RECIBIDO EL 30 DE OCTUBRE DE 2019 - ACEPTADO EL 30 DE ENERO DE 2020

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 5 5 6 - 1 5 3 6

IMPLEMENTACIÓN DE UN ALGORITMO GENÉTICO MEDIANTE UNA APLICACIÓN INFORMÁTICA BASADO EN LA COMPUTACIÓN NEURONAL Y EVOLUTIVA PARA OBTENER EL CROMOSOMA MEJOR ADAPTADO

IMPLEMENTATION OF A GENETIC ALGORITHM USING A COMPUTER APPLICATION BASED ON NEURAL AND EVOLUTIONARY COMPUTING TO OBTAIN THE BEST ADAPTED CHROMOSOME

María de los Ángeles Rodríguez Cevallos[1] **María José Andrade-Albán**[2]

Roberto Carlos Maldonado Palacios[3] **Cristhian Alfonso Cobos-Cevallos**[4]

Ecuador

1 ORCID https://orcid.org/0000-0001-8409-0530 Magister en Nutrición Clínica, maria.rodriguez@espoch.edu.ec, (593)992521013, Escuela Superior Politécnica de Chimborazo (ESPOCH), Riobamba, Ecuador.

2 ORCID https://orcid.org/0000-0002-5874-4390 Magister en Gestión de la Producción Agroindustrial, maria.andrade@espoch.edu.ec, (593)984873311, Escuela Superior Politécnica de Chimborazo (ESPOCH), Riobamba, Ecuador.

3 ORCID https://orcid.org/0000-0002-8902-7490 Magister en Redes de Comunicaciones, robertomaldonadop@hotmail.com, (593)984068875, Pontificia Universidad Católica del Ecuador (PUCE), Quito, Ecuador.

4 ORCID https://orcid.org/0000-0003-3290-6280 Estudiante Ingeniería en Sistemas y Computación, criscob_41@hotmail.com, (593)992739741, Universidad Nacional De Chimborazo (UNACH), Riobamba, Ecuador.

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

RESUMEN

El objetivo de la investigación es implementar un algoritmo genético mediante una aplicación informática basado en la computación neuronal y evolutiva para obtener el cromosoma mejor adaptado, dicho desarrollo demanda del análisis estadístico decriptivo basado en algoritmos genéticos. Sé utilizó específicamente veinte y uno tipos de cromosomas que lo datos de ingreso serán una red en formato Pajek (* .net) y los datos de salida mencionará la partición de modularidad más alta encontrada, en formato Pajek (*.clu), y su valor correspondiente de modularidad, es decir después de la selección del cromosoma, del cruce o la mutación se realiza la evaluación para la decodificación por medio del parámetro Aptitud, seleccionando de ésta manera por medio de la Rueda de la Ruleta para obtener el cromosoma mejor adaptado.

Se utilizó específicamente veinte y uno tipos de cromosomas transformadas en formato Pajek (.net) y los algoritmos genéticos (AG) funcionan entre el conjunto de soluciones de un problema llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que forman la cadena son llamados genes. Cuando la representación de los cromosomas se hace con cadenas de dígitos binarios se le conoce como genotipo. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados mediante la computación neuronal y evolutiva usando alguna medida de aptitud. Las siguientes generaciones (nuevos cromosomas), son generadas aplicando los operadores genéticos repetidamente, siendo estos los operadores de selección, cruzamiento, mutación y reemplazo, para lograr obtener el cromosoma mejor adaptado.

En este artículo se explica la implementación de un algoritmo genético mediante una aplicación informática basado en la computación neuronal y evolutiva para obtener el cromosoma mejor adaptado, que parte de una población de soluciones, y en base al valor de la función de adaptación para cada uno de los individuos (soluciones) de esa población, se seleccionan los mejores individuos (según dicha función) y se combinan para generar otros nuevos. Este proceso se repite cíclicamente hasta que se cumple un criterio de parada.

PALABRAS RESERVADAS:

algoritmo genético, computación neuronal y evolutiva, cromosoma.

ABSTRACT:

The aim of the research is to implement a genetic algorithm using a computer application based on neural and evolutionary computing to obtain the best adapted chromosome, such development demands statistical analysis of decriptive based on genetic algorithms. I know specifically used twenty types of chromosomes that the input data will be a network in Pajek format (*.net) and the output data will mention the highest modularity partition found, in Pajek format (*.clu), and its corresponding modularity value, i.e. after chromosome selection, crossing or mutation is performed the evaluation for decoding by means of the Fitness parameter , thus selecting through the Wheel of roulette to obtain the best-adapted chromosome.

Twenty types of chromosomes transformed into Pajek (.net) format were specifically used and genetic algorithms (AG) work between the solution set of a problem called phenotype, and the set of individuals from a natural population, encoding the information of each solution into a string, usually binary, called a chromosome. The symbols that make up the

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

chain are called genes. When the chromosome representation is done with binary digit strings it is known as a genotype. Chromosomes evolve through iterations, called generations. In each generation, chromosomes are evaluated using neural and evolutionary computing using some measure of fitness. The following generations (new chromosomes), are generated by applying the genetic operators repeatedly, these being the operators of selection, crossing, mutation and replacement, to obtain the best adapted chromosome.

This article explains the implementation of a genetic algorithm using a computer application based on neural and evolutionary computing to obtain the best adapted chromosome, which starts from a population of solutions, and based on the value of the adaptation function for each of the individuals (solutions) of that population, select the best individuals (according to that function) and combine to generate new ones. This process is repeated cyclically until a stop criterion is met.

KEY WORDS:

genetic algorithm, neural and evolutionary computing, chromosome.

1. INTRODUCCIÓN:

Los Algoritmos Genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio de supervivencia del más apto (Fogel, 2000) (Fogel, 2006).

Más formalmente, y siguiendo la definición dada por Goldberg, “los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas” (Goldberg, 1989).

Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida (Darwin, 2007).

