

Facultad de Ingeniería Industrial y de Sistemas

“LA INTELIGENCIA ARTIFICIAL Y EL GOBIERNO ELECTRÓNICO
EN EL INSTITUTO DE GOBERNABILIDAD Y TECNOLOGÍA”

Tesis para optar el Título Profesional de
Ingeniero de Sistemas.

AUTOR (A)

MUGRUZA TORRES, ROBERTO GUILLERMO

ASESOR (A)

Dr. Jorge Victor Mayhuasca Guerra

JURADO

Mg. Oscar Benavides Cavero

Mg. Pervis Paredes Paredes

Mg. Batállanos Casas Williams Hernán

Ing. Blasdemir Isidoro Calderón Cuenca

Lima – Perú

2018

DEDICATORIA:

A mi madre Ely por sus enseñanzas y los consejos de proyecto de vida.

RESUMEN

En estos tiempos con el avance de las nuevas tecnologías de información y la inteligencia artificial podemos realizar experimentos con algoritmos en el desarrollo de software. También tenemos investigaciones dedicadas a diseñar y desarrollar algoritmos que sean capaces de resolver problemas heurísticos, en este caso el algoritmo genético. El objetivo de este proyecto es diseñar un software en lenguaje `c#` con algoritmo genético que genere una lista para el análisis de los proyectos de inversión pública en la plataforma de Gobierno Electrónico.

En esta investigación presento conceptos fundamentales de la Inteligencia Artificial, un poco de historia, enfocándonos en los Algoritmos Genéticos y su capacidad para resolver problemas en los proyectos de Gobierno Electrónico.

Esta investigación permite acercar a los futuros profesionales de la Universidad Nacional Federico Villarreal al mundo de la investigación científica en inteligencia artificial y desarrollar soluciones para el desarrollo de nuestra sociedad.

Palabras clave: Inteligencia Artificial, Algoritmos Genéticos, Gobierno Electrónico

ABSTRACT

In these times with the advance of new information technologies and artificial intelligence we can perform experiments with algorithms in software development. We also have research dedicated to designing and developing algorithms that are capable of solving heuristic problems, in this case the genetic algorithm. The objective of this project is to design a software in c # language with a genetic algorithm that generates a list for the analysis of public investment projects in the Electronic Government platform.

In this research I present fundamental concepts of Artificial Intelligence, a bit of history, focusing on Genetic Algorithms and their ability to solve problems in Electronic Government projects.

This research allows to approach the future professionals of the National University Federico Villarreal to the world of scientific research in artificial intelligence and develop solutions for the development of our society.

Keywords: Artificial Intelligence, Genetic Algorithms, Electronic Government

ÍNDICE

I. Introducción	5
1.1 Descripción y formulación del problema	6
1.1.1 Problema General	6
1.1.2 Problema Específico	6
1.2 Antecedentes	7
1.3 Objetivos de Investigación	12
1.3.1 Objetivo General	12
1.3.2 Objetivo Específico	12
1.4 Justificación	13
1.5 Hipótesis	13
II. Marco Teórico	15
2.1 Bases teóricas sobre inteligencia artificial	15
2.1.1 Inteligencia artificial	15
2.1.2 Algoritmos genéticos	16
2.1.3 Estructura de algoritmos genéticos	18
2.2 Bases teóricas sobre gobierno electrónico	36
2.2.1 Gobierno electrónico	36
III. Método	38
3.1 Tipo de Investigación	38
3.2 Ámbito temporal y espacial	38
3.3 Variables	38
3.4 Población y muestra	38
3.5 Instrumentos	39
3.6 Procedimientos	41
3.6.1 Desarrollo del sistema de algoritmos genéticos	41
3.6.2 Aplicación del modelo de algoritmos genéticos propuesto	52
3.7 Análisis de datos	58
IV. Resultados	61
V. Discusión de resultados	63
VI. Conclusiones	65
VII. Recomendaciones	66
VIII. Referencias	67

I. INTRODUCCION

La presente investigación en inteligencia artificial y el gobierno electrónico en el instituto de gobernabilidad y tecnología, muestra los avances de los sistemas de información en el mundo actual son muy acelerados, haciendo una gestión de las nuevas tecnologías de información, sumándose los avances de la inteligencia artificial y muchos han sido los avances realizados en la historia reciente para conseguir reproducir en máquinas y software las capacidades del ser humano. La Inteligencia Artificial se encarga dotar acciones inteligentes a las entidades creadas, simulando la inteligencia del ser humano para resolver diferentes problemas en nuestra sociedad.

En ese sentido, tenemos la gestión del gobierno electrónico, donde podemos crear proyectos de plataforma electrónica en general para la toma de decisiones aplicando la inteligencia artificial.

En esta investigación se verán los conceptos y el desarrollo de la inteligencia artificial con algoritmos genéticos, los procesos que incluyen los algoritmos genéticos, sus características de codificación, cruzamiento, mutación para al final tener los mejores resultados de optimización en la selección de proyectos.

1.1 Descripción y formulación del problema

Se desea saber cuál es la influencia del uso de la Inteligencia Artificial mediante los Algoritmos Genéticos en la disminución del tiempo en los análisis de los Proyectos de Inversión Pública en una Plataforma de Gobierno Electrónico del Instituto de Gobernabilidad y Tecnología.

1.1.1 Problema General

La Inteligencia Artificial con Algoritmos Genéticos Influye en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.

1.1.2 Problema Específico

Se presentan el siguiente problema específico:

¿Cuál es la influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional?

1.2 Antecedentes

En los antecedentes internacionales tenemos las siguientes investigaciones que a continuación voy a describir.

Según Amador (2018), en su tesis de Maestría en Inteligencia Artificial, presentada a la Universidad Politécnica de Madrid – España, en su investigación de razonamiento automático sobre bases de conocimiento mediante deep learning, usan modelos de bases de conocimientos relacionados usando inteligencia artificial con aprendizaje profundo para que el sistema devuelva respuestas inteligentes a preguntas específicas gracias al razonamiento de la red artificial.

Según Antón (2015), en su tesis de Maestría en Inteligencia Artificial, presentada a la Universidad Politécnica de Madrid – España, en su investigación busca encontrar el árbol de expansión mínimo con restricción de grado de un grafo (DCMST por sus siglas en inglés) es un problema NP-complejo ampliamente estudiado. Una de sus aplicaciones más importantes es el diseño de redes. Donde se generan árboles de decisiones basados en permutaciones para codificar los bosques de información. Los algoritmos que utilizaron son: algoritmo de estimación de distribuciones (EDA), algoritmo genético generacional (gGA), algoritmo genético de estado estacionario (ssGA), estrategia evolutiva basada en la matriz de covarianzas (CMAES), evolución diferencial (DE), estrategia evolutiva elitista (ElitistES), estrategia evolutiva no elitista (NonElitistES) y optimización por enjambre de partículas (PSO). Los mejores resultados fueron para el algoritmo de estimación de distribuciones utilizado y ambos tipos de algoritmos genéticos, aunque los algoritmos genéticos fueron significativamente más rápidos.

Según Luengo (2014), en su tesis de Maestría en Inteligencia Artificial, presentada a la Universidad Politécnica de Madrid – España, en su investigación Clustering Basado en Redes Bayesianas con Predictoras Continuas - Aplicaciones en Neurociencia, donde desentrañar el funcionamiento del cerebro es uno de los principales desafíos a los que se enfrenta la ciencia actual. Un área de estudio que ha despertado muchas expectativas e interés es el análisis de la estructura cortical desde el punto de vista morfológico, de manera que se cree una simulación del cerebro a nivel molecular. Con el uso de las redes bayesianas clasifica un conjunto de clases más adecuado en el que agrupar. A partir de los resultados, se discute la existencia de las predicciones.

Según Castellanos (2013), en su tesis de Doctorado en Inteligencia Artificial, presentada a la Universidad Politécnica de Madrid – España, en su investigación Algoritmos para Minería de Datos con Redes de Neuronas, en un proceso de minería de datos con el conjunto a analizar se tiene como propósito la extracción de conocimiento mediante las redes de neuronas que es un método de aprendizaje a partir de un conjunto de patrones que se utiliza en minería de datos. La desventaja de las redes de neuronas en la minería de datos es su estructura compleja y el no poder interpretar la forma de obtener sus resultados. Se han propuesto unos algoritmos con el objetivo de extraer la información de un conjunto de datos con redes de neuronas en forma de reglas.

Según Imas (2013), en su tesis de Maestría en Inteligencia Artificial, presentada a la Universidad Politécnica de Madrid – España, en su investigación Algoritmos Inspirados en Swarm Intelligence para el Enrutamiento en Redes de Telecomunicaciones

La disciplina de swarm intelligence (SI) poco a poco ha ido ganando terreno en muchos ámbitos, entre ellos las redes de telecomunicación, donde estos algoritmos se posicionan como una posible alternativa a los algoritmos de enrutamiento convencionales. Gracias a la interacción de agentes muy simples y utilizando reglas sencillas, consiguen un comportamiento colectivo

que resuelve el problema de enrutamiento, adaptándose mejor a los cambios, sin necesidad de un ente central, de forma escalable y con una alta capacidad de tolerancia a fallos.

Según Peña (2011), en su tesis de Doctorado en Inteligencia Artificial, presentada a la Universidad Politécnica de Madrid – España, en su investigación Algoritmos de Distribución de Cargas, según esta investigación indica que con el surgir de los problemas que no se pueden resolver de forma eficiente en tiempo polinomial en base al dato de entrada, surge la computación natural como alternativa a la computación clásica, donde se trata de utilizar la naturaleza como base de cómputo para simular su comportamiento y obtener mejores resultados a los problemas obtenidos por la computación clásica.

En los antecedentes nacionales tenemos las siguientes investigaciones que a continuación voy a mencionar.

Según Angeles (2015), en su tesis para Título de Ingeniería, presentada a la Pontificia Universidad Católica del Perú, en su investigación de la presente tesis se presenta la Elaboración de una Solución Metaheurística usando un Algoritmo Genético que Permita Elaborar la Distribución de los Horarios Académicos, donde el problema es la generación de horarios académicos en instituciones de nivel superior.

La solución se construye con el uso de un algoritmo genético a partir de una población inicial generada por un algoritmo inteligente en fase construcción. Se ha tomado como caso de estudio a la facultad de Ciencia e Ingeniería de la Pontificia Universidad Católica del Perú, en la cual se contó con el apoyo del encargado de realizar el horario de la especialidad de ingeniería informática para el respectivo levantamiento de información, con lo cual se consiguió la adaptación de un algoritmo que cumpla con sus restricciones y requerimientos. Para facilitar la búsqueda de esta solución se aplicaron los operadores de selección, casamiento, mutación y elitismo, para encontrar una solución óptima usando los algoritmos genéticos.

Según Avalos (2015), en su tesis para Título de Ingeniería, presentada a la Pontificia Universidad Católica del Perú, en su investigación de la presente tesis se presenta un Algoritmo Genético para La Asignación de Tipo de Aviones A Vuelos, donde el problema es la asignaciones de vuelos de aviaciones basados en la capacidad de pasajeros, se usó los algoritmos genéticos para optimizar las condiciones de vuelo, aunque no necesariamente se encontró la solución óptima, pero se estuvo investigando y desarrollando usando el lenguaje de programación java.

