123429416113	54.202246858832225	2.8653732153649396
5	11101111110111100011	982499	30.046533628972654	65.09306725794531	3.44111069625285
6	01110000000010110100	458932	14.567834441980784	34.13566888396157	1.8045641459000066
7	00001111001000011010	61978	2.832313377679231	10.664626755358462	0.5637798731158874
8	11011001010101001101	890189	27.317487065779748	59.634974131559495	3.1525714795660247
9	00010001101111000101	72645	3.147671840354767	11.295343680709534	0.5971223909839231
10	00011100110101000111	118087	4.491116038433112	13.982232076866223	0.7391633300445093
11	10011111101001010001	653905	20.3320029563932	45.6640059127864	2.414003607443505
12	00100001111010001101	138893	5.1062232076866225	15.212446415373245	0.8041979627212082
13	00100000001011100001	131809	4.896792313377679	14.793584626755358	0.7820550550079651
14	01110110010010100100	484516	15.32419807834442	35.64839615668884	1.8845336759586753
15	00011001110111110001	105969	4.132860310421286	13.265720620842572	0.7012853295265702
16	00100011011000100100	144932	5.284759793052476	15.569519586104953	0.8230744477127773
17	01100111100111101101	424429	13.547790096082778	32.095580192165556	1.696715932344102

18	01101111010011100111	455911	14.478521803399852	33.9570436067997	1.7951212147591666
19	10000011000000110001	536625	16.864745011086477	38.72949002217295	2.047414079407205
20	01100100001111011111	410591	13.138684405025868	31.277368810051737	1.6534616188297233
21	01000100001000100100	279076	9.250583887657058	23.501167775314116	1.2423768492211456
22	11111100011011101001	1033961	31.567952697708794	68.13590539541758	3.6019687308009742
23	00010000110100010000	68880	3.036363636363636	11.072727272727272	0.5853538918957863
24	01111010011000010011	501267	15.819423503325943	36.638847006651886	1.9368933381699311
25	10011111110001000110	654406	20.346814486326682	45.693628972653364	2.415569614891887
26	01111110100100100110	518438	16.327065779748708	37.654131559497415	1.9905658209965964
27	11001011011111010111	833495	25.64138950480414	56.28277900960828	2.97535945106594
28	11110001000010100010	987298	30.18841093865484	65.37682187730968	3.4561112346655904
29	01111110011000110111	517687	16.304863266814486	37.60972653362897	1.988218372705709
30	01110110111111111100	487420	15.410051736881005	35.82010347376201	1.8936108927852222
31	11111100101100111101	1035069	31.60070953436807	68.20141906873614	3.6054320766149606
32	00101011001100100101	176933	6.230835181079083	17.461670362158166	0.923102001316884
33	00110000001011010001	197329	6.833821138211382	18.667642276422765	0.9868550710118736
34	11110011110010010101	998549	30.521034737620102	66.0420694752402	3.491279198341719
35	00110011111011010001	212689	7.2879231337767925	19.575846267553587	1.0348667963754272
36	10010100111100100010	610082	19.03642276422764	43.07284552845528	2.277023279278986
37	11100000000110110101	917941	28.137945306725793	61.27589061345159	3.239317664610904
38	11001101110011110111	842999	25.922365114560236	56.84473022912047	3.0050667061346386
39	10101101111010010001	712337	22.059482631189947	49.118965262379895	2.5966482126806905
40	10100101001010100000	676512	21.000354767184035	47.00070953436807	2.4846677399496153
41	01100100110110110110	413110	13.213155949741315	31.426311899948263	1.6613354167588115
42	01111111111111010111	524247	16.49880266075388	37.99760532150776	2.00872338041306
43	00001011011111001010	47050	2.390983000739098	9.781966001478196	0.5171184775281837
44	01111000100001011001	493657	15.594441980783444	36.18888396156689	1.9131062789240558
45	10111101101001100101	776805	23.96541019955654	52.93082039911308	2.7981599256193346
46	10000111111110001000	556936	17.46521803399852	39.93043606799704	2.1109014592157425
47	10111010110111010110	765398	23.628174427198818	52.256348854397636	2.762504342857482
48	10010011100110110010	604594	18.87417590539542	42.74835181079084	2.2598690899043