1.1 Justificación/Problema

La falta de estudio de algoritmos genéticos de estimación en el campo de la optimización combinatoria para obtener el cromosoma mejor adaptado incita la necesidad de implementar un sistema informático exclusivamente aplicado en población humana. Un algoritmo genético puede presentar diversas variaciones, dependiendo de cómo se aplican los operadores genéticos (cruzamiento, mutación) y de cómo se realiza la selección para continuar con el reemplazo de los individuos para formar la nueva población.

1.2 Revisión de la literatura

Los algoritmos genéticos son una técnica de resolución de problemas inspirada en la naturaleza. Están basados en el principio darwiniano de reproducción y supervivencia de los individuos más aptos [Beasley et alt, 1993]

La computación evolutiva es la rama de la inteligencia artificial que engloba a todas aquellas técnicas de resolución de problemas basadas en la evolución de las especies y la supervivencia del más apto [Joglekar y Tungare, 2001].

Dentro de ella encontramos a los algoritmos genéticos (genetic algorithms), las estrategias

REVISTA BOLETÍN REDIPE 9 (8): 116-131 - AGOSTO 2020 - ISSN 2256-1536

evolutivas (evolution strategies) y la programación evolutiva (evolutionary programming) entre otros [Yao, 1996].

Las técnicas evolutivas han sido aplicadas con éxito a distintos tipos de problemas como optimización de parámetros, planificación de tareas, diseño, etc. [Whitley, 2002].

Estos algoritmos codifican las posibles soluciones en estructuras llamadas cromosomas (o individuos). A un conjunto de individuos se lo conoce como población y representan un conjunto de soluciones posibles al problema [Cantú-Paz, 1997]. Mediante la aplicación de un conjunto de operadores genéticos sobre la población se va refinando gradualmente la solución hasta alcanzar un resultado que cumpla con las condiciones requeridas.

El primer paso en la aplicación de un algoritmo genético consiste en la generación de una población inicial. En general esta población se genera de manera aleatoria, y el tamaño de dicha población (la cantidad de individuos que la compone) es un parámetro que se define durante el diseño del algoritmo genético. Una vez generada esta población se debe evaluar la aptitud (fitness) de cada individuo [Deb, 2004].

El operador de selección es el encargado de decidir cuales individuos contribuirán en la formación de la próxima generación de individuos. Este mecanismo simula el proceso de selección natural, mediante el cual sólo los individuos más adaptados al ambiente se reproducen [Coello Coello, 2002]. El mecanismo de selección forma una población intermedia, que está compuesta por los individuos con mayor aptitud de la generación actual.

La siguiente fase del algoritmo consiste en la aplicación de los operadores genéticos. El primero de ellos es la cruza, y su función es recombinar el material genético. Se toman aleatoriamente dos individuos que hayan sobrevivido al proceso de selección y se recombina su material genético creando uno o más descendientes, que pasan a la siguiente población. Este operador se aplica tantas veces como sea necesario para formar la nueva población.

El último paso consiste en la aplicación del operador de mutación. Este operador, que en general actúa con muy baja probabilidad, modifica algunos genes del cromosoma, posibilitando de esta manera la búsqueda de soluciones alternativas.

Una vez finalizado el proceso de selección, cruza y mutación se obtiene la siguiente generación del algoritmo, la cual será evaluada, repitiéndose el ciclo descripto previamente. Tras cada iteración la calidad de la solución generalmente va incrementándose, y los individuos representan mejores soluciones al problema [Forrest 1996].

Al algoritmo genético detallado anteriormente se lo conoce como algoritmo genético canónico y es la forma más utilizada. Sin embargo, en algunas implementaciones particulares de algoritmos genéticos se puede agregar nuevos operadores. En todos los casos, la forma en que se implementen los operadores variará de acuerdo con las características propias del problema.

EL CROMOSOMA

Los algoritmos genéticos trabajan manipulando cromosomas, que son estructuras que codifican las distintas soluciones de un determinado problema.

La forma en que se codifican los cromosomas es dependiente de cada problema en particular, y suele variar de problema en problema.

Un cromosoma está compuesto por un conjunto de genes. Cada gen representa una característica particular del individuo y ocupa una posición determinada en el cromosoma,

REVISTA BOLETÍN REDIPE 9 (8): 116-131 - AGOSTO 2020 - ISSN 2256-1536

llamada locus. A cada uno de los valores que puede tomar un gen se lo conoce como alelo. [Ilachinski, 1997]

Es importante destacar dos conceptos que muchas veces suelen confundirse: genotipo y fenotipo. El genotipo es el conjunto de genes de un individuo, la descripción genética del individuo. El fenotipo es la forma en que se expresa el genotipo, como resultado de la interacción con su entorno [Morrison, 1998]. La morfogénesis es el proceso de decodificar el genotipo para producir un fenotipo [Biondi y Michel, 1995].

Una cuestión para resolver cuando se diseñan algoritmos evolutivos es la forma de codificar los genes. Estos se pueden representar como cadenas binarias, números enteros, números reales, etc. [Whitley, 2001]. En general se utilizan las cadenas binarias por varios motivos:

- Cualquier parámetro se puede codificar como una cadena de bits. Una cadena de bits permite expresar valores booleanos, enteros, reales, etc.

- La codificación binaria se puede ajustar a cualquier problema. Por este motivo la mayor parte de los trabajos teóricos se basa en este esquema de codificación. El uso de cadenas de bits permite contar con gran cantidad de herramientas de análisis.

- Las cadenas de bits son fáciles de manipular. El desarrollo de operadores genéticos es simple y eficiente.

Sin embargo, las características propias del problema podrían llevar a la utilización de otro esquema de codificación. En dicho caso será necesario desarrollar operadores genéticos que se adapten al esquema seleccionado.

En lo que sigue de este capítulo se supone el uso de cadenas binarias, debido a que estas son más fáciles de comprender.

EVALUACIÓN Y APTITUD

El uso de los conceptos de función de evaluación y función de aptitud se suelen utilizar como sinónimos. Sin embargo, ambos conceptos son diferentes, y es conveniente hacer una distinción entre ellos.