Según Damián (2015), en su tesis de Maestría en Ciencias, presentada a la Universidad Nacional de Ingeniería, en su investigación de la presente tesis se emplea una Red Neuronal Artificial (RNA) para determinar la respuesta a una acción/desplazamiento lateral en muros de albañilería confinada. Para ello se preparan los datos de entrada al modelo, correspondiente al desplazamiento lateral y la carga vertical del muro confinado; así como la información de salida de la red, modelando el agrietamiento del muro como una secuencia de ceros y unos (0: no agrietado, 1: agrietado), además de la respuesta/fuerza lateral del muro que corresponde al patrón de agrietamiento. Luego se diseña la arquitectura; una Red Neuronal feedforward con propagación del error hacia atrás (Backpropagation), un algoritmo de entrenamiento de tipo gradiente descendente con momento y aprendizaje variable.

Según Leyva (2015), en su tesis de Maestría en Ingeniería de Sistemas, presentada a la Universidad Nacional Mayor de San Marcos, en su investigación de la presente tesis se utiliza la Inteligencia Artificial Algoritmo Greedy, la tesis aborda el problema de distribución en planta, el cual, en líneas generales, pretende asignar o distribuir instalaciones en un sistema de producción.

Uno de los planteamientos más abordados ha sido el problema de distribución en planta con instalaciones de áreas iguales, generalizándose como un problema de asignación cuadrática. Para solucionar este tipo de problemas, el presente trabajo desarrolla un algoritmo greedy para la distribución de las instalaciones en la planta con los siguientes tipos de naturaleza: sistema productivo por procesos, con rutas de procesamiento, presencia de equipos para el manejo de materiales en fila múltiple, distribución en un solo nivel y horizonte de planeamiento estático.

Según Gavidia (2014), en su tesis de Maestría en Informática, presentada a la Pontificia Universidad Católica del Perú, en su investigación de la presente tesis se utiliza la Inteligencia Artificial con los Algoritmos de Colonia de Hormigas, donde existen diversas técnicas de segmentación de imágenes, y en este trabajo se plantea un procedimiento de segmentación de imágenes médicas basado en la metaheurística de Algoritmos de Colonia de Hormigas. Los algoritmos de esta metaheurística imitan el comportamiento de las hormigas durante su búsqueda de alimento, dado que siempre produce rutas óptimas entre la fuente de comida y el nido. Dicha conducta fue implementada mediante hormigas artificiales con el objeto de realizar tareas específicas de procesamiento de imágenes.

Según Utrilla (2014), en su tesis de Doctorado en Ingeniería, presentada a la Universidad Nacional Mayor de San Marcos, en su investigación de la presente tesis se utiliza la Inteligencia Artificial para un Sistema de Arquitectura de Seguridad industrial, la investigación responde al siguiente problema de formular el modelo de un sistema multisensorial para la supervisión y seguridad industrial aplicando tecnología de inteligencia artificial para las empresas de nuestro país, considerando las variables más importantes del sistema, lo cual disminuyó la probabilidad en los trabajadores de sufrir riesgos y accidentes en el trabajo.

Según Dávila (2013), en su tesis de Maestría en Ingeniería, presentada a la Pontificia Universidad Católica del Perú, en su investigación de la presente tesis se utiliza la Inteligencia Artificial con las Redes Neuronales Artificiales , donde ha diseñado un Sistema de Estimación y Control de Calidad del Producto de Fondo de una Columna de Destilación de Multicomponentes, basado en red neuronal artificial dinámica; verificándose precisión y rapidez del estimador neuronal para determinar la calidad del producto, y con ello se garantizó la factibilidad de introducir dicha estimación en un esquema de control, que disminuyó la variabilidad del parámetro de calidad, a pesar de las perturbaciones en las condiciones operativas del proceso.

1.3 Objetivos

1.3.1 Objetivo General

Determinar la influencia de la Inteligencia Artificial con Algoritmos Genéticos en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.

1.3.2 Objetivo Específico

Se presentan el siguiente objetivo específico:

Determinar la influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

1.4 Justificación

Actualmente en la era de la Inteligencia Artificial es importante saber sus aplicaciones de los algoritmos genéticos artificiales en los tiempos de los procesos de análisis de los datos . Porque los analistas de información se toman un determinado tiempo para el análisis, siendo estos tiempos muy largos.

Según las investigaciones en la inteligencia artificial, los algoritmos genéticos optimizan las búsquedas y optimización de los resultados de la información de un determinado problema. Por lo tanto se justifica el uso de esta clase de algoritmos para ayudar a determinar su influencia en los tiempos de análisis de la información de los proyectos.

1.5 Hipótesis

1.5.1 Hipótesis General

Hipótesis nula:

H₀: No Existe Influencia de la Inteligencia Artificial con Algoritmos Genéticos en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.

Hipótesis alternativa:

H_a: Existe Influencia de la Inteligencia Artificial con Algoritmos Genéticos en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.

1.5.2 Hipótesis Específica

Hipótesis nula específica:

H_{0e}: No existe influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

Hipótesis alternativa específica:

H_{ae}: Existe influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

II. MARCO TEORICO

2.1 Bases Teóricas sobre Inteligencia Artificial

2.1.1 Inteligencia Artificial

Según Romero (2007), la Inteligencia Artificial (I.A.) se puede definir como aquella “inteligencia” exhibida por “cientefactos” o artefactos científicos contruidos por humanos, o sea que dice que un sistema artificial posee inteligencia cuando es capaz de llevar a cabo tareas que, si fuesen realizadas por un humano, se diría de este que es inteligente. Dentro de las ciencias de la computación, la rama de la I.A. se basa en intentar dotar al funcionamiento de las aplicaciones informáticas de un comportamiento inteligente similar al humano para la toma de decisiones. La I.A. dentro de la computación no es algo reciente ya que sus albores se pueden referir al comienzo de la Cibernética, hace más de 60 años, pero en los últimos años su avance se ha hecho más notable y se ha extendido a numerosas áreas. Así, ha sido llevada a intentar simular comportamientos humanos, no sólo razonamiento, sino también aspectos más difíciles de medir como es el caso de la creatividad o arte artificial.

La IA es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que si fuera llevado a cabo por una persona, se diría que es inteligente.

La contribución del matemático inglés Alan Turing al mundo de la computación en general, y al de la IA en particular, fue muy considerable. Turing, participó en el diseño de uno de los primeros computadores que existieron, desarrollados para el ejército inglés, entre los años 40 y 50.

Además de aportar los conceptos básicos de la arquitectura secuencial de los computadores, publicó en 1950 un provocador artículo que tiene por título “Computer Machinery and Intelligence”, que comienza con la no menos provocadora pregunta: ¿Pueden las máquinas pensar? En dicho artículo el autor intenta abordar formalmente acciones consideradas hasta ese momento propias de los seres humanos, de forma que después pudieran implementarse en las máquinas, dotándolas así de capacidades como: aprender y crear.

Según Benítez (2011), Turing en el año 1952, durante un debate, hizo la profecía, que en 100 años la inteligencia artificial iba a tener un gran avance que va a existir una maquina capaz de superar el test de Turing.

2.1.2 Algoritmos Genéticos

Según Gestal (2010), 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. Más formalmente, y siguiendo la definición dada, “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”.

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.

Según Gestal (2010), durante millones de años las diferentes especies se han adaptado para poder sobrevivir en un medio cambiante. De la misma manera se podría tener una población de potenciales soluciones a un problema, de las que se irían seleccionando las mejores hasta que se adaptasen perfectamente al medio, en este caso el problema a resolver. En términos muy generales se podría definir la computación evolutiva como una familia de modelos computacionales inspirados en la evolución.

Más formalmente, el término de computación evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas basadas en los principios de la evolución natural. Estas técnicas heurísticas podrían clasificarse en 3 grandes categorías o grupos, dando lugar a la ecuación evolutiva.

(Computación Evolutiva) = (Algoritmos Genéticos) + (Estrategias de Evolución) + (Programación Evolutiva)

Según Gestal (2010), el desarrollo de los Algoritmos Genéticos se debe en gran medida a John Holland, investigador de la Universidad de Michigan. A finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la selección natural. Aunque originalmente esta técnica recibió el nombre de “planes reproductivos”, a raíz de la publicación en 1975 de su libro “Adaptation in Natural and Artificial Systems”(Holland, 1975) se conoce principalmente con el nombre de Algoritmos Genéticos. A grandes rasgos un Algoritmo Genético consiste en una población de soluciones codificadas de forma similar a cromosomas. Cada uno de estos cromosomas tendrá asociado un ajuste, valor de bondad o fitness, que cuantifica su validez como solución al problema. En función de este valor se le darán más o menos oportunidades de reproducción.

Además, con cierta probabilidad se realizarán mutaciones de estos cromosomas.

Este proceso hará posible que los individuos genéticos tiendan hacia las soluciones a un problema dado, aunque las condiciones del espacio de búsqueda varíen con el transcurso del tiempo. Las bases de las Estrategias de Evolución fueron apuntadas en 1973 por Rechember.

2.1.3 Estructura de los Algoritmos Genéticos

Todas las definiciones de la estructura de los algoritmos genéticos de la presente investigación se ha tomado de la referencia bibliográfica de Gestal (2010).

Según Gestal (2010), cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros. El conjunto de todos los parámetros (genes en la terminología de Algoritmos Genéticos) se codifica en una cadena de valores denominada cromosoma.

El conjunto de los parámetros representado por un cromosoma particular recibe el nombre de genotipo. El genotipo contiene la información necesaria para la construcción del organismo, es decir, la solución real al problema, denominada fenotipo. Por ejemplo, en términos biológicos, la información genética contenida en el ADN de un individuo sería el genotipo, mientras que la expresión de ese ADN (el propio individuo) sería el fenotipo.

Desde los primeros trabajos de John Holland la codificación suele hacerse mediante valores binarios. Se asigna un determinado número de bits a cada parámetro y se realiza una discretización de la variable representada por cada gen. El número de bits asignados dependerá del grado de ajuste que se desee alcanzar. Evidentemente no todos los parámetros tienen por qué estar codificados con el mismo número de bits. Cada uno de los bits pertenecientes a un gen suele recibir el nombre de alelo.

En la Figura 1 se muestra un ejemplo de un individuo binario que codifica 3 parámetros.

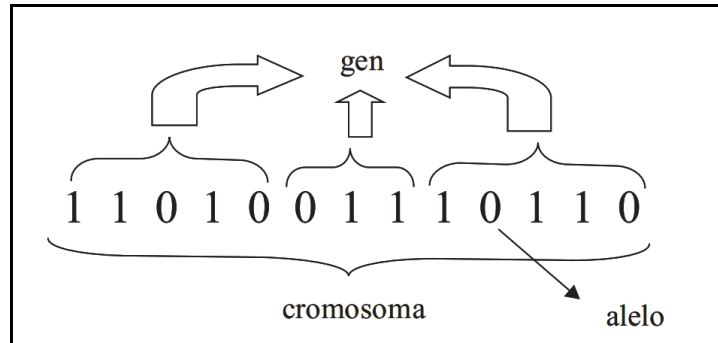


Figura 1. Individuo Genético Binario
Fuente: Gestal (2010).

