UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO

LICENCIATURA DE INGENIERO EN SOFTWARE

"AGRUPAMIENTO VIA CLASIFICACIÓN"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN SOFTWARE

PRESENTA:

JESÚS PATIÑO TUDELA

DIRECTOR DE TESIS:

DR. RENÉ ARNULFO GARCÍA HERNÁNDEZ

TIANGUISTENCO, ESTADO DE MÉXICO; JULIO 2016

RESUMEN

En la actualidad se han intentado adaptar técnicas de aprendizaje supervisado para ser utilizadas en aprendizaje no supervisado. Por ejemplo, los árboles de decisión no supervisados ó también se ha utilizado el algoritmo K-NN como algoritmo de agrupamiento, donde se utiliza la regla del k vecino más cercano para crear grupos.

Cuando no se conocen las clases del aprendizaje supervisado, el investigador tiene que realizar de forma manual la creación de las clases de acuerdo a su amplio conocimiento del tema para realizar dicha clasificación. En este sentido, el aprendizaje no supervisado puede ser un paso previo de la clasificación, porque ayuda a obtener la muestra de entrenamiento que posteriormente se utilizará para la clasificación de nuevos objetos. Aunque en principio esto podría parecer lógico, en la práctica no es así, ya que el algoritmo específico de aprendizaje supervisado que se vaya a utilizar tiene un criterio diferente al algoritmo utilizado por el aprendizaje no supervisado para generar las clases.

Entonces, el problema es ¿cómo generar aprendizaje supervisado y no supervisado utilizando el mismo criterio?

El método que se propone en este trabajo es la realización de un algoritmo de agrupamiento basado en un algoritmo de genético, el cual, a su vez sea dirigido por un algoritmo de clasificación.

Los operadores que se realizaron o se adaptaron fueron el operador de selección, el operador de cruza, el operador de mutación y el módulo de la función de aptitud.

De acuerdo a la experimentación que se realizaron, se puede afirmar que es factible la idea presentada en esta tesis. Ya que los resultados son favorables e interesantes.

CONTENIDO

INDICE DE FIGURAS ____ **VII**

INDICE DE TABLAS ____ **VIII**

CAPÍTULO I ANTECEDENTES ____ **- 1 -**

1.1 INTRODUCCIÓN ____ - 1 -

1.2 PLANTEAMIENTO DEL PROBLEMA ____ - 5 -

1.3 OBJETIVOS ____ - 6 -

1.4 JUSTIFICACIÓN ____ - 6 -

1.5 HIPÓTESIS ____ - 7 -

1.6 ORGANIZACIÓN DE LA TESIS ____ - 7 -

CAPÍTULO II MARCO TEÓRICO ____ **- 9 -**

2.1 INTRODUCCIÓN ____ - 9 -

2.2 RECONOCIMIENTO DE PATRONES ____ - 10 -

2.3 APRENDIZAJE ____ - 12 -

2.4 TIPOS DE APRENDIZAJE ____ - 13 -

2.4.1 Supervisado ____ *- 13 -*

2.4.2 No supervisado ____ *- 13 -*

2.5 APRENDIZAJE SUPERVISADO ____ - 14 -

2.5.1 Algoritmos de clasificación ____ *- 14 -*

2.5.2 Criterios de clasificación ____ *- 16 -*

2.5.3 Medidas de evaluación ____ *- 16 -*

2.6 APRENDIZAJE NO SUPERVISADO ____ - 17 -

2.6.1 Algoritmos de agrupamiento ____ *- 18 -*

2.6.2 Tipos de agrupamiento ____ *- 18 -*

2.6.3 Medidas de evaluación ____ *- 21 -*

2.7 ALGORITMOS GENÉTICOS ____ - 21 -

2.8 RESUMEN ____ - 24 -

CAPÍTULO III ESTADO DEL ARTE ____ **- 26 -**

3.1 INTRODUCCIÓN ____ - 26 -

3.2 TRABAJOS RELACIONADOS - 26 -
3.3 RESUMEN - 32 -

CAPÍTULO IV PROPUESTA DE SOLUCIÓN - 34 -

4.1 INTRODUCCIÓN - 34 -
4.2 DESARROLLO DE LA PROPUESTA DE SOLUCIÓN - 34 -
4.2.1 Creación de la población inicial - 34 -
4.2.2 Evaluación de la población - 35 -
4.2.3 Condiciones de Paro - 37 -
4.2.4 Selección - 37 -
4.2.5 Cruza - 38 -
4.2.6 Mutación - 41 -
4.3 RESUMEN - 42 -