La función de evaluación, o función objetivo, provee una medida de la performance de conjunto de parámetros. Indica que tan buena es la solución obtenida tras la aplicación de un conjunto de parámetros, independientemente de la aplicación de esta función sobre otro conjunto de parámetros [Whiltey, 1994]. En general, esta medida se obtiene tras la decodificación de un cromosoma en el conjunto de parámetros que representa.

Por su parte, la función de adaptación siempre está definida con respecto al resto de los individuos de la población. Indica que tan buena (o mala) es una solución comparada con el resto de las soluciones obtenidas hasta el momento. El valor obtenido por esta función se traduce, tras la aplicación del operador de selección, en oportunidades de reproducción. Aquellos que tengan mayor aptitud tendrán mayores posibilidades de reproducirse y transmitir su material genético a la próxima generación.

En el algoritmo genético canónico, la función de adaptación se define (Ecuación 1.)

$$F_i = \frac{f_i}{\bar{f}} \qquad \bar{f} = \frac{\sum_{i=1}^{N} f_i}{N}$$

$$F_i = \frac{f_i}{\bar{f}} \qquad \bar{f} = \frac{\sum_{i=1}^{N} f_i}{N}$$

Ecuación 1. Ecuación de la Función de Adaptación

donde f_i es el valor de la función de evaluación para el cromosoma i

REVISTA BOLETÍN REDIPE 9 (8): 116-131 - AGOSTO 2020 - ISSN 2256-1536

POBLACIÓN

Los algoritmos genéticos trabajan sobre una población de individuos [Bäck,1992], donde cada uno de los cuales representa una posible solución. La población es un concepto muy importante en los algoritmos genéticos y existen dos cuestiones fundamentales a resolver: como generar la primera población, y cuál debe ser el tamaño de la población.

La población inicial debe poseer la mayor diversidad posible. Idealmente, la población inicial debería contener todos los posibles valores (alelos) que pueda tomar un gen. De esta manera se aseguraría una exploración completa del espacio de búsqueda. En la práctica, esta situación no suele ser factible por lo que la primera población se genera comúnmente de manera azarosa, asignándole aleatoriamente a cada gen uno de los posibles alelos. De esta manera se asegura la diversidad de alelo por gen. En algunos casos se pueden utilizar heurísticas para la generación de la población inicial, las que permiten obtener un punto de partida más próximo a la solución del problema. En estos casos, es importante que la heurística asegure la diversidad mencionada. De lo contrario el algoritmo genético no será capaz de realizar una búsqueda completa sobre el espacio de soluciones.

El tamaño de la población no es problema crítico. Si el tamaño de la población es muy grande se realizará una exploración del espacio de búsqueda más rápida en términos de generaciones. Sin embargo, el tiempo necesario para obtener una nueva generación a partir de la generaron actual será mayor, dado que se estarán procesando individuos innecesarios. Si el tamaño de la población es muy chico podría llegar a darse el caso en que la población converja rápidamente a una solución que no sea lo suficientemente buena. En la práctica se suelen utilizar poblaciones de 20 individuos, pudiéndose aumentar el mismo en base a la complejidad del problema o a la calidad de la solución requerida. En algunos casos se utiliza un tamaño de población inicial, el cual se va incrementando a medida que la población tiende a ser homogénea.

VARIANTES DE LOS OPERADORES BÁSICOS

El objetivo de cada uno de los tres operadores básicos está bien definido: elegir los individuos de la población que tengan mayor grado de adaptación (selección), recombinar el material genético de ellos para producir nuevos individuos (cruza) y alterar características de algunos de ellos para garantizar a la diversidad (mutación).

Siguiendo estos objetivos se han desarrollado gran cantidad de variantes para cada uno de los operadores. Cada variante tiene características particulares que afectaran el comportamiento del algoritmo genético. Una elección adecuada para cada operador puede influir decisivamente en la eficiencia del proceso de búsqueda.

A continuación, se realiza una mención sobre las variantes más utilizadas de cada uno de los operadores.

SELECCIÓN

SELECCIÓN POR RULETA

El primer paso para la aplicación de este operador consiste en obtener aleatoriamente un número R de la siguiente manera (Ecuación 2.):

$$R = Random(0, \sum_{i=1}^{N} F_i)$$

Ecuación 2. Ecuación Randómica de la Ruleta

Luego se selecciona al individuo j tal que (Ecuación 3.):

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

$$\sum_{i=1}^{j} F_i \leq R < \sum_{i=1}^{j+1} F_i$$

Ecuación 3. Ecuación para la siguiente generación a obtener

Este individuo pasa a la siguiente generación, repitiéndose el procedimiento tantas veces como sea necesario hasta completar la siguiente generación. Al contener una componente aleatoria, la cantidad de copias de un individuo que efectivamente pasan a la siguiente generación puede variar de la cantidad de copias esperadas.

SELECCIÓN PROPORCIONAL

Este operador asigna un número de copias proporcional a la aptitud de un individuo. El número de copias del individuo i que pasarán a la siguiente población se calcula en base a la siguiente ecuación (Ecuación 4.):

$$c_i = N * \frac{f_i}{\sum_{i=1}^{N} F_i}$$

Ecuación 4. Ecuación de Copias proporcional a la Aptitud

SELECCIÓN POR TORNEO

Se seleccionan aleatoriamente dos individuos de la población, pasando a la siguiente generación aquel individuo que tiene mayor aptitud. Este proceso se repite N veces, donde N es el tamaño de la población. La selección por torneo no asegura que el número de copias que pasan a la próxima generación sea igual al esperado. Este operador se puede implementar tomando cualquier número de individuos.