GRAFICA



Grafica 1. Datos iniciales

Selección torneo [20]

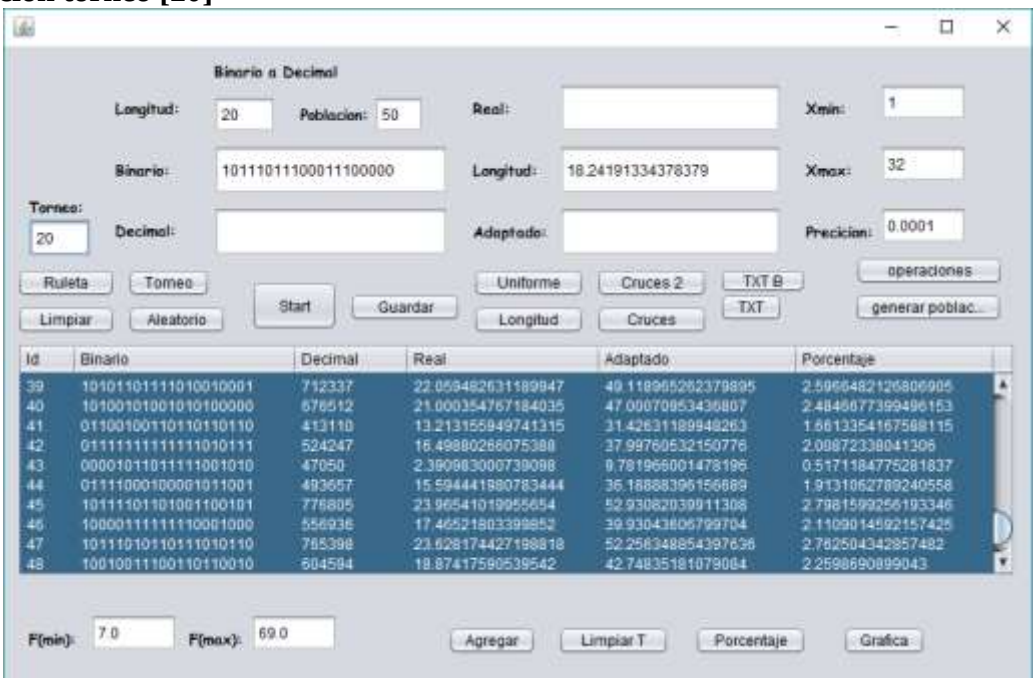


Figura 5. Selección por torneo

Con selección torneo

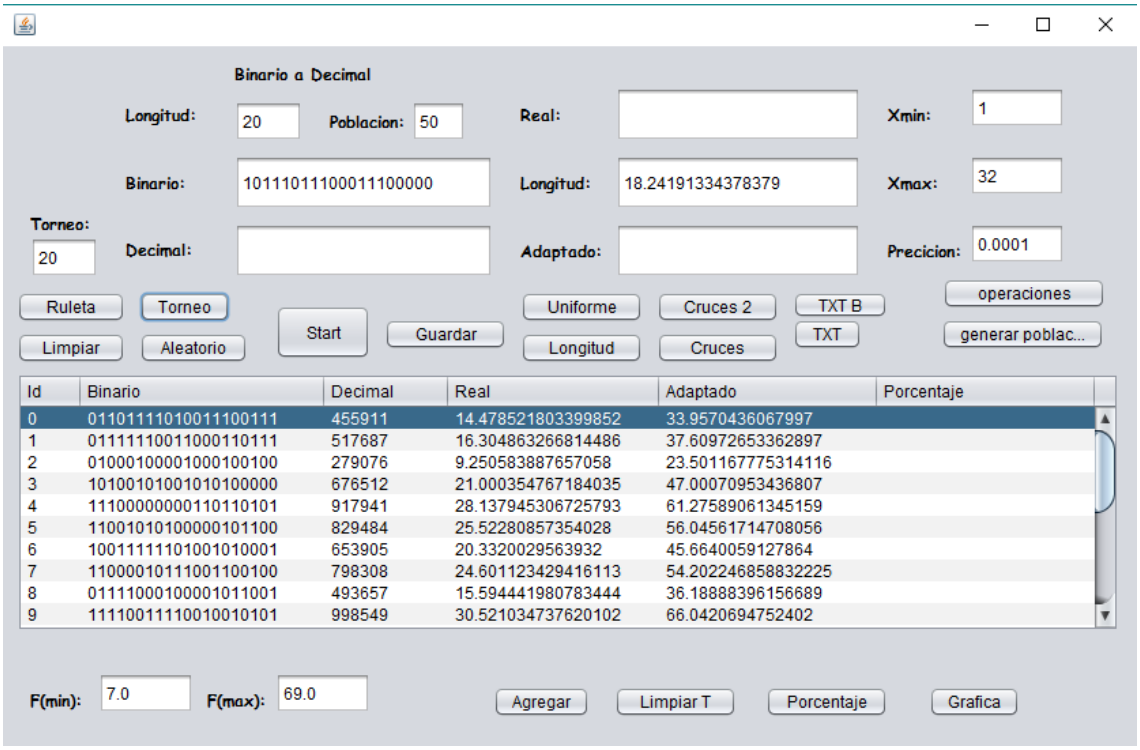
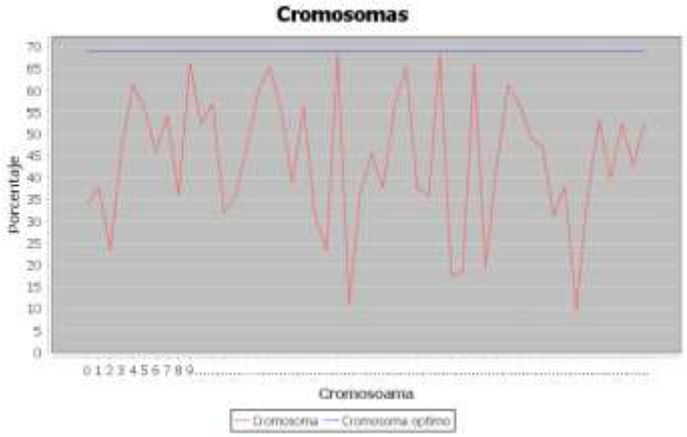


Figura 6. Selección por torneo

Datos

0	01101111010011100111	455911	14.478521803399852	33.9570436067997	3.530786565761609
1	01111110011000110111	517687	16.304863266814486	37.60972653362897	3.9105853479045005
2	01000100001000100100	279076	9.250583887657058	23.501167775314116	2.4436051742788627
3	10100101001010100000	676512	21.000354767184035	47.00070953436807	4.887041278587042
4	11100000000110110101	917941	28.137945306725793	61.27589061345159	6.371346513208515
5	11001010100000101100	829484	25.52280857354028	56.04561714708056	5.827512971509251
6	10011111101001010001	653905	20.3320029563932	45.6640059127864	4.748053466687529
7	11000010111001100100	798308	24.601123429416113	54.202246858832225	5.635842956744831
8	01111000100001011001	493657	15.594441980783444	36.18888396156689	3.762848933521995
9	11110011110010010101	998549	30.521034737620102	66.0420694752402	6.866924411275306
10	10111011100011100000	768224	23.71172209903917	52.42344419807834	5.450886556816271
11	11001101110011110111	842999	25.922365114560236	56.84473022912047	5.910603177101409
12	01100111100111101101	424429	13.547790096082778	32.095580192165556	3.337235263323405
13	01110110111111111100	487420	15.410051736881005	35.82010347376201	3.72450386417103
14	10100101001010100000	676512	21.000354767184035	47.00070953436807	4.887041278587042
15	11011001010101001101	890189	27.317487065779748	59.634974131559495	6.200727246793895
16	11110001000010100010	987298	30.18841093865484	65.37682187730968	6.797753275269582
17	11001010100000101100	829484	25.52280857354028	56.04561714708056	5.827512971509251
18	10000011000000110001	536625	16.864745011086477	38.72949002217295	4.0270161517154
19	11001011011111010111	833495	25.64138950480414	56.28277900960828	5.852172595233272