CAPÍTULO V EXPERIMENTACIÓN - 44 -

5.1 INTRODUCCIÓN - 44 -
5.2 BASES DE DATOS UTILIZADAS EN LOS EXPERIMENTOS - 45 -
5.3 PRIMER EXPERIMENTO - 45 -
5.3.1 Objetivo del primer experimento - 46 -
5.3.2 Diferentes valores de k - 46 -
5.3.3 Diferente mutación - 47 -
5.3.4 Conclusiones del primer experimento - 47 -
5.4 SEGUNDO EXPERIMENTO - 48 -
5.4.1 Objetivo del segundo experimento - 48 -
5.4.2 Diferentes valores de k - 49 -
5.4.3 Diferente mutación - 50 -
5.4.4 Diferente operador de selección - 51 -
5.4.5 Diferente Mutación - 52 -
5.4.6 Tamaño del operador por Torneo Selección(k) - 52 -
5.4.7 Conclusiones del segundo experimento - 53 -
5.5 TERCER EXPERIMENTO - 55 -
5.5.1 Objetivo del tercer experimento - 55 -
5.5.2 Diferentes Generaciones - 55 -
5.5.3 Conclusiones del tercer experimento - 56 -

5.6 Cuarto Experimento, los mejores resultados - 57 -
5.6.1 Objetivo del cuarto experimento - 57 -
5.6.2 Parámetros de los experimentos - 57 -
5.6.3 Conclusiones del tercer experimento - 61 -
5.7 Resumen - 62 -

CAPÍTULO VI CONCLUSIONES, APORTACIONES Y TRABAJOS FUTUROS - 66 -

6.1 Conclusiones - 66 -
6.2 Aportaciones - 66 -
6.3 Trabajos futuros - 67 -

ANEXOS - 68 -

Anexo A1: Diagrama de flujo del Algoritmo General - 68 -
Anexo A2: Diagrama de flujo de la Función de Aptitud - 69 -
Anexo A3: Diagrama de flujo de la Validación Cruzada *(Cross Validation)* - 70 -
Anexo A4: Diagrama de flujo de los K-vecino más cercanos - 71 -
Anexo A5: Diagrama de flujo del Operador de Selección por Ruleta - 72 -
Anexo A6: Diagrama de flujo del Operador de Selección por Torneo - 73 -
Anexo A7: Diagrama de flujo del Operador de Cruza por Intersección - 74 -
Anexo A8: Diagrama de flujo del Métodos de Cruza Intersección - 75 -
Anexo A9: Diagrama de flujo de Intercambio de Grupos - 76 -
Anexo A10: Diagrama de flujo de Re-etiquetación - 77 -
Anexo A11: Diagrama de flujo de la Validación - 78 -
Anexo A12: Diagrama de flujo del Operador de Mutación por Intercambio - 79 -
Anexo B1: Base de Datos sintética - 80 -
Anexo B2: Base de Datos sintética - 82 -