SELECCIÓN POR RANKING

Este operador asigna una probabilidad de selección proporcional a la aptitud del individuo. Sin embargo, la selección se realiza con un esquema análogo al de selección por ruleta. Se asigna a un individuo una probabilidad de selección igual a (Ecuación 5.):

$$p_i = \frac{N - i}{\sum_{i=1}^{N} i}$$

Ecuación 5. Ecuación de la Selección por Ranking

Luego se selecciona un número aleatorio R de la siguiente manera (Ecuación 6.):

$$R = Random(0, \sum_{i=1}^{N} p_i)$$

Ecuación 6. Ecuación de la Función de Adaptación aleatorio

Por **último,** se elige al individuo j tal que (Ecuación 7.):

$$\sum_{i=1}^{j} p_i \leq R < \sum_{i=1}^{j+1} p_i$$

Ecuación 7. Ecuación de la Función de Adaptación de la siguiente generación

Este individuo pasa a la siguiente generación. El procedimiento se repite tantas veces como sea necesario hasta completar la siguiente generación.

Nuevamente, la cantidad de copias que se pasan a la siguiente generación puede diferir de la cantidad esperada con un esquema proporcional.

CRUZA

CRUZA UNIFORME

Este operador de cruza asigna aleatoriamente los pesos. Cada hijo tiene una probabilidad de 0.5 de recibir los genes de su padre y por ende de recibirlos de su madre.

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

CRUZA SIMPLE

En esta variante del operador de cruza se toma aleatoriamente un punto de cruza. Luego, a un hijo se le asignan todos los genes del padre ubicados a la izquierda del punto de cruza, y todos los genes de la madre ubicados a la derecha del punto de cruza. El segundo hijo es el complemento del primero. La Figura 1 muestra un ejemplo de cruza simple en el cual se toma como punto de cruza el primero de ellos.

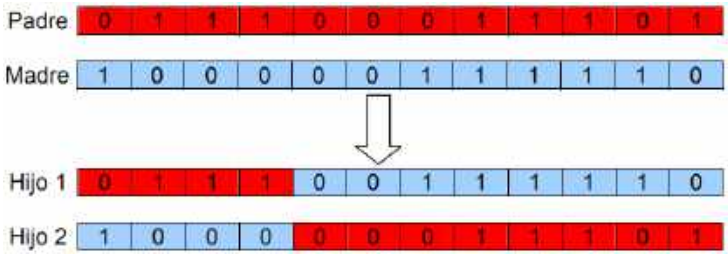

Figura 1. Cruza Simple

CRUZA MULTIPUNTO

Esta variante del operador de cruza es similar al operador de cruza simple, sólo que la cantidad de puntos de cruza se determina aleatoriamente. Cada hijo recibe los genes entre dos puntos de cruza sucesivos de cada uno de los padres, de manera intercalada. Cabe destacar que la cruza simple es un caso particular de la cruza multipunto, con un único punto de cruza. La Figura 2, muestra un ejemplo de cruza multipunto, tomando dos puntos de cruza.

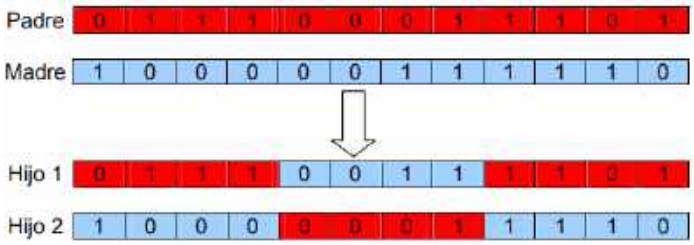

Figura 2. Cruza Multipunto

CRUZA BINOMIAL

Este operador es similar al operador de cruza uniforme, sólo que las probabilidades se definen en función de la aptitud de los padres de la siguiente manera (Ecuación 8.):

$$\begin{cases} p_{padre} = \dfrac{f_{padre}}{f_{padre} + f_{madre}} \\ p_{madre} = 1 - p_{padre} \end{cases}$$

Ecuación 8. Ecuación de la Cruza Binomial

MUTACIÓN

La mutación binaria es el tipo de mutación tradicionalmente utilizado en el algoritmo genético canónico, y consiste en invertir un gen aleatoriamente. Dada la representación binaria del gen, se invierte el bit que lo representa con una determinada probabilidad, como lo muestra la Figura 3.

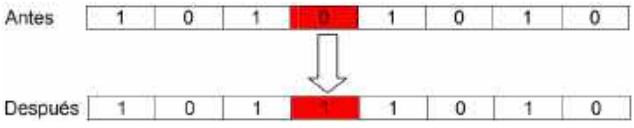

Figura 3. Mutación Binaria

REVISTA BOLETÍN REDIPE 9 (8): 116-131 - AGOSTO 2020 - ISSN 2256-1536

MUTACIÓN SIMPLE

En este caso la probabilidad de mutación permanece constante a lo largo de las distintas generaciones, como lo muestra la Ecuación 9.

$$P_m(t) = P_m(0) = P_m^{\max} = P_m^{\min}$$

Ecuación 9. Ecuación Mutación Simple

MUTACIÓN ADAPTATIVA POR TEMPERATURA ASCENDENTE

La probabilidad de mutación se va incrementando a medida que transcurren las generaciones. El objetivo de este aumento es mantener la diversidad de individuos en la población, que tiende a hacerse homogénea con el transcurso de las generaciones. La probabilidad de mutación para una generación está dada por la Ecuación 10.

$$P_m(t) = P_m^{\min} + \frac{(P_m^{\max} - P_m^{\min})}{T} * t$$

Ecuación 10. Ecuación Mutación adaptativa por temperatura ascendente

MUTACIÓN ADAPTATIVA POR TEMPERATURA DESCENDENTE

La probabilidad de mutación va decreciendo a medida que transcurren las generaciones. De esta manera se asegura una alta diversidad de individuos en las generaciones iniciales. La probabilidad mínima debe ser mayor a cero para permitir continuar la exploración del espacio de búsqueda a medida que las generaciones avanzan. La Ecuación 11 muestra cómo se calcula la probabilidad de mutación para una determinada generación.

$$P_m(t) = P_m^{\max} - \frac{(P_m^{\max} - P_m^{\min})}{T} * t$$

Ecuación 11. Ecuación Mutación adaptativa por temperatura descendente

1.3 PROPÓSITO

Implementar un algoritmo genético mediante una aplicación informática basado en la computación neuronal y evolutiva para obtener el cromosoma mejor adaptado.