Grafica 2. Selección torneo

Cruce de un punto

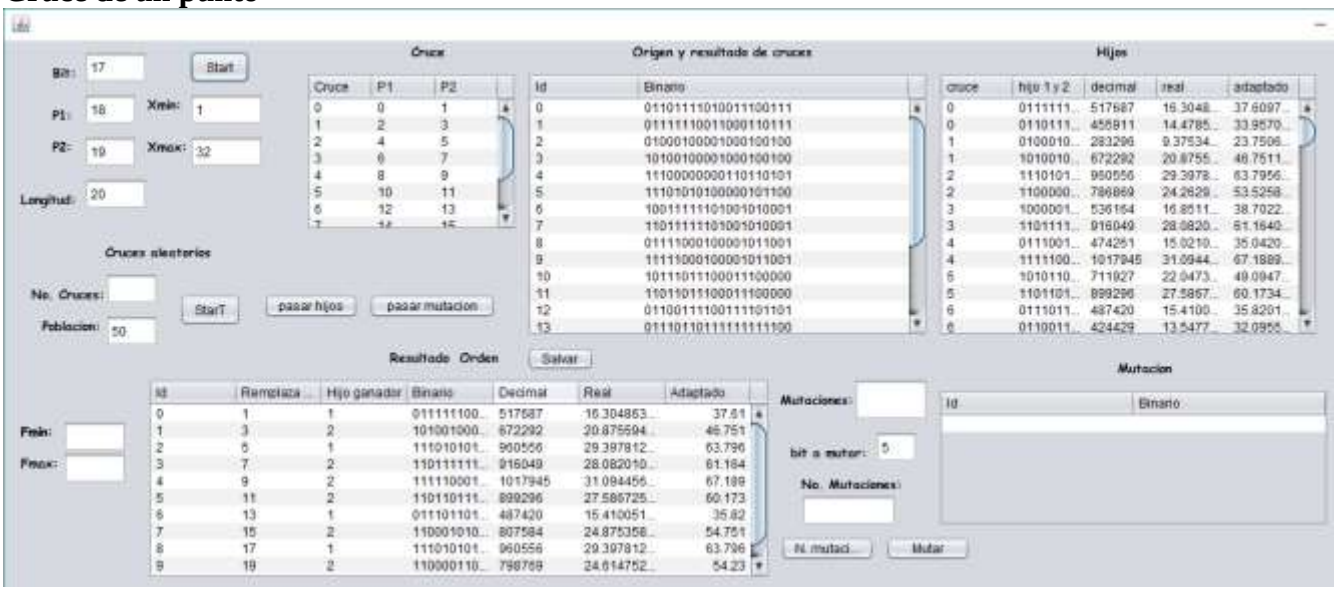


Figura 7. Cruce de un punto

Cruces realizados

id	Reemplaza a padre	Hijo ganador	Binario	Decimal	Real	Adaptado
0	1	1	01111110011000110111	517687	16.304863266814486	37.60972653362897
1	3	2	10100100001000100100	672292	20.87559497413156	46.75118994826312
2	5	1	11101010100000101100	960556	29.397812269031782	63.795624538063564
3	7	2	11011111101001010001	916049	28.0820103473762	61.1640206947524
4	9	2	11111000100001011001	1017945	31.094456762749445	67.18891352549889
5	11	2	11011011100011100000	899296	27.586725794530672	60.173451589061344
6	13	1	01110110111111111100	487420	15.410051736881005	35.82010347376201
7	15	2	11000101001010100000	807584	24.875358462675536	54.75071692535107
8	17	1	11101010100000101100	960556	29.397812269031782	63.795624538063564
9	19	2	11000011000000110001	798769	24.614752402069474	54.22950480413895