Sin embargo, también existen representaciones que codifican directamente cada parámetro con un valor entero, real o en punto flotante. A pesar de que se acusa a estas representaciones de degradar el paralelismo implícito de las representaciones binarias, permiten el desarrollo de operadores genéticos más específicos al campo de aplicación del Algoritmo Genético.

En la Figura 2, se muestra un ejemplo típico de la aplicación de los Algoritmos Genéticos: cómo representar una red de neuronas artificiales para posteriormente realizar el proceso de optimización de los pesos sinápticos. Codificar una red de neuronas artificiales en forma de cromosoma es tan sencillo como asignar un gen del cromosoma a cada uno de los pesos de la red. También se podrían añadir genes que indiquen el número de capas y el número de elementos de procesado en cada una.

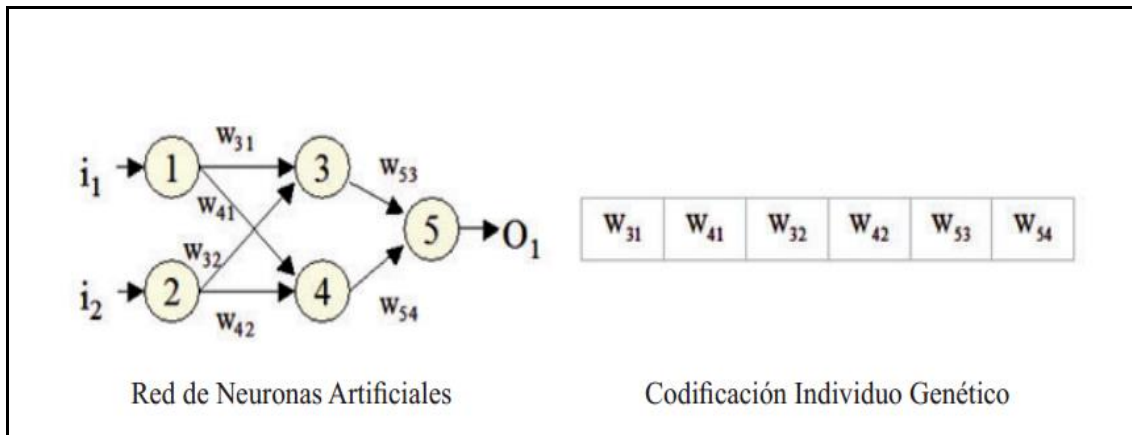


Figura 2. Codificación de una Red Neuronas Artificiales.
Fuente: Gestal (2010).

Según Gestal (2010), los Algoritmos Genéticos trabajan sobre una población de individuos.

Cada uno de ellos representa una posible solución al problema que se desea resolver. Todo individuo tiene asociado un ajuste de acuerdo a la bondad con respecto al problema de la solución que representa (en la naturaleza el equivalente sería una medida de la eficiencia del individuo en la lucha por los recursos).

Una generación se obtiene a partir de la anterior por medio de los operadores de reproducción. Existen 2 tipos: cruce y copia.

Cruce, se trata de una reproducción de tipo sexual. Se genera una descendencia a partir del mismo número de individuos (generalmente 2) de la generación anterior. Existen varios tipos que se detallarán en un punto posterior.

Copia, se trata de una reproducción de tipo asexual. Un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación.

```

Inicializar población actual aleatoriamente
MIENTRAS no se cumpla el criterio de terminación
    crear población temporal vacía
    SI elitismo: copiar en población temporal mejores individuos
    MIENTRAS población temporal no llena
        seleccionar padres
        cruzar padres con probabilidad  $P_c$ 
        SI se ha producido el cruce
            mutar uno de los descendientes (prob.  $P_m$ )
            evaluar descendientes
            añadir descendientes a la población temporal
        SINO
            añadir padres a la población temporal
        FIN SI
    FIN MIENTRAS
    aumentar contador generaciones
    establecer como nueva población actual la población temporal
FIN MIENTRAS

```

Figura 3. Funcionamiento de un Algoritmo Genético.

Fuente: Gestal (2010).

El funcionamiento genérico de un Algoritmo Genético puede apreciarse en el pseudocódigo, reflejado en la Figura 3.

Si desea optarse por una estrategia elitista, los mejores individuos de cada generación se copian siempre en la población temporal, para evitar su pérdida.

A continuación comienza a generarse la nueva población en base a la aplicación de los operadores genéticos de cruce y/o copia. Una vez generados los nuevos individuos se realiza la mutación con una probabilidad P_m . La probabilidad de mutación suele ser muy baja, por lo general entre el 0.5% y el 2%.

Se sale de este proceso cuando se alcanza alguno de los criterios de parada fijados. Los más usuales comentado a continuación.

Los mejores individuos de la población, representan soluciones suficientemente buenas para el problema que se desea resolver.

La población ha convergido, un gen ha convergido cuando el 95% de la población tiene el mismo valor para él, en el caso de trabajar con codificaciones binarias, o valores dentro de un rango especificado en el caso de trabajar con otro tipo de codificaciones. Una vez que todos los genes alcanzan la convergencia se dice que la población ha convergido. Cuando esto ocurre la media de bondad de la población se aproxima a la bondad del mejor individuo.

Cuando se alcanza un máximo, se ha alcanzado el número de generaciones máximo especificado.

Sobre este algoritmo inicialmente propuesto por Holland se han definido numerosas variantes. Quizás una de las más extendidas consiste en prescindir de la población temporal de manera que los operadores genéticos de cruce y mutación se aplican directamente sobre la población genética. Con esta variante el proceso de cruces varía ligeramente. Ahora no basta, en el caso de que el cruce se produzca, con insertar directamente la descendencia en la población. Puesto que el número de individuos de la población se ha de mantener constante, antes de insertar la descendencia en la población se le ha de hacer sitio. Es decir, para ubicar a los descendientes generados previamente se han de eliminar otros individuos de la población genética. Existen para ello diversas opciones, que se comentarán con más detalle en un punto posterior.

Evidentemente, trabajando con una única población no se puede decir que se pase a la siguiente generación cuando se llene la población, pues siempre está llena. En este caso el paso a la siguiente generación se producirá una vez que se hayan alcanzado cierto número de cruces y mutaciones. Este número dependerá de la tasa de cruces y mutaciones especificadas por el usuario y del tamaño de la población. Así, con una tasa de cruces del 90%, una tasa de mutaciones del 2% y trabajando con 100 individuos se pasará a la siguiente generación cuando se alcanzasen 45 cruces (cada cruce genera 2 individuos con lo que se habrían insertado en la población 90 individuos, esto es el 90%) o 2 mutaciones.

Otra variación común consiste en la modificación del esquema de selección de los individuos que serán mutados. En el esquema mostrado, sólo los descendientes originados a partir de un cruce son mutados (proceso que imita los errores de transcripción del ADN que tienen lugar en la naturaleza); otra opción habitual es la selección aleatoria del individuo a mutar entre todos los que forman parte de la población genética.

Según Gestal, M. (2010), para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce, copia y mutación. En el caso de no trabajar con una población intermedia temporal también cobran relevancia los algoritmos de reemplazo. A continuación se verán en mayor detalle.

Selección, los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto, la selección de un individuo estará relacionada con su valor de ajuste.

No se debe, sin embargo, eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea.

En cuanto a algoritmos de selección se refiere, estos pueden ser divididos en dos grandes grupos: probabilísticos y determinísticos. Ambos tipos de algoritmos basan su funcionamiento en el principio indicado anteriormente (permitir escoger una mayor cantidad de veces a los más aptos). Sin embargo, como su nombre indica, el primer tipo adjudica estas posibilidades con un importante componente basado en el azar. Es en este grupo donde se encuentran los algoritmos de selección por ruleta o por torneo que, dado su importancia por ser los más frecuentemente utilizados, se describen con detalle en esta sección. El segundo grupo engloba una serie de algoritmos que, dado el ajuste conocido de cada individuo, permite asignar a cada uno el número de veces que será escogido para reproducirse. Esto puede evitar problemas de predominancia de ciertos individuos y cada uno de estos algoritmos presentan variaciones respecto al número de veces que se tomarán los mejores y peores y, de esta forma, se impondrá una presión en la búsqueda en el espacio de estados en la zona donde se encuentra el mejor individuo (en el caso de que se seleccionen más veces los mejores), o bien que se tienda a repartir la búsqueda por el espacio de estados, pero sin dejar de tender a buscar en la mejor zona (caso de repartir más la selección).

Una opción bastante común consiste en seleccionar el primero de los individuos participantes en el cruce mediante alguno de los métodos expuestos en esta sección y el segundo de manera aleatoria.

Selección por ruleta, este método es el más utilizado desde los orígenes de los Algoritmos Genéticos. A cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente, la población está ordenada en base al ajuste, por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo $[0..1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

Es un método muy sencillo pero ineficiente a medida que aumenta el tamaño de la población. Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez.

Selección por torneo, la idea principal de este método de selección consiste en escoger a los individuos genéticos en base a comparaciones directas entre sus genotipos.

Existen dos versiones de selección mediante torneo, el torneo determinístico y el torneo probabilístico, que a continuación pasan a detallarse.

En la versión determinística se selecciona al azar un número p de individuos (generalmente se escoge $p=2$). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

La versión probabilística únicamente se diferencia en el paso de selección del ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo $[0..1]$, si es mayor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo

más alto y en caso contrario el menos apto. Generalmente p toma valores en el rango $0.5 < p \leq 1$ Variando el número de individuos que participan en cada torneo se puede modificar la presión de selección. Cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. Un caso particular es el elitismo global. Se trata de un torneo en el que participan todos los individuos de la población, con lo cual la selección se vuelve totalmente determinística. Cuando el tamaño del torneo es reducido, la presión de selección disminuye y los peores individuos tienen más oportunidades de ser seleccionados.

Elegir uno u otro método de selección determinará la estrategia de búsqueda del Algoritmo Genético. Si se opta por un método con una alta presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Existen muchos otros algoritmos de selección. Unos buscan mejorar la eficiencia computacional, otros el número de veces que los mejores o peores individuos pueden ser seleccionados. Algunos de estos algoritmos son muestreo determinístico, escalamiento sigma, selección por jerarquías, estado uniforme, sobrante estocástico, brecha generacional, etc.

Cruce, una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación. Tal y como se ha indicado anteriormente, el cruce es una estrategia de reproducción sexual.

Su importancia para la transición entre generaciones es elevada puesto que las tasas de cruce con las que se suele trabajar rondan el 90%.

Los diferentes métodos de cruce podrán operar de dos formas diferentes. Si se opta por una estrategia destructiva los descendientes se insertarán en la población temporal aunque sus padres tengan mejor ajuste (trabajando con una única población esta comparación se realizará con los individuos a reemplazar).

Por el contrario, utilizando una estrategia no destructiva la descendencia pasará a la siguiente generación únicamente si supera la bondad del ajuste de los padres (o de los individuos a reemplazar).