REFERENCIAS - 83 -

INDICE DE FIGURAS

FIGURA 2.2-1 ETAPA DE UN SISTEMA DE RECONOCIMIENTO DE PATRONES - 11 -

FIGURA 2.3-1 PROCESO DE APRENDIZAJE - 12 -

FIGURA 2.5-1 RUIDO EN CLASIFICACIÓN - 15 -

FIGURA 2.5-2 MEDIDA PARA CALCULAR DISTANCIAS ENTRE OBJETOS - 16 -

FIGURA 2.6-1 DENDOGRAMA AGLOMERATIVO - 19 -

FIGURA 2.6-2 DENDOGRAMA DIVISIVO - 19 -

FIGURA 2.7-1 REPRESENTACIÓN DE UN CROMOSOMA - 22 -

FIGURA 2.7-2 EJEMPLO DE CRUZA - 23 -

FIGURA 2.7-3 EJEMPLO DE MUTACIÓN DE UN PUNTO - 24 -

FIGURA 4.2-1 ALGORITMO GENÉTICO - 35 -

FIGURA 4.2-2 REPRESENTACIÓN DEL INDIVIDUO - 36 -

FIGURA 4.2-3 EJEMPLO DE SELECCIÓN POR RULETA - 38 -

FIGURA 4.2-4 OBTENCIÓN DE ÍNDICES - 39 -

FIGURA 4.2-5 OBTENCIÓN DE MATRIZ DE INTERSECCIONES - 40 -

FIGURA 4.2-6 BUSCAR EN NÚMERO MAYOR - 40 -

FIGURA 4.2-7 ELIMINACIÓN DE LA COLUMNA Y RENGLÓN - 41 -

FIGURA 4.2-8 CRUZA POR INTERSECCIÓN - 41 -

FIGURA 4.2-9 MUTACIÓN POR INTERCAMBIO - 42 -

FIGURA 5.3-1 GRÁFICA DE LA FUNCIÓN DE APTITUD CON DIFERENTES VALORES DE K - 47 -

FIGURA 5.3-2 GRÁFICA DE DIFERENTES MUTACIONES - 48 -

FIGURA 5.4-1 GRÁFICA DE DIFERENTES VALORES DE K, CON LA BASE DE DATOS IRIS Y SELECCIÓN POR RULETA. - 49 -

FIGURA 5.4-2 GRÁFICA DE DIFERENTES VALORES DE K, CON LA BASE DE DATOS IRIS Y SELECCIÓN POR TORNEO. - 50 -

FIGURA 5.4-3 GRÁFICA DE LOS DIFERENTES OPERADORES DE SELECCIÓN - 51 -

FIGURA 5.4-4 GRÁFICA VARIANDO EL PARÁMETRO DE MUTACIÓN - 52 -

FIGURA 5.4-5 GRÁFICA CON DIFERENTES VALORES DE K-SELECT - 53 -

FIGURA 5.4-6 GRÁFICA CON RESULTADOS DE DIFERENTES OPERADORES DE SELECCIÓN - 54 -

FIGURA 5.5-1 GRÁFICA CON DIFERENTE NÚMERO DE GENERACIONES - 56 -

FIGURA 5.6-1 GRÁFICA DEL PRIMER EXPERIMENTO - 58 -

FIGURA 5.6-2 GRÁFICA DEL SEGUNDO EXPERIMENTO - 59 -

FIGURA 5.6-3 GRÁFICA DEL TERCER EXPERIMENTO - 61 -

INDICE DE TABLAS

TABLA 4.2-1 FUNCIÓN DE APTITUD _____ - 36 -
TABLA 4.2-2 FUNCIÓN DE SIMILITUD PARA DATOS NUMÉRICOS _____ - 37 -
TABLA 4.2-3 EJEMPLO DE OPERACIÓN DE CRUZA _____ - 39 -
TABLA 5.3-1 PRIMER EXPERIMENTO _____ - 46 -
TABLA 5.4-1 PARÁMETROS DEL SEGUNDO EXPERIMENTO _____ - 49 -
TABLA 5.4-2 MEJORES PARÁMETROS PARA EL ALGORITMO GENÉTICO _____ - 54 -
TABLA 5.5-1 MEJORES PARÁMETROS, GENERACIONES _____ - 56 -
TABLA 5.6-1 MEJORES PARÁMETROS PARA EL ALGORITMO GENÉTICO _____ - 58 -
TABLA 5.6-2 PARÁMETROS DEL SEGUNDO EXPERIMENTO _____ - 59 -
TABLA 5.6-3 PARÁMETROS DEL TERCER EXPERIMENTO _____ - 60 -
TABLA 5.7-1 RESULTADOS DEL MÉTODO PROPUESTO _____ - 63 -

CAPÍTULO I ANTECEDENTES

1.1 Introducción

Desde el comienzo de la humanidad, y al paso de su evolución, el hombre ha tenido que aprender a reconocer su entorno, animales, plantas, personas, etc. para poder sobrevivir. Conforme sigue pasando el tiempo, el reconocimiento de patrones se vuelve más complejo.