2. MÉTODO

TIPOS DE INVESTIGACIÓN

Investigación documental: Búsquedas en diversas fuentes de información como son: bases de datos digitales, libros, revistas, manuales, internet, entre otros**.**

MÉTODO

Científico: Es un estudio sistemático, lógico y organizado de la proposición planteada para adquirir conocimientos y brindar una solución.

Descriptivo: Se realizó un estudio descriptivo que consistió en llevar a conocer situaciones relevantes a través de la descripción de las variables de investigación para exponer de manera cuidadosa los resultados a fin de extraer generalizaciones significativas.

2.1 INSTRUMENTOS Y MATERIALES

- Lenguaje de Programación Java.
- Entorno de desarrollo integrado libre Netbeans.
- Microsoft Office Excel 2016.
- Sistema Operativo Windows 10 Professional.
- Paquete Pajek64

2.2 PROCEDIMIENTO

Los datos del muestreo que se usaron son 20 redes en formato Pajek (.net), los mismos que para su manipulación y procesamiento se realiza la **Modularidad,** que G sea una red no dirigida (gráfico) con *L* bordes y *N* vértices, no autobucles y actividades conjuntas de matriz de

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

adyacencia a_{ij}. Si tenemos una partición *C* de los nodos del gráfico en clusters disjuntos, la *modularidad Q* se define como (Ecuación 12.)

$$Q(C) = \frac{1}{2L}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(a_{ij} - \frac{k_i k_j}{2L}\right)\delta(C_i, C_j)$$

Ecuación 12. Ecuación de la Modularidad

donde k_i es el grado (número de bordes) del nodo *i*, y $\delta(C_i, C_j)$ es 1 si los nodos *i* y *j* pertenecen al mismo cluster, 0 de lo contrario. Más precisamente (Ecuación 13.),

$$k_i = \sum_{j=1}^{N} a_{ij}$$

$$2L = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} = \sum_{i=1}^{N} k_i$$

Ecuación 13. Ecuación de la Modularidad mejorada

Modularidad es cero cuando todos los nodos pertenecen al mismo cluster, y en general puede tomar valores en el rango de (Ecuación 14.)

$$-1 \le Q(C) \le 1$$

Ecuación 14. Ecuación de la Modularidad con rangos

La modularidad de una partición dada es entonces la probabilidad de que los bordes caigan dentro de los grupos en la red menos la probabilidad esperada en una red equivalente (caso nulo). Esta red de caso nulo tiene el mismo número de nodos que la red original y los bordes colocados al azar preservando los grados de los nodos. Tener un gran valor de modularidad significa una mayor desviación del caso nulo y, por lo tanto, una mejor partición de la red. Tenga en cuenta que la optimización de la modularidad no se puede realizar mediante una búsqueda exhaustiva, ya que el número de particiones diferentes es igual a los números exponenciales, que crecen al menos exponencialmente en la cantidad de nodos. De hecho, la optimización de la modularidad es un problema NP-difícil.

Solo nos interesan las particiones de los nodos en dos clusters, que llamaremos clusters izquierdo y derecho, respectivamente. Por lo tanto, si definimos S_i como 0 si el nodo *i* pertenece al clúster izquierdo y 1 si pertenece al clúster derecho, entonces la partición en dos clústeres está completamente determinada por el vector *S* (Ecuación 15.)

$$\delta(C_i, C_j) = S_i S_j + (1 - S_i)(1 - S_j)$$

Ecuación 15. Ecuación de la Modularidad particionado

El Algoritmo genético dado la red *G*, cualquier vector *S* se puede ver como el cromosoma correspondiente a una partición en dos clústeres, y la aptitud está relacionada con su modularidad *Q(S)*. El objetivo es la implementación de un algoritmo genético para obtener la partición que maximiza la modularidad.

ENTRENAMIENTO

La modularidad no se puede usar directamente como aptitud, ya que puede tomar valores negativos. Debe definir una transformación de la modularidad en una función adecuada de aptitud positiva.

PARÁMETROS

Pruebe diferentes variantes del algoritmo genético y de sus parámetros correspondientes, por ejemplo:

- Función de la Entrenamiento
- Esquemas de selección: proporcional (rueda de la ruleta), truncamiento,

REVISTA BOLETÍN REDIPE 9 (8): 116-131 - AGOSTO 2020 - ISSN 2256-1536

clasificación, torneo, uniforme de entrenamiento (FUSS), etc.

- Crossover: crossover de un punto, cruce de dos puntos, crossover uniforme.
- Elitismo

Código fuente desarrollado en Java (Figura 5)

La Declaración e inicialización de variables (parámetros): La primera y segunda variable respectivamente representan el NÚMERO DE NODOS y representa el TAMAÑO DE LA RED ES: y éstos pueden cambiar de valor y sirve para las respectivas pruebas. EL VECTOR DE GRADOS ES: la representación de un individuo se puede hacer mediante una codificación discreta, y en particular binaria y me codifica el valor positivo de cada gen del cromosoma ingresado en binario una variable que calcula cuantos bits del cromosoma (Figura 4.).

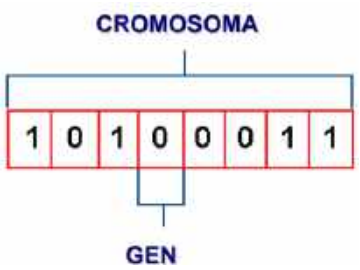

Figura 4. Codificación de un cromosoma

EL TOTAL DEL VECTOR DE GRADOS ES: la sumatoria del vector de grados individual de cada uno de los cromosomas que son los vértices N, EL VALOR DE L es el número de bordes de los cromosomas. Para obtener el valor de la Modularidad Q de cada uno de los cromosomas, para obtener la Modularidad de toda la red.