La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de los padres. Al compartir las características buenas de dos individuos, la descendencia, o al menos parte de ella, debería tener una bondad mayor que cada uno de los padres por separado. Si el cruce no agrupa las mejores características en uno de los hijos y la descendencia tiene un peor ajuste que los padres no significa que se esté dando un paso atrás. Optando por una estrategia de cruce no destructiva garantizamos que pasen a la siguiente generación los mejores individuos. Si, aún con un ajuste peor, se opta por insertar a la descendencia, y puesto que los genes de los padres continuarán en la población –aunque dispersos y posiblemente levemente modificados por la mutación–, en posteriores cruces se podrán volver a obtener estos padres, recuperando así la bondad previamente pérdida.

Existen multitud de algoritmos de cruce. Sin embargo los más empleados son los que se detallarán a continuación: cruce de 1 punto, cruce de 2 puntos y cruce uniforme.

Cruce de 1 punto, es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres.

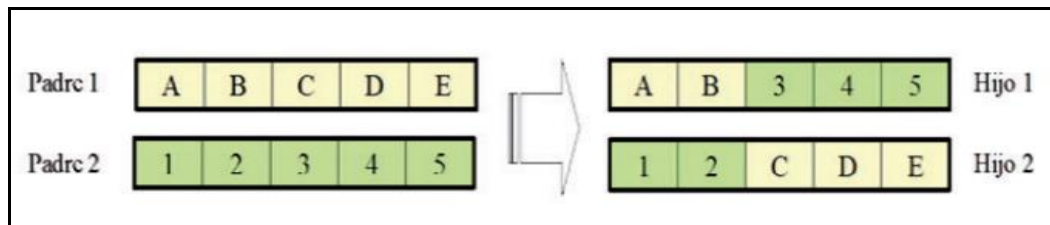


Figura 4. Cruce de un Punto.
Fuente: Gestal (2010).

En la Figura 4, se puede ver con claridad el proceso descrito anteriormente.

Cruce de 2 puntos, se trata de una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres, como en el caso anterior, se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre.

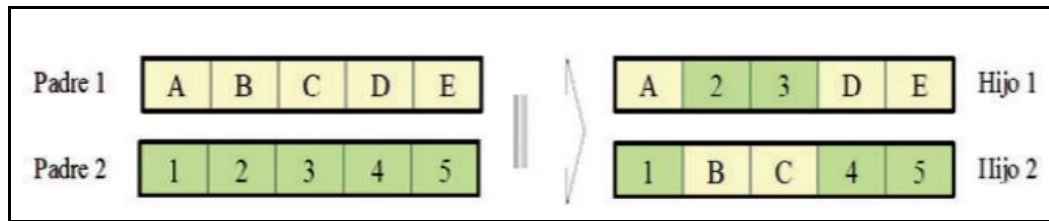


Figura 5. Cruce de dos Puntos.
Fuente: Gestal (2010).

Generalmente, es habitual referirse a este tipo de cruce con las siglas DPX (Double Point Crossover). En la Figura 5, se muestra un ejemplo de cruce por dos puntos.

Generalizando, se pueden añadir más puntos de cruce dando lugar a algoritmos de cruce multipunto. Sin embargo existen estudios que desaprueban esta técnica. Aunque se admite que el cruce de 2 puntos aporta una sustancial mejora con respecto al cruce de un solo punto, el hecho de añadir un mayor número de puntos de cruce reduce el rendimiento del Algoritmo Genético. El problema principal de añadir nuevos puntos de cruce radica en que es más fácil que los segmentos originados sean mal formados, es decir, que por separado quizás pierdan las características de bondad que poseían conjuntamente. Hay que evitar, por lo tanto, romper dichos segmentos, denominados bloques constructivos.

Sin embargo no todo son desventajas y añadiendo más puntos de cruce se consigue que el espacio de búsqueda del problema sea explorado con más intensidad.

Cruce uniforme, 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.

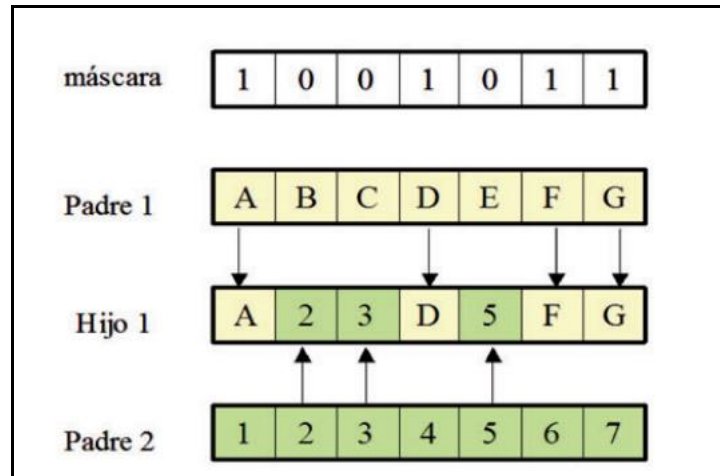


Figura 6. Cruce Uniforme.
Fuente: Gestal (2010).

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.

Tal y como se puede apreciar en la Figura 6, la descendencia contiene una mezcla de genes de cada uno de los padres. El número efectivo de puntos de cruce es fijo pero será por término medio $L/2$, siendo L la longitud del cromosoma (número de alelos en representaciones binarias o de genes en otro tipo de representaciones).

La máscara de cruce puede no permanecer fija durante todo el proceso evolutivo. Se genera de manera aleatoria para cada cruce. Se suele referir a este tipo de cruce con las siglas UPX (Uniform Point Crossover).

Cruces específicos de codificaciones no binarias, los tres tipos de cruce vistos hasta el momento son válidos para cualquier tipo de representación del genotipo. Si se emplean genotipos compuestos por valores enteros o reales pueden definirse otro tipo de operadores de cruce.

Media: el gen de la descendencia toma el valor medio de los genes de los padres. Tiene la desventaja de que únicamente se genera un descendiente en el cruce de dos padres.

Media geométrica: cada gen de la descendencia toma como valor la raíz cuadrada del producto de los genes de los padres. Presenta el problema añadido de qué signo dar al resultado si los padres tienen signos diferentes.

Extensión: se toma la diferencia existente entre los genes situados en las mismas posiciones de los padres y se suma al valor más alto o se resta del valor más bajo. Solventa el problema de generar un único descendiente. Una variante basada en este cruce son los Algoritmos genéticos de evolución diferencial.

Algoritmos de reemplazo, cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, deberá tenerse en cuenta que para insertar un nuevo individuo deberá de eliminarse previamente otro de la población. Existen diferentes métodos de reemplazo:

Aleatorio: el nuevo individuo se inserta en un lugar escogido de manera aleatoria en la población.

Reemplazo de padres: se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.

Reemplazo de similares: una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.

Reemplazo de los peores: de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

Copia, la copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación.

El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de ese individuo.

Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce y, si éste finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados.

Elitismo, el elitismo es un caso particular del operador de copia consistente en copiar siempre al mejor, o en su caso mejores, individuos de una generación en la generación siguiente. De esta manera se garantiza que el proceso de búsqueda nunca dará un paso atrás en cuanto a la calidad de la mejor solución obtenida, sino que un cambio en ésta siempre implicará una mejora.

Una variación de este proceso consiste en copiar al mejor o mejores individuos de una generación en la siguiente, únicamente cuando tras el paso de una generación no se haya mejorado con los operadores de cruce o mutación la mejor solución de la generación actual.

Mutación, la mutación de un individuo provoca que alguno de sus genes, generalmente uno sólo, varíe su valor de forma aleatoria.

Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la mutación se suele utilizar de manera conjunta con el operador de cruce. Primeramente se seleccionan dos individuos de la población para realizar el cruce. Si el cruce tiene éxito entonces uno de los descendientes, o ambos, se muta con cierta probabilidad P_m . Se imita de esta manera el comportamiento que se da en la naturaleza, pues cuando se genera la descendencia siempre se produce algún tipo de error, por lo general sin mayor trascendencia, en el paso de la carga genética de padres a hijos.

La probabilidad de mutación es muy baja, generalmente menor al 1%. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de mutados. Sin embargo se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga una probabilidad nula de ser examinado.

Tal y como se ha comentado, la mutación más usual es el reemplazo aleatorio. Este consiste en variar aleatoriamente un gen de un cromosoma. Si se trabaja con codificaciones binarias, consistirá simplemente en negar un bit.

También es posible realizar la mutación intercambiando los valores de dos alelos del cromosoma. Con otro tipo de codificaciones no binarias existen otras opciones.

Incrementar o decrementar a un gen una pequeña cantidad generada aleatoriamente, multiplicar un gen por un valor aleatorio próximo a 1.

Aunque no es lo más común, existen implementaciones de Algoritmos Genéticos en las que no todos los individuos tienen los cromosomas de la misma longitud.

Esto implica que no todos ellos codifican el mismo conjunto de variables. En este caso existen mutaciones adicionales como puede ser añadir un nuevo gen o eliminar uno ya existente.

Evaluación, para el correcto funcionamiento de un Algoritmo Genético se debe de poseer un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Por lo tanto, para cada tipo de problema que se desee resolver deberá derivarse un nuevo método, al igual que ocurrirá con la propia codificación de los individuos.

De esto se encarga la función de evaluación, que establece una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste. En la naturaleza el ajuste (o adecuación) de un individuo puede considerarse como la probabilidad de que ese individuo sobreviva hasta la edad de reproducción y se reproduzca. Esta probabilidad deberá estar ponderada con el número de individuos de la población genética.

En el mundo de los Algoritmos Genéticos se empleará esta medición para controlar la aplicación de los operadores genéticos. Es decir, permitirá controlar el número de selecciones, cruces, copias y mutaciones llevadas a cabo.

La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se le asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido. Tal y como se ha comentado,

este procedimiento de evaluación será específico del dominio del problema en el que se aplica el Algoritmo Genético.

También puede calcularse el ajuste mediante una manera 'co-evolutiva'. Por ejemplo, el ajuste de una estrategia de juego se determina aplicando esa estrategia contra la población entera (o en su defecto una muestra) de estrategias de oposición.

2.2 Base teóricas sobre Gobierno Electrónico

2.2.1 Gobierno Electrónico

Según Kettani y Moulin (2014), Gobierno Electrónico se define como un proceso de reforma en la manera que los gobiernos trabajan, comparten información y ofrecen servicios a los clientes externos e internos. En este sentido, el gobierno electrónico se produce cuando una institución gubernamental o institución privada utiliza las TIC para analizar, transformas datos de la gestión de estado.

En su definición más simple, el gobierno electrónico se refiere al uso de las tecnologías digitales para transformar las operaciones del gobierno con el fin de mejorar la eficacia, la eficiencia y prestación de servicios.

Según Barrenechea (2014), indica que el uso de la Internet en ordenadores ha sido sustituido por la del dispositivo móvil. En el sector privado, la Internet móvil es ya una fuente de accionamiento para la productividad, la eficiencia, y la producción. En el sector público, sin embargo, muchas organizaciones no han podido mantener el ritmo en la adopción de las tecnologías de punta como herramientas de productividad.

Según Pando y Fernández (2013) indica que el Gobierno Electrónico a pesar que el impacto que tiene la utilización intensiva de tecnologías de información (TI) para fortalecer la eficacia y eficiencia de la gestión pública local, este fenómeno (genéricamente denominado gobierno electrónico) no concitó la debida atención de aquellos que se dedican a estudiar los procesos de reforma desde una mirada integral. El valor de las herramientas tecnológicas es innegable, ya que su influencia es fundamental en la gestión y las políticas públicas.