El reconocimiento es algo tan natural para las personas, pero también algo complejo. Por ejemplo, un niño desde pequeño y conforme va creciendo comienza a reconocer personas, colores, sabores, olores, textos escritos, piezas de música, palabras, etc. Conforme el niño va conociendo nuevos objetos es capaz de relacionarlos con otros objetos parecidos, ya sea por su color, forma o tamaño. Cuando al niño se le presenta un nuevo objeto que no conoce, crea una clasificación del nuevo objeto. Por ejemplo, cuando un niño es capaz de acomodar un libro en un grupo de libros, tuvo que utilizar las características del libro, como su color, tamaño, forma, para reconocer en donde debería situar el libro.

Con la llegada de las computadoras y las ventajas que trajo consigo en la automatización de procesos y tareas, se busca que, con su velocidad de procesamiento, realice el proceso de reconocimiento de patrones de manera automática y con mayor rapidez en grandes cantidades de objetos.

El reconocimiento de patrones es una disciplina científica que se encarga de clasificar en clases, categorías o grupos, un conjunto de objetos (Theodoridis & Koutroumbas, 2003). Dentro del reconocimiento de patrones encontramos dos formas de realizar esta clasificación. El aprendizaje supervisado (Clasificación) y el aprendizaje no supervisado (Agrupamiento)

La primera forma de catalogar objetos es el aprendizaje supervisado o clasificación, donde cada objeto ya cuenta con una clasificación previa. A diferencia del aprendizaje no supervisado, los nuevos objetos son comparados con los que ya están previamente clasificados y se les asigna la clasificación a la que pertenecen (Carrasco & Martínez, 2011). En este aprendizaje ya no se descubre conocimiento.

Para el problema de clasificación existen varios métodos, destacando algunos como las redes neuronales (Haykin, 1998), árboles de decisión (García, 2012), vecinos más cercanos (Morales, et al., 2008), máquinas de soporte vectorial (Igel, 2002), entre otros.

Para saber qué tan preciso es un determinado método de clasificación se utiliza la validación cruzada. La validación cruzada es la que evalúa los resultados y lo hace dividiendo la muestra de objetos en dos partes, la parte de entrenamiento donde aprende a qué clase pertenece cada objeto y la parte de prueba, en la que se comprueba qué tan preciso y exacto es el aprendizaje (Talavera & Rodríguez, 2008). Todos los objetos de la muestra son utilizados para entrenar y probar. De esta forma, el error se obtiene con el promedio del error de los n experimentos que se realizan (Moreno, 2004).

La segunda forma de catalogar objetos es el aprendizaje no supervisado o también conocida como agrupamiento (*Clustering*), donde a partir de una muestra de objetos, hay que encontrar los grupos a los que pertenecen los objetos. La regla es que los grupos generados, deben tener características muy parecidas, entre los miembros del mismo grupo; pero muy diferentes a los de otros grupos (Bokan, et al., 2011), generalmente se utiliza para descubrir conocimiento.

Por su parte para resolver el problema de aprendizaje no supervisado o agrupamiento se han desarrollado diversos métodos o técnicas como son: el agrupamiento jerárquico (Hernández, 2006), el agrupamiento de particionamiento (Berzal, 1999), el agrupamiento basado en densidad por mencionar algunos (Bokan, et al., 2011).