Cruce de dos puntos

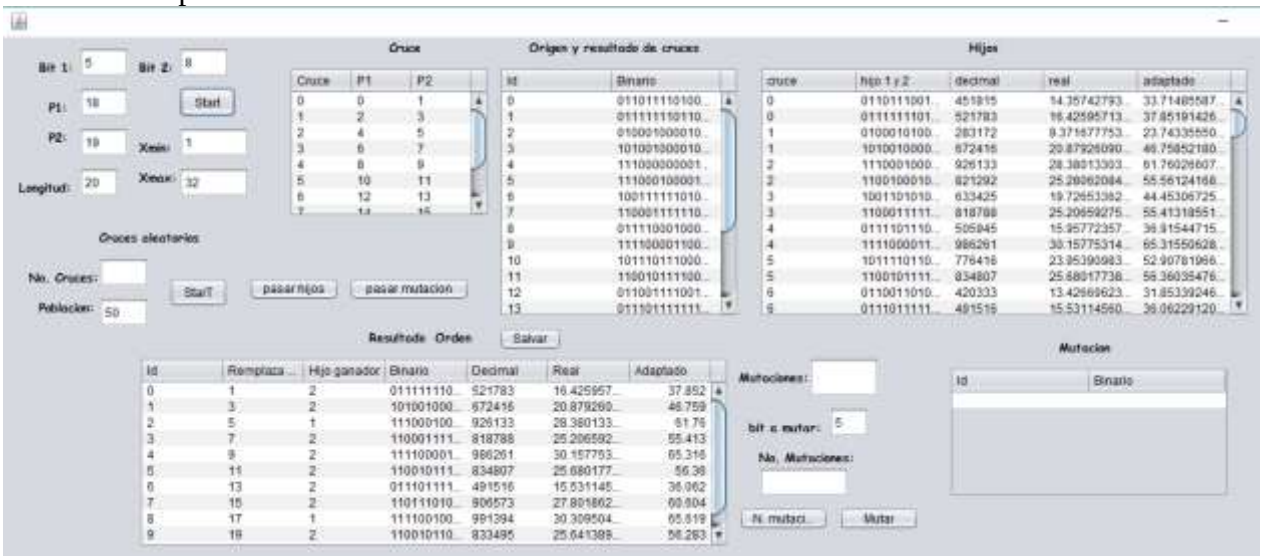


Figura 8. Cruce de dos puntos

Resultados

id	Reemplaza a padre	Hijo ganador	Binario	Decimal	Real	Adaptado
0	1	2	01111111011000110111	521783	16.425957132298596	37.85191426459719
1	3	2	10100100001010100000	672416	20.879260901699926	46.75852180339985
2	5	1	11100010000110110101	926133	28.380133037694012	61.760266075388024
3	7	2	11000111111001100100	818788	25.20659275683666	55.413185513673332
4	9	2	11110000110010010101	986261	30.157753141167774	65.31550628233555
5	11	2	11001011110011110111	834807	25.680177383592017	56.360354767184035
6	13	2	01110111111111111100	491516	15.531145602365115	36.06229120473023
7	15	2	11011101010101001101	906573	27.801862527716185	60.60372505543237
8	17	1	11110010000010100010	991394	30.30950480413895	65.6190096082779
9	19	2	11001011011111010111	833495	25.64138950480414	56.28277900960828