El concepto de Gobierno Electrónico según Rodríguez (2011) nos menciona que la continua optimización de la entrega de servicios, la participación ciudadana y el ejercicio del gobierno mediante la transformación de las relaciones internas y externas a través de la tecnología, la Internet y los nuevos medio, en este marco el Gobierno Electrónico supone una nueva manera de abordar las relaciones entre las TIC, la gestión pública y la acción política. Abarca las dimensiones de e-administración, e-política y el diseño e implementación de políticas para la Sociedad de la Información.

III.MÉTODO

3.1 Tipo de Investigación

El tipo de estudio es investigación aplicada. Diseño experimental.

Siendo su diseño en concordancia con los aportes teóricos de Hernández (2010) refiere el término experimento tiene al menos dos acepciones, una general y otra particular. La general se refiere a “elegir o realizar una acción” y después observar las consecuencias.

3.2 Ámbito temporal y espacial

En el ámbito temporal, la presente investigación se ha desarrollado entre los meses de julio del 2017 a julio del 2018. En el ámbito espacial, la presente investigación se ha desarrollado en el Instituto de Gobernabilidad y Tecnología.

3.3 Variables

Por ser un estudio de investigación experimental se tienen que la existencia de una variable independiente y la variable dependiente. Donde la variable independiente ejerce influencia en la variable dependiente.

Variable Independiente : Algoritmos Genéticos Artificiales

Variable Dependiente : Proyectos de Plataforma Electrónica

3.4 Población y muestra

Las 25 Regiones del Perú. Se procesan información en base a las 25 Regiones por ser los proyectos referentes a estas 25 regiones.

Se aplicaron los siguientes valores para sacar la muestra:

Nivel de Confianza: 99%

Porcentaje de Error: 1%

Tamaño de la Población: 25 regiones.

La muestra es de 25 regiones.

3.5 Instrumentos

Se usa la técnica de **Observación**.

El indicador que se usa es el tiempo de proceso de análisis, donde la Unidad de medida es expresada en **minutos**.

Tabla 1

Tiempo en minutos de Análisis en los Proyectos de Inversión Regional antes del experimento.

Item	Regiones	Minutos
1	Amazonas	10
2	Ancash	10
3	Apurímac	10
4	Arequipa	10
5	Ayacucho	10
6	Cajamarca	10
7	Callao	11
8	Cusco	10
9	Huancavelica	10
10	Huánuco	10
11	Ica	10
12	Junín	10
13	La Libertad	10
14	Lambayeque	10
15	Lima	10
16	Loreto	10

17	Madre de Dios	10
18	Moquegua	10
19	Pasco	10
20	Piura	10
21	Puno	10
22	San Martín	10
23	Tacna	10
24	Tumbes	10
25	Ucayali	10

**Tabla 1. Tiempo expresado en minutos, antes del experimento.
Fuente: Elaboración propia**

3.6 Procedimientos

3.6.1 Desarrollo del Sistema de Algoritmos Genéticos

Los siguientes procesos se siguieron para crear el sistema de algoritmos genéticos:

- a- Creación de Tablas y Procedimientos Almacenados de Base de Datos.
- b- Diseño del modelo de inteligencia artificial de algoritmos genéticos.
- c- Construcción de software de inteligencia artificial en algoritmos genéticos.

a- Creación de las tablas y procedimientos almacenados:

Se crea las tablas en la base de datos usando Sql Server.

Tabla de las regiones con sus poblaciones:

En la siguiente instrucción se comprende los comandos Sql Server para la construcción de la tabla tRegionPoblación.

```
CREATE TABLE tRegionPoblacion  
(  
codreg int,  
nomreg varchar(200),  
numpob int  
)
```

En la siguiente instrucción se comprende los comandos Sql Server para la inserción de datos de la tabla tRegionPoblación.

```
insert into tRegionPoblacion values (1,'Amazonas',375993)
insert into tRegionPoblacion values (2,'Áncash',1063459)
insert into tRegionPoblacion values (3,'Apurímac',404190)
insert into tRegionPoblacion values (4,'Arequipa',1152303)
insert into tRegionPoblacion values (5,'Ayacucho',612489)
insert into tRegionPoblacion values (6,'Cajamarca',1387809)
insert into tRegionPoblacion values (7,'Callao',876877)
insert into tRegionPoblacion values (8,'Cusco',1171403)
insert into tRegionPoblacion values (9,'Huancavelica',454797)
insert into tRegionPoblacion values (10,'Huánuco',762223)
insert into tRegionPoblacion values (11,'Ica',711932)
insert into tRegionPoblacion values (12,'Junín',1225474)
insert into tRegionPoblacion values (13,'La Libertad',1617050)
insert into tRegionPoblacion values (14,'Lambayeque',1112868)
insert into tRegionPoblacion values (15,'Lima',8445211)
insert into tRegionPoblacion values (16,'Loreto',891732)
insert into tRegionPoblacion values (17,'Madre de Dios',109555)
insert into tRegionPoblacion values (18,'Moquegua',161533)
insert into tRegionPoblacion values (19,'Pasco',280449)
insert into tRegionPoblacion values (20,'Piura',1676315)
insert into tRegionPoblacion values (21,'Puno',1268441)
insert into tRegionPoblacion values (22,'San Martín',728808)
insert into tRegionPoblacion values (23,'Tacna',288781)
insert into tRegionPoblacion values (24,'Tumbes',200306)
```

En la siguiente instrucción se comprende los comandos Sql Server para la construcción de la tabla tRegionInversion.

Tabla de regiones con la inversión pública:

CREATE TABLE tRegionInversion

```
(
CodReg int,
PIA decimal(16,2),
PIM decimal(16,2),
PorAva decimal(16,2),
MesInv int,
AioInv int
)
```

En la siguiente instrucción se comprende los comandos Sql Server para la inserción de datos de la tabla tRegionInvesrion con los datos al mes de julio del 2018 del Ministerio de Economía y Finanzas del Perú.

```
insert into tRegionInversion values (1,151324021,386344049, 24.1,7,2018)
insert into tRegionInversion values (2,237128764,463388587, 32.5,7,2018)
insert into tRegionInversion values (3,128644546,324766807, 21.2,7,2018)
insert into tRegionInversion values (4,349998400,822834022, 19.7,7,2018)
insert into tRegionInversion values (5,193600983,441980535, 16.3,7,2018)
insert into tRegionInversion values (6,199956252,576795796, 16.4,7,2018)
insert into tRegionInversion values (7,86189661,177203442, 4.7,7,2018)
insert into tRegionInversion values (8,382665977,724021886, 19.5,7,2018)
insert into tRegionInversion values (9,153717520,298824902, 15.7,7,2018)
insert into tRegionInversion values (10,134526061,383513862, 20.5,7,2018)
insert into tRegionInversion values (11,72740668,102974398, 12.7,7,2018)
insert into tRegionInversion values (12,154545871,431624802, 22.4,7,2018)
insert into tRegionInversion values (13,587864078,733206383, 7.0,7,2018)
```

```

insert into tRegionInversion values (14,111126657,293816368, 36.0,7,2018)
insert into tRegionInversion values (15,140504501,274007060, 12.5,7,2018)
insert into tRegionInversion values (16,109394536,278045136, 32.3,7,2018)
insert into tRegionInversion values (17,79904938,180047571, 24.0,7,2018)
insert into tRegionInversion values (18,106072685,205233569, 27.3,7,2018)
insert into tRegionInversion values (19,84018871,384331384, 24.3,7,2018)
insert into tRegionInversion values (20,184649834,1003547854, 20.3,7,2018)
insert into tRegionInversion values (21,126857560,378962724, 37.0,7,2018)
insert into tRegionInversion values (22,201539788,420328476, 23.0,7,2018)
insert into tRegionInversion values (23,51189244,247367592, 29.3,7,2018)
insert into tRegionInversion values (24,53624910,90828800, 35.6,7,2018)
insert into tRegionInversion values (25,61005758,369408759, 18.4,7,2018)

```

Creación de los procedimientos almacenados, se crea los procedimientos almacenados en la base de datos usando Sql Server.

Procedimiento almacenado de inversión:

```

CREATE PROCEDURE sp_Inversion
@codigo int
as
Begin
    if (@codigo>0)
        Begin
            Select I.CodReg,R.NomReg, R.NumPob,I.PIA, I.PIM,I.PorAva
            From tRegionInversion I inner join tRegionPoblacion R on
            (I.CodReg=R.CodReg)
            Where I.CodReg=@codigo
        End
    Else
        Begin
            Select I.CodReg,R.NomReg, R.NumPob,I.PIA, I.PIM,I.PorAva
            From tRegionInversion I inner join tRegionPoblacion R on
            (I.CodReg=R.CodReg)
        End
End

```

Procedimiento almacenado de regiones:

```
CREATE PROCEDURE sp_Regiones
    @codigo int
as
Begin
    if (@codigo>0)
        Begin
            Select * From tRegionPoblacion where codreg=@codigo
        End
    Else
        Begin
            Select * From tRegionPoblacion
        End
    End
End
```

Procedimiento almacenado de inteligencia artificial de selección:

```
CREATE PROCEDURE sp_AISeleccion
    @codigo int
as
Begin
    if (@codigo>0)
        Begin
            Select I.CodReg asCodigo,
            '' as Cromosoma,
            R.NomReg as Region, R.NumPob as Poblacion,
            I.PIA as InversionApertura, I.PIM as InversionModificada,I.PorAva as
            PorcentajeAvance
            From tRegionInversion I inner join tRegionPoblacion R on
            (I.CodReg=R.CodReg)
            Where I.CodReg=@codigo
        End
    End
End
```

b- Diseño del modelo de inteligencia artificial de algoritmos genéticos:

Se crea los componentes de ingeniería de software para la implementación del sistema de algoritmos genéticos artificiales.