La forma en que se evalúa el aprendizaje no supervisado es a través de los índices de validación, los cuales determinan qué tan buenos son los grupos que se forman (Ming-Hseng Tseng, *et al*., 2010). Los índices de validación se dividen en dos, los índices de validación internos y los índices de validación externos (Sabau, 2012).

Los índices de validación internos son los que evalúan qué tan cercanos están los elementos del grupo unos de otros. Por ejemplo, el índice de Davies-Boulding (Desgraupes, 2013), el índice de Silhouette (Desgraupes, 2013), entre otros. Los índices de validación externos son los que se encargan de medir qué tan distantes están los elementos de un grupo de otro. Por ejemplo, el índice de Dunn (Desgraupes, 2013), índice de Rand (Desgraupes, 2013); por mencionar algunos.

Como se mencionó, el agrupamiento no tiene información *a priori* de a qué grupo pertenecen los elementos, por lo que los resultados pueden no ser satisfactorios para el usuario (Ingaramo, *et al*., 2007). Para trabajar, el algoritmo de aprendizaje no supervisado agrupa tratando de optimizar un criterio. A partir de esto, el usuario debe comenzar con un proceso repetitivo de exploración tratando de buscar el mejor algoritmo de agrupamiento y las mejores combinaciones de parámetros que éste pueda tener para el usuario. En este sentido, no hay un buen agrupador para todos los problemas no supervisados, es decir, ni uno es más bueno o más malo. En otras palabras, el aprendizaje no supervisado depende de la muestra de objetos y del problema que el usuario intenta resolver, ya que los resultados de cada agrupador son diferentes.

Suponiendo que el usuario encontrara una agrupación satisfactoria, el problema sería aún mayor, si desea utilizar esa agrupación como muestra de entrenamiento para clasificar nuevos objetos. Esto sucede porque cada algoritmo de clasificación tiene un criterio diferente, además de que encontrar la mejor combinación de parámetros para dicho algoritmo es otro problema.

Otro problema al que se enfrentan los algoritmos de agrupamiento es generar grupos de igual tamaño, grupos homogéneos (Moreno, *et al*., 2010). Es decir, que cada grupo

contenga casi los mismos elementos que otros grupos, y esto nos lleva a tener que estar realizando varias iteraciones para encontrar el mejor agrupamiento y obtener buenos resultados, que sean usables es decir esto también llevaría algo de tiempo.

Un ejemplo sencillo para poder entender el problema sería el siguiente.

Suponiendo que un niño pequeño que no sabe leer quisiera organizar un conjunto de libros en una biblioteca especializada entonces los podría agrupar por colores o tamaños. En cambio, si la organización la realizará un bibliotecario la realizaría por temas, o por disciplina.

Ambos casos serían una buena organización, pero son muy distintos entre ambos porque cada uno tiene su criterio para organizar. Por lo tanto, el agrupamiento depende de la necesidad que tenga el usuario.

Entonces el problema es cómo generar aprendizaje supervisado y no supervisado utilizando el mismo criterio.

Se han intentado adaptar técnicas de aprendizaje supervisado para ser utilizadas en aprendizaje no supervisado, como son árboles de decisión no supervisados (Gutierrez, *et al*., 2012), donde no se tienen en cuenta las clases, porque los objetos no están etiquetados y para cada nodo del árbol, es dividido de acuerdo a un índice de validación del agrupamiento. También se ha utilizado el método k-NN para agrupamiento, donde se utiliza la regla del k vecino más cercano (Pascual, *et al*., 2007).

Como consecuencia de lo anterior el agrupamiento ha utilizado un método de búsqueda y optimización que se encuentra en la computación evolutiva y son los algoritmos genéticos (Gestal, 2010).

Los algoritmos genéticos permiten buscar los parámetros que ayuden a mejorar los grupos que forman o su mejor homogeneidad en los elementos de cada grupo.