Para de ésta manera empezar la Inicilización de la Población con la codificación del cromosoma en binario, el cruce de un punto, la mutación, para proceder con la Evaluación (decodificación, soluciones y el Cálculo de la Aptitud) para finalmente llegar a los Esquemas de selección: proporcional (rueda de la ruleta), truncamiento, clasificación, torneo, uniforme de entrenamiento (FUSS)

```java
public class Geneticos_V6 {

    /**
     * @param args the command line arguments
     */
    static int filas;//6
    static int columnas;//5
    static int Nganadores;//3
    static int Ngenes;  //Numero de poblacion 7

    static String [][] Poblacion;

    static String [][] PoblacionTem=new String [filas][columnas];
    static String [] Parejas = new String [filas];
    static String [] Ganadores = new String [Nganadores];
    static double sumaToria=0;

    static int N=31;
    private static StringBuilder num_nodes_str;
    static int L;
    double Q;


    public static void main(String[] args) {
        // TODO code application logic here

        Poblacion = leerFicheroNet();
        //vector K de grados
        int [] vector_k = funcion_Grados(Poblacion);
        System.out.println("El vector de grados es:");
        int sum=0;
```

Figura 5.
Código fuente, declaración de variables

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

2.3 Resultados

Para el análisis de datos se realizó con dos cromosomas de 8 genes cada uno (Figura 6.).

Figura 6. Genes del cromosoma

Los mismos que al realizar la respectiva función de la Entrenamiento, el esquemas de selección proporcional (roulette-wheel), se logra obtener el valor del número de grados de cada gen y el valor total del cromosoma siendo 26, obteniendo la modularidad de 13 (Figura 7.).

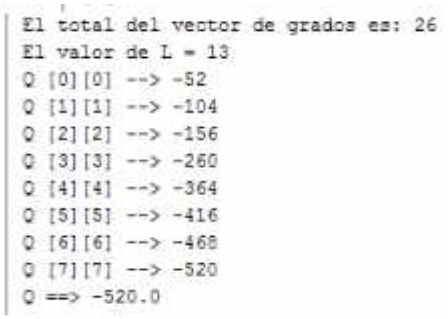

Figura 7. Valores de cada gen

De esta manera el truncamiento, clasificación, torneo, uniforme de entrenamiento (FUSS), y su respectivo crossover en un punto, se logra el elitismo el cromosoma mejor de todos, para de esta lograr obtener el comosoma mejor adaptado 112 y su número de generaciones (Figura 8).

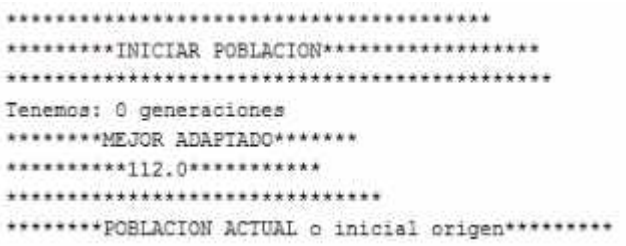

Figura 8. Cromosoma mejor adaptado

CASO PRÁCTICO CON UN CROMOSOMA COMPUESTO DE 8 GENES CONSTANTES

Para el análisis de datos (genes) se realizó un cromosoma con 8 genes constantes (Figura 9.)

Figura 9. Valores constantes de los Genes del Cromosoma

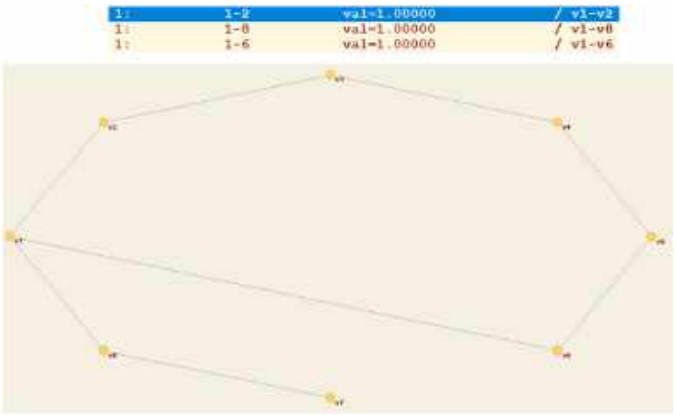

Figura 10. Estructura del Cromosoma con genes iguales

El total de número de líneas se obtuvo 0 arcos y 8 genes (Figura 10.), con la densidad1 de bucles permitidos de 0,25000000; así como una densidad2debuclesnopermitidosde0,28571429; de esta manera la red de centralización de todos los genes es 0,19047619; además se logra obtener la Estadística descriptiva de los genes

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

para su evaluación, siendo la media aritmética y media sea 2,0000; así como su frecuencia total sea del 100% (Figura 11.).

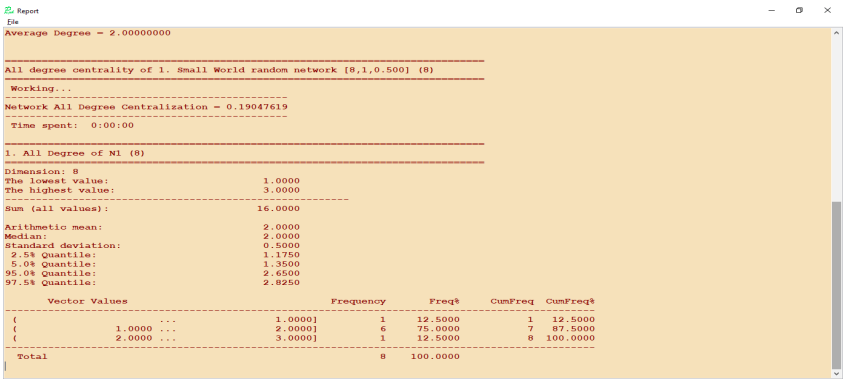

Figura 11. Valores estadísticos del cromosoma de genes iguales

De esta manera logramos obtener centralización de intermediación del cromosoma de 0,40816327 de un cromosoma con 8 genes, con el generador de genes aleatorios de 54 y finalmente obtener la centralización de intermediación del cromosoma 0,04640952 (Figura 12.).