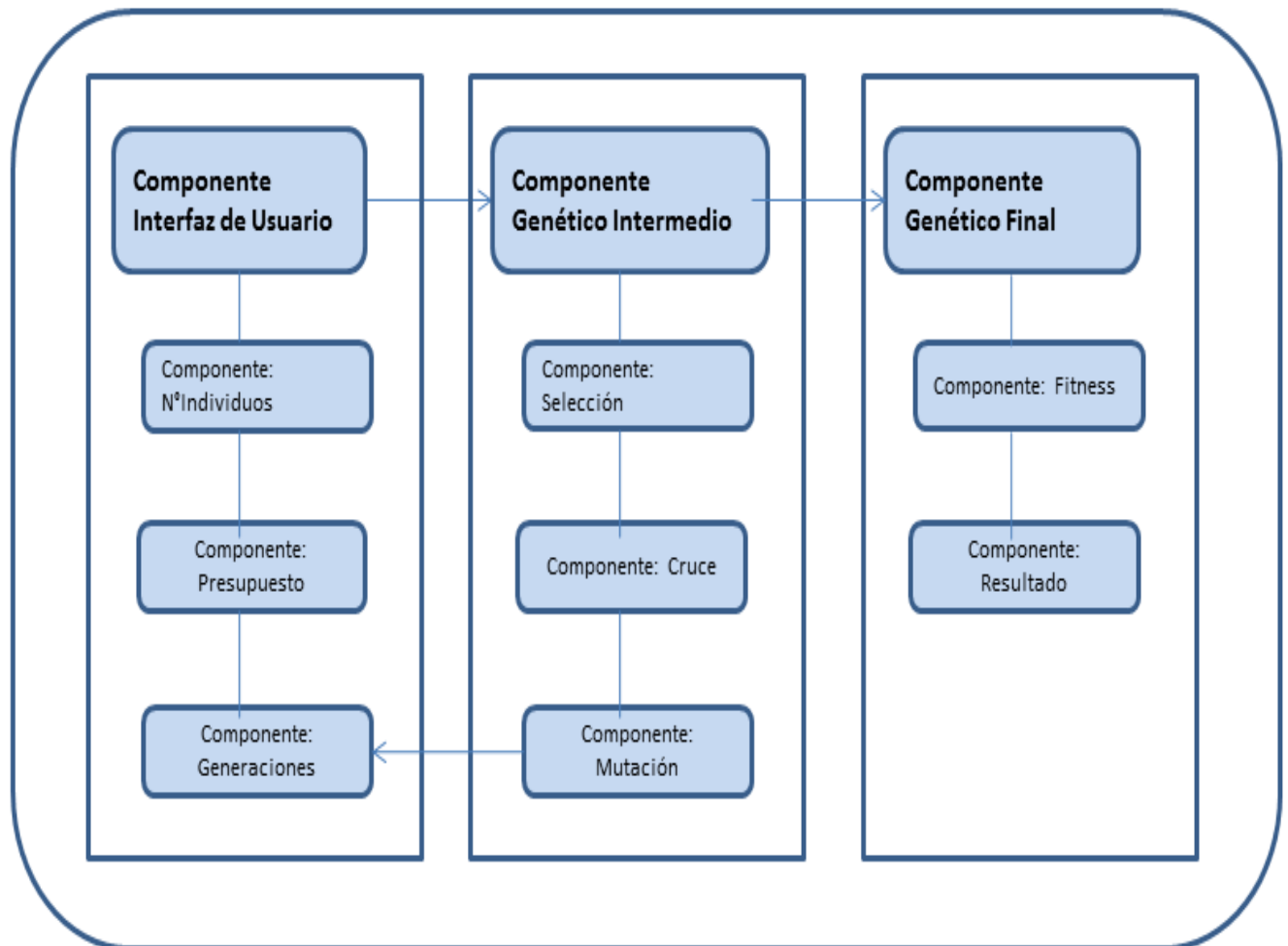


Figura 7. Modelo de Ingeniería de Algoritmo Genético.
Fuente: Elaboración Propia.

c- Construcción de software de inteligencia artificial en algoritmos genéticos

Se ha desarrollado las clases de software para la elaboración y construcción del sistema de algoritmos genéticos para la inversión en regiones. Se diseñó tres capas la capa de dato, la capa de la lógica y la capa de la interfaz de usuario.

Las clases fundamentales:

- blSelección.cs
- blCodificacion.cs
- blCruce.cs
- blMutacion.cs
- blFitness.cs
-

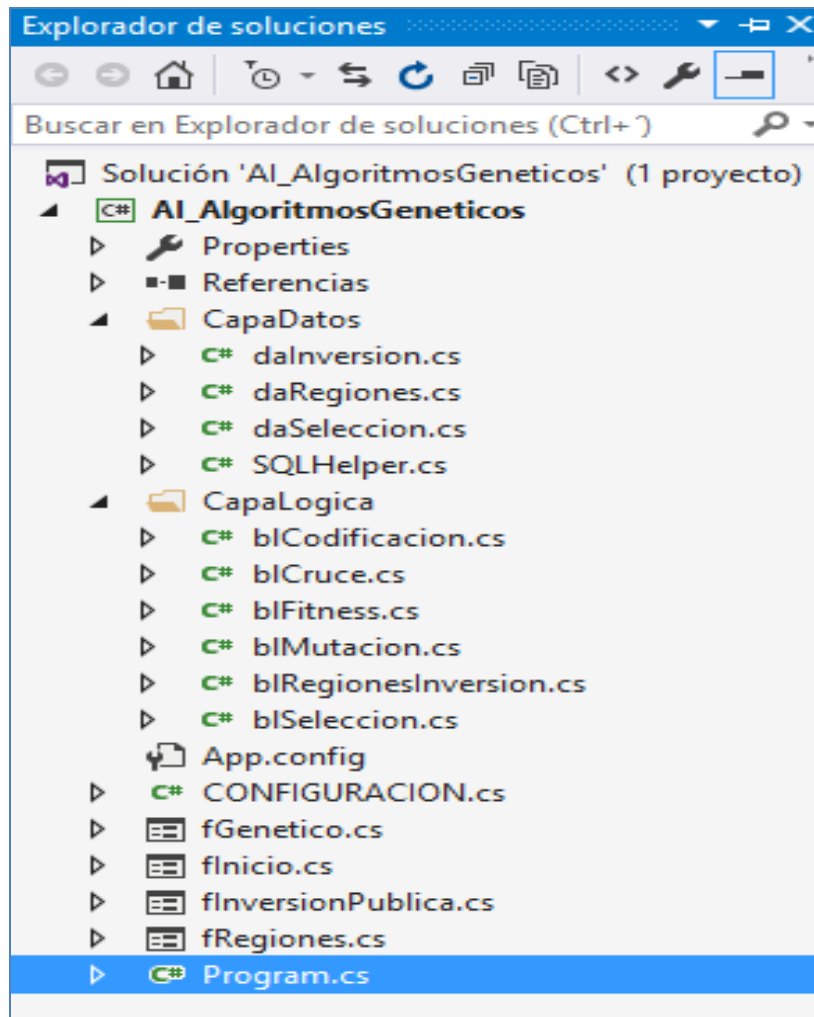


Figura 8: Construcción del Software con sus Clases en C#.
Fuente: Elaboración Propia.

En la codificación se usó el lenguaje de programación `c#` donde el investigador posee más de diez años de experiencia en su uso, además respecto del costo del software `c#`, existe una licencia de costo libre llamada Visual Studio Community.

El desarrollo de la codificación se sigue en la obtención de los cromosomas iniciales de la población, para ello se tiene que codificar la información o los datos y transformarlos en datos binarios.

```

public class blCodificacion
{
    5 referencias
    public string ObtenerCromosoma(int iValor)
    {
        string sCromosoma = "";
        sCromosoma= Convert.ToString(iValor, 2);
        sCromosoma= CompletarCromosoma(sCromosoma);
        return sCromosoma;
    }

    1 referencia
    private string CompletarCromosoma(string sBinario)
    {
        string sCromosoma = "";

        switch (sBinario.Length)
        {
            case 1:
                sCromosoma = "0000" + sBinario;
                break;
            case 2:
                sCromosoma = "000" + sBinario;
                break;
            case 3:
                sCromosoma = "00" + sBinario;
                break;
            case 4:
                sCromosoma = "0" + sBinario;
                break;
            case 5:
                sCromosoma = sBinario;
                break;
        }
        return sCromosoma;
    }
}

```

Figura 9: Obtención de los cromosomas en C#.
Fuente: Elaboración Propia.

Se desarrolla el cruce genético uniforme, para ellos se obtienen los cromosomas de los padres, los cromosomas del padre y los cromosomas de la madre para poder realizar mas adelante el cruce genético. Luego de ello se pasa al proceso de generación de hijos genéticos.

```

public DataSet ObtenerCruceUniforme(DataSet dsInicial)
{
    DataSet dsTotal = new DataSet();
    DataSet dsHijos = new DataSet();
    string sCromosomaMadre = "";
    string sCromosomaPadre = "";
    GenerarMascara();
    // Creando Lista de Nuevos Individuos

    for (int h=0; h<dsInicial.Tables[0].Rows.Count; h++)
    {
        // Obteniendo la Madre
        if (h == 0)
        {sCromosomaMadre = dsInicial.Tables[0].Rows[h]["Cromosoma"].ToString();}

        if (h > 0)
        {
            // Obteniendo el Padre
            sCromosomaPadre = dsInicial.Tables[0].Rows[h]["Cromosoma"].ToString();
            // Cruzamiento
            dsHijos = ObtenerHijos(sCromosomaMadre, sCromosomaPadre, false);
            dsTotal.Merge(dsHijos);

            if ((h+1) < dsInicial.Tables[0].Rows.Count)
            {
                // Obteniendo la Madre
                sCromosomaMadre = dsInicial.Tables[0].Rows[h+1]["Cromosoma"].ToString();
                if (((h + 1) ==dsInicial.Tables[0].Rows.Count-1))
                { dsHijos = ObtenerHijos(sCromosomaMadre, sCromosomaPadre, true);
                  dsTotal.Merge(dsHijos); }
                h = h + 1;
            }
        }
    }
    return dsTotal;
}

```

Figura 10: Cruce uniforme genético en C#.
Fuente: Elaboración Propia.

La obtención de hijos se realiza por medio de los cruces, en el proceso de “ObtenerHijos” se desarrolla los cruces de los cromosomas, para obtener parientes genéticos. En este caso se observa que se tienen dos hijos codificados con nuevos cromosomas.

```

public DataSet ObtenerHijos (string sCromosomaMadre, string sCromosomaPadre, Boolean HijoUnico)
{
    // Declarando
    DataSet ds = new DataSet();
    DataSet dsHijo1 = new DataSet();
    DataSet dsHijo2 = new DataSet();
    daSeleccion dSeleccion = new daSeleccion();
    blCodificacion dCodifica = new blCodificacion();
    blMutacion dMutacion = new blMutacion();
    string sCromosomaHijo1 = "";
    string sCromosomaHijo2 = "";
    string sCaracterMascara = "";
    int Hijo1;
    int Hijo2;

    // Inicio de Cruce
    for (int i=0; i< MascaraCromosoma.Length; i++)
    {
        sCaracterMascara = MascaraCromosoma.Substring(i, 1);
        if (sCaracterMascara == "0")
        {
            sCromosomaHijo1 += sCromosomaMadre.Substring(i, 1);
            sCromosomaHijo2 += sCromosomaPadre.Substring(i, 1);
        }
        else
        {
            sCromosomaHijo1 += sCromosomaPadre.Substring(i, 1);
            sCromosomaHijo2 += sCromosomaMadre.Substring(i, 1);
        }
    }
    // Termino de Cruce
}

```

Figura 11: Obtención de los hijos genéticos en C#.
Fuente: Elaboración Propia.