Cruce uniforme

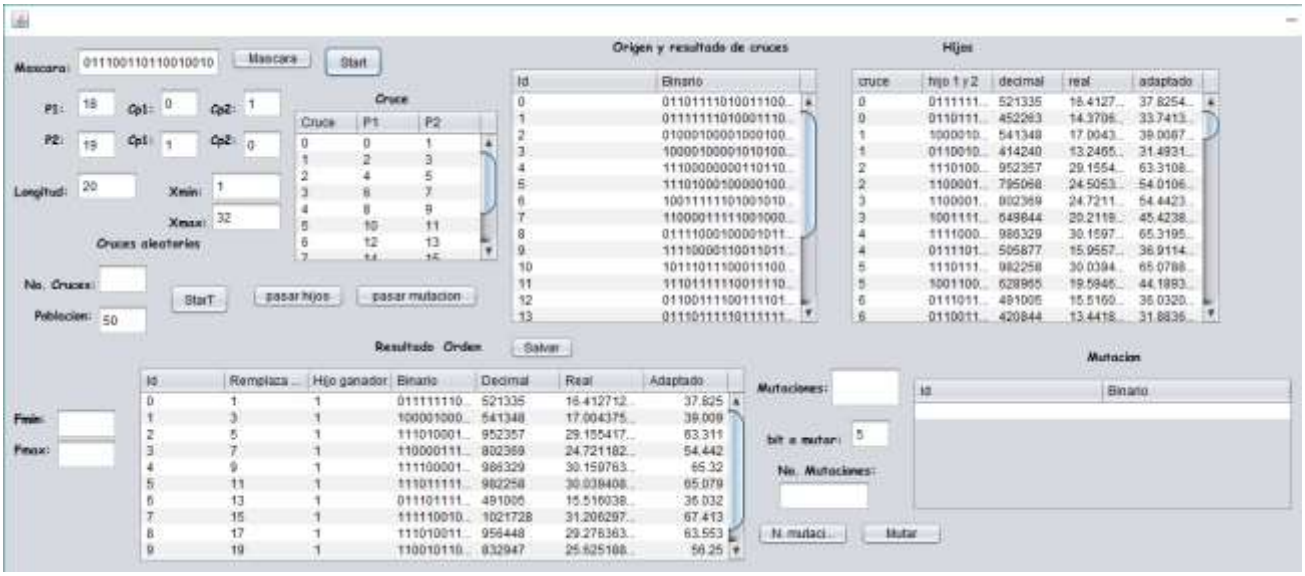


Figura 9. Cruce uniforme

Resultados

id	Reemplaza a padre	Hijo ganador	Binario	Decimal	Real	Adaptado
0	1	1	01111111010001110111	521335	16.412712490761272	37.825424981522545
1	3	1	10000100001010100100	541348	17.004375461936437	39.008750923872874
2	5	1	11101000100000100101	952357	29.155417590539543	63.310835181079085
3	7	1	11000011111001000001	802369	24.72118255728012	54.44236511456024
4	9	1	11110000110011011001	986329	30.159763488543977	65.31952697708795
5	11	1	11101111110011110010	982258	30.03940872135994	65.07881744271988
6	13	1	01110111110111111101	491005	15.516038433111603	36.03207686622321
7	15	1	11111001011100100000	1021728	31.20629711751663	67.41259423503325
8	17	1	11101001100000100000	956448	29.276363636363637	63.552727272727275
9	19	1	11001011010110110011	832947	25.62518847006652	56.25037694013304

Deberá diseñar su propio programa y mostrar pantallas de la ejecución correspondiente. Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

BIBLIOGRAFÍA

Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones,
   ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. PRÁCTICAl Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3. Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4. Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.


ESTADOS UNIDOS MEXICANOS
UNIVERSIDAD AUTONOMA
PATRIA • CIENCIA Y TRABAJO