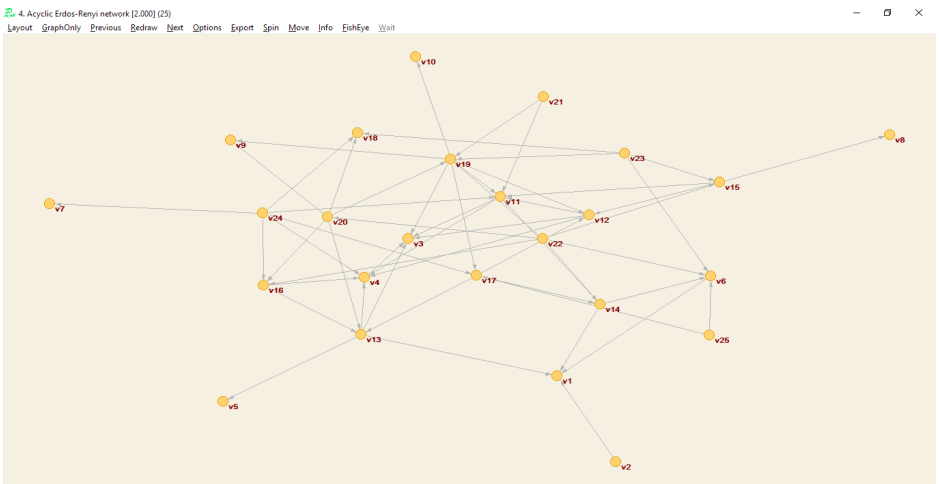

Figura 12. Distribución de los genes iguales en el cromosoma mejorado

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

Para finalmente obtener el cromosoma mejor adaptado, el gen11 con el número de generaciones 112 (Figura 13.).

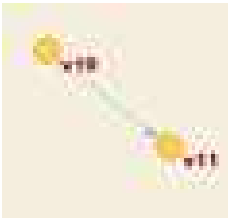

Figura 13. Cromosoma mejor adaptado

CASO PRÁCTICO CON UN CROMOSOMA COMPUESTO DE 8 GENES DISTINTOS

Para el análisis de datos (genes) se realizó un cromosoma con 8 genes diferentes (Figura 14.).

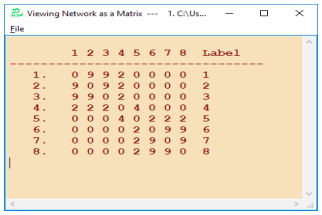

Figura 14. Valores distintos de los Genes del Cromosoma

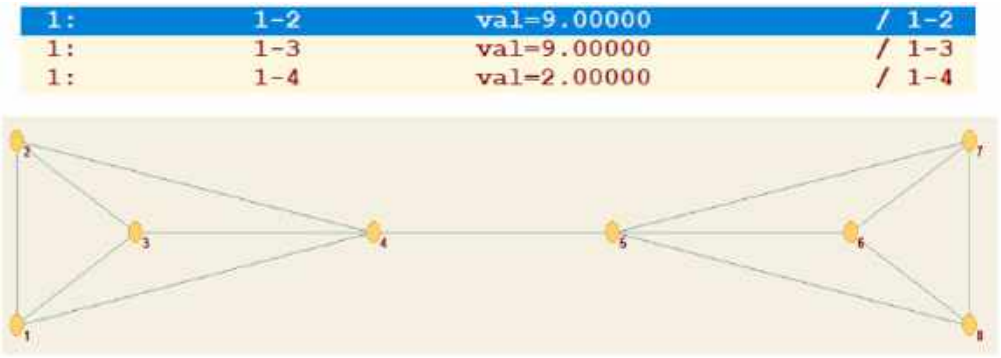

Figura 15. Estructura del Cromosoma con genes distintos

El total de número de líneas se obtuvo 0 arcos y 8 genes (Figura 15.), con la densidad1 de bucles permitidos de 0,40625000, así como una densidad2 de bucles no permitidos de 0,46428571; de esta manera la red de centralización de todos los genes es 0,14285714; además se logra obtener la Estadística descriptiva de los genes para su evaluación, siendo la media aritmética 3,2500 y media sea 3,000; así como su frecuencia total sea del 100% (Figura 16.)

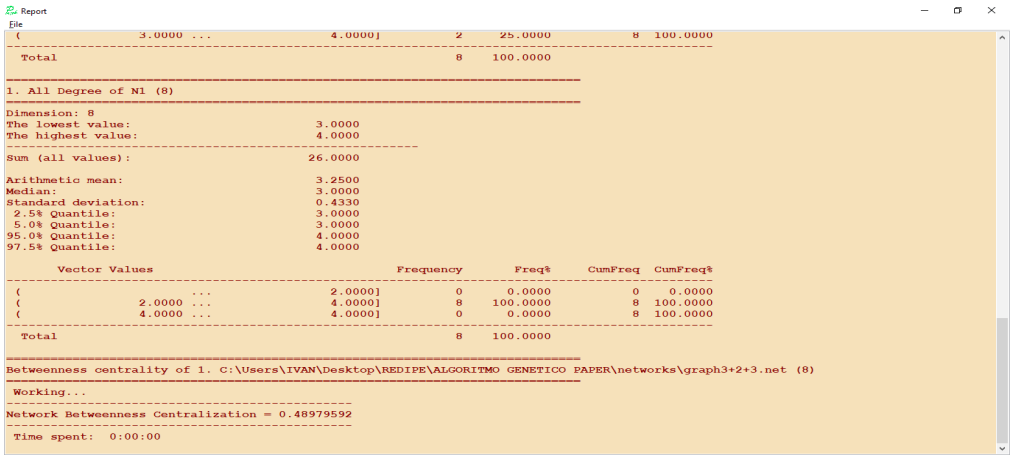

Figura 16. Valores estadísticos del cromosoma de genes distintos

R E V I S T A B O L E T Í N R E D I P E 9 (8) : 1 1 6 - 1 3 1 - A G O S T O 2 0 2 0 - I S S N 2 2 5 6 - 1 5 3 6

De esta manera logramos obtener centralización de intermediación del cromosoma de 0,48979592 de un cromosoma con 8 genes, con el generador de genes aleatorios de 25 y finalmente obtener la centralización de intermediación del cromosoma 0,48979592 (Figura 17.).