Después de la obtención de los hijos, se desarrolla la mutación, en cual se muta aleatoriamente un gen del cromosoma de cada hijo, si este hijo genético debe estar dentro de los límites de mutación positiva inferior y mutación positiva superior.

```

// Mutacion Positiva Superior
if (sCromosomaHijo1 == "00000") sCromosomaHijo1 = dMutacion.MutacionPositivaSuperior(sCromosomaHijo1);
if (sCromosomaHijo2 == "00000") sCromosomaHijo2 = dMutacion.MutacionPositivaSuperior(sCromosomaHijo2);
Hijo1 = Convert.ToInt32(sCromosomaHijo1, 2);
Hijo2 = Convert.ToInt32(sCromosomaHijo2, 2);
// Mutacion Positiva Inferior
if (Hijo1 > 24) sCromosomaHijo1 = dMutacion.MutacionPositivaInferior(sCromosomaHijo1);
if (Hijo2 > 24) sCromosomaHijo2 = dMutacion.MutacionPositivaInferior(sCromosomaHijo2);
Hijo1 = Convert.ToInt32(sCromosomaHijo1, 2);
Hijo2 = Convert.ToInt32(sCromosomaHijo2, 2);
// Mutacion
sCromosomaHijo1 = dMutacion.Mutacion(sCromosomaHijo1);
sCromosomaHijo2 = dMutacion.Mutacion(sCromosomaHijo2);
Hijo1 = Convert.ToInt32(sCromosomaHijo1, 2);
Hijo2 = Convert.ToInt32(sCromosomaHijo2, 2);

dsHijo1 = dSeleccion.Seleccion(Hijo1.ToString());
dsHijo1.Tables[0].Rows[0]["Cromosoma"] = dCodifica.ObtenerCromosoma(Hijo1);

dsHijo2 = dSeleccion.Seleccion(Hijo2.ToString());
dsHijo2.Tables[0].Rows[0]["Cromosoma"] = dCodifica.ObtenerCromosoma(Hijo2);

// Retornando Individuos Hijos
if (HijoUnico == true)
{
    ds.Merge(dsHijo1);
}
else
{
    ds.Merge(dsHijo1);
    ds.Merge(dsHijo2);
}

return ds;
}

```

Figura 12: Mutación genética en C#.

Fuente: Elaboración Propia.

3.6.2 Aplicación del Modelo de Algoritmos Genéticos Propuesto

Se ha desarrollado un sistema inteligente con algoritmos genéticos para determinar las regiones que más están avanzando en la ejecución de proyectos de inversión, dado un presupuesto nacional inmediato.



Figura 13: Sistema de Algoritmos Genéticos.
Fuente: Elaboración Propia.

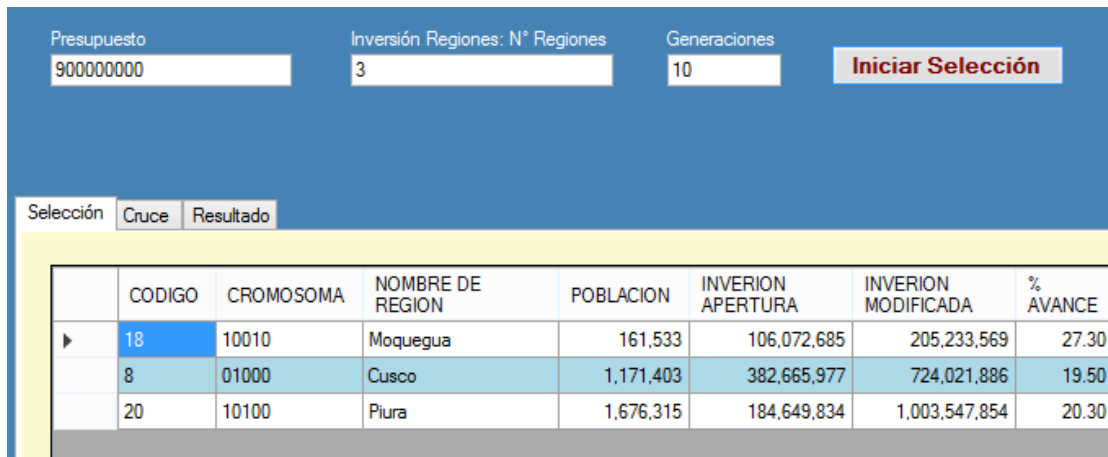
Se tiene los datos de la inversión pública del Ministerio de Economía y Finanzas (actualizado a Julio del 2018) de las regiones del Perú.

Inversion Publica						
	CODIGO	NOMBRE DE REGION	POBLACION	PIA	PIM	%Avance
▶	1	Amazonas	375,993.00	151,324,021.00	386,344,049.00	24.10
	2	Áncash	1,063,459.00	237,128,764.00	463,388,587.00	32.50
	3	Apurímac	404,190.00	128,644,546.00	324,766,807.00	21.20
	4	Arequipa	1,152,303.00	349,998,400.00	822,834,022.00	19.70
	5	Ayacucho	612,489.00	193,600,983.00	441,980,535.00	16.30
	6	Cajamarca	1,387,809.00	199,956,252.00	576,795,796.00	16.40
	7	Callao	876,877.00	86,189,661.00	177,203,442.00	4.70
	8	Cusco	1,171,403.00	382,665,977.00	724,021,886.00	19.50
	9	Huancavelica	454,797.00	153,717,520.00	298,824,902.00	15.70
	10	Huánuco	762,223.00	134,526,061.00	383,513,862.00	20.50
	11	Ica	711,932.00	72,740,668.00	102,974,398.00	12.70
	12	Junín	1,225,474.00	154,545,871.00	431,624,802.00	22.40
	13	La Libertad	1,617,050.00	587,864,078.00	733,206,383.00	7.00
	14	Lambayeque	1,112,868.00	111,126,657.00	293,816,368.00	36.00
	15	Lima	8,445,211.00	140,504,501.00	274,007,060.00	12.50
	16	Loreto	891,732.00	109,394,536.00	278,045,136.00	32.30
	17	Madre de Dios	109,555.00	79,904,938.00	180,047,571.00	24.00
	18	Moquegua	161,533.00	106,072,685.00	205,233,569.00	27.30
	19	Pasco	280,449.00	84,018,871.00	384,331,384.00	24.30
	20	Piura	1,676,315.00	184,649,834.00	1,003,547,854.00	20.30
	21	Puno	1,268,441.00	126,857,560.00	378,962,724.00	37.00
	22	San Martín	728,808.00	201,539,788.00	420,328,476.00	23.00
	23	Tacna	288,781.00	51,189,244.00	247,367,592.00	29.30
	24	Tumbes	200,306.00	53,624,910.00	90,828,800.00	35.60
	25	Ucayali	432,159.00	61,005,758.00	369,408,759.00	18.40

Figura 14: Las regiones del Perú con su Inversión y su Porcentaje de Avance en la Ejecución.

Fuente Datos: MEF – Julio del 2018. Elaboración Propia.

Se desea obtener 3 regiones presupuestales, según un presupuesto de 900 millones de soles, que tengan un mayor avance de ejecución presupuestal.



	CODIGO	CROMOSOMA	NOMBRE DE REGION	POBLACION	INVERION APERTURA	INVERION MODIFICADA	% AVANCE
▶	18	10010	Moquegua	161,533	106,072,685	205,233,569	27.30
	8	01000	Cusco	1,171,403	382,665,977	724,021,886	19.50
	20	10100	Piura	1,676,315	184,649,834	1,003,547,854	20.30

Figura 15: Sistema de Algoritmos Genéticos: Selección de Individuos.
Fuente: Elaboración Propia.

Selección de regiones: se ha seleccionado 3 regiones de manera aleatoria, con un porcentaje de avance en la ejecución de inversión pública al 67.1 %, sumando las 3 regiones. Se puede observar la codificación de los cromosomas.

Resultado genético: se ha procesado la información con los proyectos de inversión pública aplicando algoritmos genéticos con cruce uniforme y con mutación normal, mutación inferior y mutación superior.

Se obtiene 3 mejores individuos con un porcentaje de avance en la ejecución de inversión pública al 82.8 %, sumando las 3 regiones

Selección	Cruce	Resultado				
Presupuesto	Inversión Regiones: N° Regiones	Generaciones				
900000000	3	10				
Iniciar Selección						
CODIGO	CROMOSOMA	NOMBRE DE REGION	POBLACION	INVERION APERTURA	INVERION MODIFICADA	% AVANCE
16	10000	Loreto	891,732	109,394,536	278,045,136	32.30
23	10111	Tacna	288,781	51,189,244	247,367,592	29.30
▶ 3	00011	Apurímac	404,190	128,644,546	324,766,807	21.20

Figura 16: Sistema de Algoritmos Genéticos: Resultado Genético.
Fuente: Elaboración Propia.

Se obtuvieron las tres regiones en consulta que tienen el mayor avance de ejecución presupuestal.

En la siguiente tabla 2, se muestra los tiempos de análisis después del experimento, ello se logra con la observación y toma de datos con el instrumento expresados en tiempo en minutos.

Tabla 2

Tiempo en minutos de Análisis en los Proyectos de Inversión Regional después del experimento.

Item	Regiones	Minutos
1	Amazonas	1
2	Ancash	1
3	Apurímac	1
4	Arequipa	1
5	Ayacucho	1
6	Cajamarca	1
7	Callao	2
8	Cusco	1
9	Huancavelica	1
10	Huánuco	1
11	Ica	1
12	Junín	1
13	La Libertad	1
14	Lambayeque	1
15	Lima	1
16	Loreto	1
17	Madre de Dios	1
18	Moquegua	1
19	Pasco	1

20	Piura	1
21	Puno	1
22	San Martín	1
23	Tacna	1
24	Tumbes	1
25	Ucayali	1

**Tabla 2. Tiempo expresado en minutos, después del experimento.
Fuente: Elaboración propia**

3.7 Análisis de datos

Se procesó los datos usando el SPSS usando el Método de T-Student, porque nuestras muestras son inferiores a 30.

Según Bernal (2010), la prueba *t de Student* es una prueba estadística para evaluar hipótesis en torno a una media, cuando los tamaños de la muestra son menores de 30 mediciones.

El valor *t* se obtiene mediante la siguiente fórmula:

$$t = \frac{\bar{x} - \mu}{\frac{S}{\sqrt{n}}}$$

Dónde:

\bar{x} : Media de la Muestra.

μ : Media poblacional.

S : Desviación estándar.

n : Población.

El procesamiento de datos se ha desarrollado siguiendo el objetivo del estudio de la investigación.

Objetivo específico:

Determinar la influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

Para determinar la influencia de los objetivos de estudio, se procesaron los datos con los valores antes y después del experimento. En la Figura 17, se muestra el ingreso de los datos al software estadístico SPSS.

	PROCESO	NOMBRE	TIEMPO_ANTES	TIEMPO_DESPUES
1	1	Amazonas	10	1
2	2	Ancash	10	1
3	3	Apurímac	10	1
4	4	Arequipa	10	1
5	5	Ayacucho	10	1
6	6	Cajamarca	10	1
7	7	Callao	11	2
8	8	Cusco	10	1
9	9	Huancavelica	10	1
10	10	Huánuco	10	1
11	11	Ica	10	1
12	12	Junín	10	1
13	13	La Libertad	10	1
14	14	Lambayeque	10	1
15	15	Lima	10	1
16	16	Loreto	10	1
17	17	Madre de Dios	10	1
18	18	Moquegua	10	1
19	19	Pasco	10	1
20	20	Piura	10	1
21	21	Puno	10	1
22	22	San Martín	10	1
23	23	Tacna	10	1

Figura 17. Ingreso de Datos en el SPSS para realizar el análisis de datos.

Fuente: Elaboración Propia.

En la figura 18, se muestra el porcentaje de intervalo de confianza siendo de 99%, en una prueba T para muestras relacionadas.