Los algoritmos genéticos simulan la evolución natural (Kuri & Galaviz, 2007) donde dada una población inicial se selecciona los individuos más aptos que se cruzán para que se reproduzcan. A partir de esto se pueden obtener nuevas generaciones, mejores que las anteriores. Cada uno de los individuos de la población son evaluados mediante una función de aptitud (Pajares & Santos, 2006), la que indica qué tan apto es el individuo. El individuo pasa por operadores genéticos como la selección, cruza y mutación.

El operador de selección se encargará de seleccionar a los dos individuos más aptos de la población. El operador de cruza es el encargado de mezclar los genes de los individuos que fueron seleccionados para que se puede obtener hijos más fuertes (López, 2010). El operador de mutación es donde puede haber alteraciones en un gen o en varios genes del hijo, ésta puede ser de forma aleatoria y pueden ayudar a mejorar la función de aptitud del individuo (Kuri & Galaviz, 2007).

Con el uso de estos algoritmos podemos resolver alguno de los problemas del agrupamiento, porque, aunque no garantizan obtener la mejor solución, si garantizan obtiene una de las mejores soluciones.

1.2 Planteamiento del problema

Se puede ver que cuando no se conocen las clases del aprendizaje supervisado, el aprendizaje no supervisado puede ser un paso previo del aprendizaje supervisado, porque es donde se obtiene la muestra de entrenamiento que sirve para la clasificación de los nuevos objetos que lleguen, los cuales serán clasificados. Aunque en principio esto podría llegar a ser cierto no es así en la práctica, ya que el algoritmo que se utilice como algoritmo de aprendizaje supervisado llámese (*k* vecinos, arboles de decisión, etc.) que se vaya a utilizar tiene un criterio diferente, al algoritmo utilizado por el aprendizaje no supervisado *(k means*, DBMSCAN, o cualquier otro).

Es decir, el problema es cómo encontrar la mejor partición de un conjunto de elementos, con el objetivo de encontrar las clases para formar una muestra de entrenamiento dado un algoritmo de aprendizaje supervisado.

1.3 Objetivos

General:

- Adaptar un algoritmo de agrupamiento basado en un algoritmo genético, el cual a su vez sea dirigido por un algoritmo de clasificación.

Específicos:

- Saber si es posible utilizar el mismo criterio para agrupar y para clasificar.
- Realizar un marco de trabajo que puedan utilizar los datos de un agrupamiento en cualquiera de los algoritmos de clasificación que existen.
- Implementar la metodología incremental.
- Probar con bases de datos clásicas y evaluar sus resultados con medidas internas y externas.
- Adaptar y crear operadores genéticos para el problema planteado.

1.4 Justificación

Hasta el momento, un problema de aprendizaje no supervisado, se resuelve realizando muchas experimentaciones buscando un mejor agrupamiento. Cabe señalar que el agrupamiento se podría ver como un paso previo de la clasificación, pero en la mayoría de las veces no se utiliza como tal, ya que la principal diferencia es como se llevan a cabo los procesos de agrupar y clasificar los objetos.

Un ejemplo donde podríamos ver el uso de nuestro método, sería en el desbalance de clases. Por ejemplo, en el trabajo de (Osorio, 2013) donde tiene una muestra de alumnos clasificados como dados de baja o no. Esta muestra de entrenamiento tiene un desequilibrio en el número de alumnos lo que provoca que, a la hora de clasificar un

nuevo alumno, se clasifique como la clase mayoritaria de un ser dado de baja. Esto deteriora de manera importante la efectividad del clasificador. El método que se propone en esta tesis podría ayudar a balancear la muestra de datos aumentando la precisión.

1.5 Hipótesis

Si dos puntos u objetos que están cercanos en una muestra de objetos normalmente pertenecen al mismo grupo (hablando de agrupamiento) o a la misma clase (hablando de clasificación) de una muestra de objetos (Chapelle, et al., 2006), entonces un algoritmo genético puede utilizar a un algoritmo de clasificación para que a partir de una clasificación propuesta indique qué tan preciso es su clasificación. De esta manera se busca maximizar la precisión de clasificación para que el algoritmo genético pueda encontrar los grupos de una muestra de objetos de manera que, los objetos de un grupo sean muy parecidos entre ellos, pero muy diferentes de otros grupos.