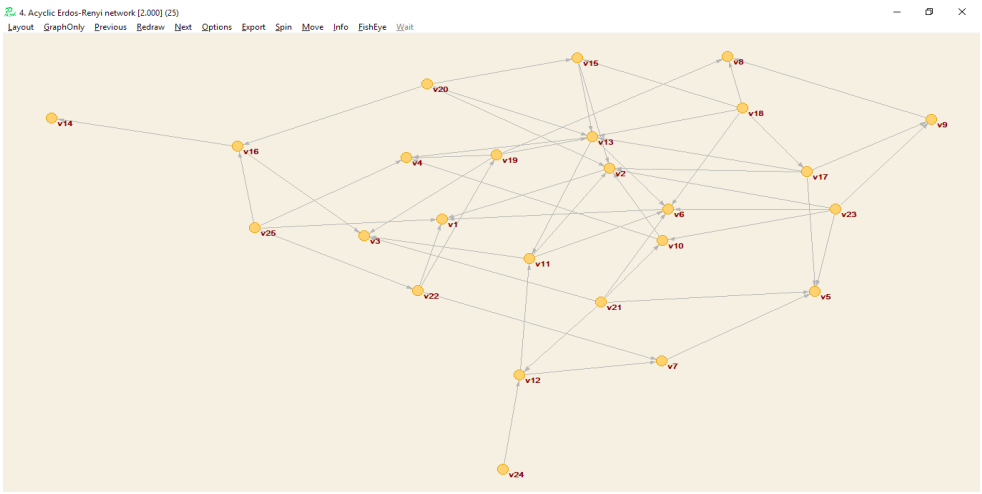

Figura 17. Distribución de los genes distintos en el cromosoma mejorado

Finalmente en este caso no se logró obtener el cromosoma mejor adaptado

CONCLUSIÓN

- Hemos visto que los algoritmos genéticos están indicados para resolver todo tipo de problemas que se puedan expresar como un problema de optimización donde definimos una representación adecuada para las soluciones y para la función a optimizar.

- El algoritmo genético resolvió los diferentes pasos, como es la selección, cruce, mutación, para obtener el mejor gen del gen mutado, dependiendo de los cromosomas ingresados que a su vez genes que en su procesamiento serán mutados y seleccionados en un cruzamiento al azar en un punto determinado después de los entrenamientos que se requiera, para poder predecir a cuál clúster pertenecería según la Aptitud para obtener el Elitismo del cromosoma.

- Si el espacio de búsqueda es grande, uniforme y monomodal, o el problema no se comprende bien, y la tarea no necesita encontrar un óptimo global, es decir, es suficiente encontrar rápidamente un buen optimo, entonces la elección de los AG es buena. Si el espacio no es demasiado grande se pueden utilizar algoritmos de búsqueda exhaustiva, con los que se está seguro de encontrar la mejor solución, mientras que el AG puede converger prematuramente a un óptimo local.

- Si el espacio es uniforme o monomodal, los algoritmos de gradiente ascendente como el hill-climbing serán mucho más eficientes que los AGs. Si el espacio es bien conocido, se pueden diseñar técnicas heurísticas que usen dominio específico y funcionan mejor que los algoritmos de propósito general como los AG.

REVISTA BOLETÍN REDIPE 9 (8): 116-131 - AGOSTO 2020 - ISSN 2256-1536

REFERENCIAS

Bäck, T. (1992) *Self-Adaptation in Genetic Algorithms*. En Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life. F. J. Varela and P. Bourgine (eds.). pp. 263-271. MIT Press

Beasley, D., Bull, D. R., Martin, R. R. (1993*) An Overview of Genetic Algorithms:* Part 1, Fundamentals. En University Computing, 15(2) 58-69.

Biondi, J., Michel, O. (1995). *From the chromosome to the neural network.* En Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms. D.W. Pearson, N.C. Steele, and R.F. Albrecht (eds)

Cantú-Paz, E. (1997). *A survey of Parallel Genetic Algorithms*. Technical Report Illinois Genetic Algorithms Laboratory. University of Illinois at Urbana-Champaign.

Coello Coello, C. (2002). *Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A survey of the state of the Art*. En Computer Methods in Applied Mechanics and Engineering. 191(11-12):1245-1287

Deb, K. (2004). *Genetic Algorithms for optimization*. En Statistical Computing: Existing Methods and Recent Developments. D. Kundu y A. Basu (eds.). pp. 85-123. New Delhi, India: Narosa Publishing House

Fogel, R. B. (2000). *La ecorregión de Neembucú: Infortunio, dignidad y sabiduría de sus antiguos pobladores*. Asunción: Centro de Estudios Rurales Interdisciplinarios.

Fogel, A., Hsu, H.-C., Shapiro, A. F., Nelson-Goens, G. C., & Secrist, C. (2006). *Effects of normal and perturbed social play on the duration and amplitude of different types of infant smiles.* Developmental Psychology, 42(3), 459–473.

Forrest, S. (1996). *Genetic Algorithms.* En ACM Computer Survey. 28(1). pp. 77-80

Ilachinsky, A. (1997*). Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Combat*. Research Memorandum CRM 97-61.10. Center for Naval Analyses

Joglekar, A., Tungare, M. (2001). *Genetic Algorithms and their use in the design of Evolvable Hardware.* Fr. Conceicao Rodrigues College of Engineering

Whitley, D. (1994*). A Genetic Algorithm Tutorial*. En Statistics and Computing. Vol 4, pp. 65-85.

Whitley, D. (2001). *An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls*. En Journal of Information and Software Technology. Vol 43(14), pp. 817-831.

Whitley, D. (2002) *Genetic Algorithms and Evolutionary Computing.* En Van Nostrand‘s Scientific Encyclopedia.

Yao, X. (1996). *An Overview of Evolutionary Computation*. En Chinese Journal of Advanced Software Research. Vol 3(1), pp. 12-29, Allerton Press Inc.