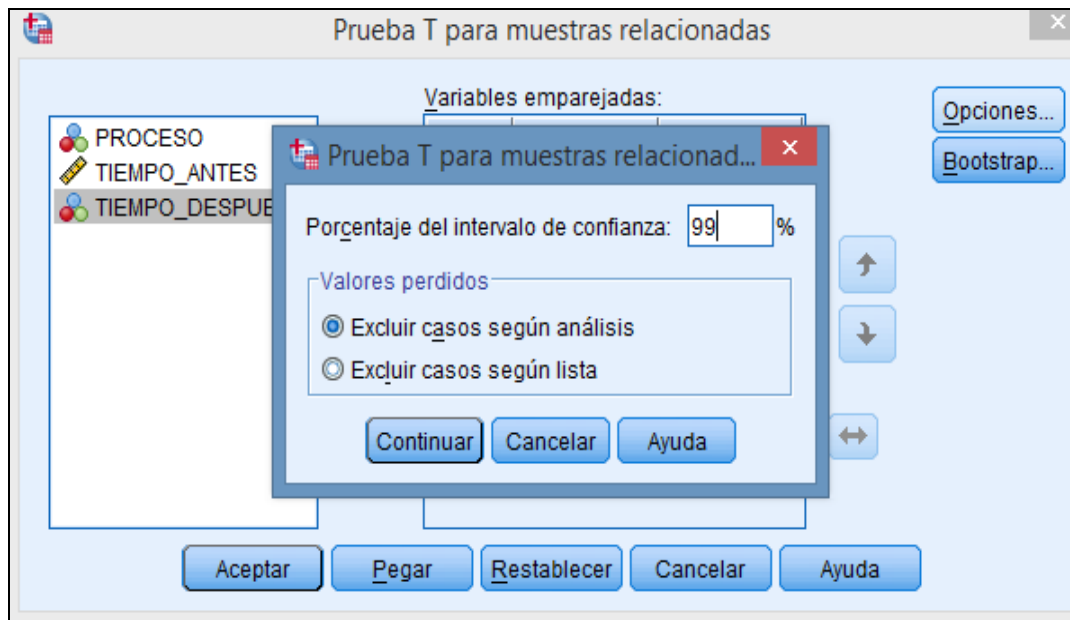


Figura 18. Ingreso de Datos en el SPSS prueba T para muestras relacionadas.

Fuente: Elaboración Propia.

El proceso de Determinar la influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional. Se ha usado el proceso con la prueba T de Student para muestras relacionadas. Los resultados se exponen en el capítulo siguiente donde se muestra la presentación de resultados.

IV. RESULTADOS

Los resultados han sido obtenidos usando el software estadístico SPSS, dando como resultado de la prueba t, siguiendo los objetivos de la investigación.

Objetivo específico:

Determinar la influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

Resultado del objetivo específico:

En la figura 19, se muestran los resultados estadísticos de la prueba T, donde el número de muestras es 25. Donde tenemos que la media expresada en minutos de la muestra antes del experimento es de 10.08 comparándola con la media después del experimento es 1.04, donde podemos observar una notable diferencia entre las medias.

Donde podemos observar además, que la desviación estándar es 0.20 .

Formula t de student:

$$t = \frac{\bar{x} - \mu}{\frac{S}{\sqrt{n}}}$$

Aplicando la fórmula :

$$t = \frac{10.08 - 1.04}{\frac{0.20}{\sqrt{25.00}}} = 226.00$$

Tenemos el valor de t igual a 126.00 con 24 grados de libertad y nos da como resultado un nivel de significancia de 0.000 .

→ **Prueba T**

Estadísticas de muestras emparejadas

		Media	N	Desviación estándar	Media de error estándar
Par 1	TIEMPO_ANTES	10,08	25	,400	,080
	TIEMPO_DESPUES	1,04	25	,200	,040

Correlaciones de muestras emparejadas

		N	Correlación	Sig.
Par 1	TIEMPO_ANTES & TIEMPO_DESPUES	25	1,000	,000

Prueba de muestras emparejadas

		Diferencias emparejadas				t	gl	Sig. (bilateral)	
		Media	Desviación estándar	Media de error estándar	99% de intervalo de confianza de la diferencia				
					Inferior				Superior
Par 1	TIEMPO_ANTES - TIEMPO_DESPUES	9,040	,200	,040	8,928	9,152	226,000	24	,000

Figura 19. Resultados estadísticos de la prueba T procesados con el SPSS del objetivo específico.

Fuente: Elaboración Propia.

V. DISCUSIÓN DE RESULTADOS

Hipótesis nula:

H₀: No Existe Influencia de la Inteligencia Artificial con Algoritmos Genéticos en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.

Hipótesis alternativa:

H_a: Existe Influencia de la Inteligencia Artificial con Algoritmos Genéticos en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.

Hipótesis específica:

H_{0e}: No Existe influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

H_{ae}: Existe influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

Según los resultados a un nivel de confianza del 99% y siendo el porcentaje de error del 1%, se observa como resultado que el nivel de significancia (p-valor) es 0.00 siendo menor a 0.01(1%), descartándose la hipótesis nula y se confirma la hipótesis alternativa.

Tenemos:

(p valor) $0.00 < 0.01$. Se descarta la hipótesis nula H_0 . Se afirma la hipótesis alternativa H_A .

Los resultados obtenidos de esta investigación comprueban la hipótesis propuesta, donde existe influencia de los Algoritmos Genéticos Artificiales en la disminución de los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

VI. CONCLUSIONES

- La implementación de las técnicas de la inteligencia artificial haciendo uso de los algoritmos genéticos, con la construcción de un software desarrollado en el lenguaje C# de visual studio de microsoft, optimiza los resultados de análisis en los datos de los presupuesto de inversión pública regionales, datos publicados por el Ministerio de Economía y Finanzas del Perú.
- La Inteligencia Artificial con Algoritmos Genéticos influye positivamente en los Análisis de Proyectos de Plataforma Electrónica en el Instituto de Gobernabilidad y Tecnología.
- Los Algoritmos Genéticos Artificiales disminuyen los tiempos en el análisis según un presupuesto, en los Proyectos de Inversión Pública Regional.

VII. RECOMENDACIONES

- Se recomienda desarrollar innovaciones experimentales con el uso de las nuevas tecnologías de información y las técnicas de inteligencia artificial basadas en algoritmos genéticos. Se debe usar lenguajes de programación que puedan trabajar en una arquitectura en capas dentro de la ingeniería de software, haciendo uso de los servicios web para crear servicios inteligentes.
- Se debe destacar un presupuesto universitario de inversión en inteligencia artificial para lograr estudios y avances sustanciales en la universidad. Este presupuesto universitario debe permitir una buena remuneración de los investigadores de la inteligencia artificial por ser consideradas investigaciones de alto nivel de impacto tecnológico y social.
- Se debe profundizar en las técnicas de algoritmos genéticos, redes neuronales, desarrollo cognitivo, aprendizaje profundo, en las investigaciones futuras, para una mayor variedad de investigaciones y soluciones de los problemas usando estas técnicas del conocimiento en inteligencia artificial.

VIII. REFERENCIAS

- Amador, E. (2018). *Maestría en Inteligencia Artificial: Razonamiento Automático sobre Bases de Conocimiento mediante Deep Learning*
España. Universidad Politecnica de Madrid.
- Antón, L. (2015). *Maestría en Inteligencia Artificial: Computación Evolutiva de Bosques de Expansión Mínimos con Restricción de Grado y de Rol.*
Madrid – España. Universidad Politecnica de Madrid.
- Angeles, A. (2015). *Título de Ingeniero Informático: Elaboración de una Solución Metaheurística usando un Algoritmo Genético que Permita Elaborar la Distribución de los Horarios Académicos.*
Lima – Perú. Pontificia Universidad Católica del Perú.
- Avalos, V. (2015). *Título de Ingeniero Informático: Algoritmo Genético para la Asignación de Tipo de Aviones a Vuelos.*
Lima – Perú. Pontificia Universidad Católica del Perú.
- Castellanos, A. (2013). *Doctorado en Inteligencia Artificial: Algoritmos para Minería de Datos con Redes de Neuronas.*
Madrid – España. Universidad Politecnica de Madrid.
- Damían, M. (2013). *Maestría en Ciencias: Predicción de la Respuesta Sísmica en Muros Empleando Redes Neuronales.*
Lima – Perú. Universidad Nacional de Ingeniería.
- Dávila, S. (2013). *Maestría en Ingeniería: Diseño de un Sistema de Control Neural para el Monitoreo y Control de Calidad.*
Lima – Perú. Pontificia Universidad Católica del Perú.
- Gavidia, C. (2014). *Maestría en Informática: Segmentación de Imágenes Médicas Mediante Algoritmos de Colonia de Hormigas.*
Lima – Perú. Pontificia Universidad Católica del Perú.

- Gestal, M. (2010). *Introducción a los Algoritmos Genéticos y la Programación Genética*. La Coruña – España. Editorial: Consorcio Editorial Gallego.
- Hernandez, R. (2010). *Metodología de la Investigación Científica*. México. Editorial Mc Graw Hill.
- Imas, A. (2013). *Maestría en Inteligencia Artificial: Algoritmos Inspirados en Swarm Intelligence para el Enrutamiento en Redes de Telecomunicaciones*. Madrid – España. Universidad Politecnica de Madrid.
- Kettani, D. y Moulin, B. (2014). *E-Government for Good Governance in Developing Countries*. Canada. Editorial Anthem Press.
- Leyva, M. (2015). *Maestría en Ingeniería de Sistemas: Un Algoritmo Greedy para el Diseño de distribución de Planta con Multi Productos*. Lima – Perú. Universidad Nacional Mayor de San Marcos.
- Luengo, S. (2014). *Maestría en Inteligencia Artificial: Clustering Basado en Redes Bayesianas con Predictoras Continuas - Aplicaciones en Neurociencia*. Madrid – España. Universidad Politecnica de Madrid.
- Pando, D. y Fernández, N. (2013). *El gobierno Electrónico a Nivel Local. Experiencias, tendencias y reflexiones*. Buenos Aires. Argentina. CIPPEC y Universidad de San Andrés.
- Peña, M. (2011). *Doctorado en Inteligencia Artificial: Algoritmos de Distribución de Cargas*. Madrid – España. Universidad Politecnica de Madrid.
- Rodríguez, E. (2011). *Ideas y Lineamientos para la Formulación de Políticas y Estrategias de E-Gob en los Gobiernos Locales Iberoamericanos*. Argentina. Editorial Passalacqua & Asociados.

- Romero, J. (2007). *Inteligencia Artificial y Computación Avanzada*. Santiago de Compostela – España. Editorial: Fundación Alfredo Brañas.
- Russell, S. J., Norvig, P., Corchado, R. J. M., & Joyanes, A. L. (2004). *Inteligencia Artificial: Un enfoque moderno*. Madrid: Pearson Prentice Hall
- Utrilla, D. (2014). *Doctorado en Ingeniería: Sistema de Arquitectura Multisensorial para Supervisión y Seguridad Industrial Aplicando Tecnología de Inteligencia Artificial*. Lima – Perú. Universidad Nacional Mayor de San Marcos.