1.6 Organización de la tesis

En el capítulo I se muestran las causas que me motivaron a tomar este camino. En la introducción se ven las causas del problema, con el objetivo de saber cómo podemos resolverlo, así como qué se pretende alcanzar a lo largo de esta tesis y qué se plantea cumplir.

A grandes rasgos se describen cómo funcionan los algoritmos de aprendizaje supervisado y no supervisado, los problemas a los que se enfrentan y el problema que queremos tratar de la diferencia de criterios el aprendizaje.

También se ve la utilización de los algoritmos genéticos para encontrar mejores soluciones a problemas de búsqueda y optimización.

En el capítulo II se describen los principales conceptos para realizar esta tesis como son aprendizaje supervisado y no supervisado. Los algoritmos de agrupamiento o

clustering, los algoritmos de clasificación basados en instancias (k vecinos más cercanos) algoritmos genéticos. Para cada uno de estos algoritmos se verá su definición, su funcionamiento cómo es que son evaluados cada uno de ellos.

En el capítulo III, se describirán, algunos trabajos que tratan de solucionar problemas parecidos al nuestro, o que tienen alguna relación a nuestro problema, cómo han resuelto su problema, cómo lo enfocaron, la solución que le dieron y qué técnicas utilizaron para lograr llegar a su solución. Con esta información se tiene panorama de cómo orientar y dar solución al problema planteado.

En el capítulo IV se expone nuestra propuesta de solución al problema, así como el funcionamiento y posible comportamiento que deben llevar acabo cada una de las partes relacionadas.

En el capítulo V se verán los resultados que se han obtenido, las pruebas que se han realizado, el objetivo de cada prueba y a la conclusión que se ha llegado con cada una de estas pruebas, lo que permite verificar el método que se ha propuesto realmente está funcionando.

Como capítulo final, en el capítulo VI se analizan las conclusiones a las que se han llegado, si se obtuvieron los resultados deseados y sobre todo si se puede contestar nuestra pregunta de investigación con todo lo que se ha realizado.

Al final de esta tesis se encontrarán los anexos y las referencias correspondientes utilizadas en esta tesis.

CAPÍTULO II MARCO TEÓRICO

2.1 Introducción

En este capítulo se describen los conceptos teóricos que se utilizan en esta tesis su definición, funcionamiento y la relación entre ellos.

Lo primero que se ve es el reconocimiento de patrones, que es una ciencia encargada de etiquetar objetos en categorías o clases de acuerdo a las propiedades de los elementos. Mediante el aprendizaje automático, el reconocimiento de patrones, va aprendiendo y mejora conforme a la experiencia que va adquiriendo. Para lograr su aprendizaje lo hace mediante dos formas: uno es el aprendizaje supervisado, o también llamado clasificación, y el otro es el aprendizaje no supervisado o agrupamiento.

Dentro del aprendizaje supervisado encontramos los algoritmos basados en instancias, los cuales son un tipo de aprendizaje perezoso. Para su funcionamiento, los algoritmos basados en instancias utilizan un conjunto de elementos ya etiquetados (al cual se le llama conjunto de entrenamiento) para clasificar nuevos datos. El proceso busca, de los objetos ya etiquetados, a los más parecidos al nuevo objeto, a partir de los que sean más parecidos, se etiqueta el nuevo objeto. Es por ello que el algoritmo k vecinos más cercanos es uno de los algoritmos más utilizados y simple.

El otro enfoque del aprendizaje es el no supervisado, en donde podemos encontrar muchas formas para realizar este proceso, los algoritmos más utilizados son: el agrupamiento de particionamiento y el jerárquico.

Finalizaremos este capítulo con los algoritmos genéticos, que será el mecanismo utilizado como el algoritmo de búsqueda y optimización para resolver el problema planteado.

2.2 Reconocimiento de patrones

El reconocimiento de patrones es una disciplina científica que tiene como objetivo clasificar o identificar objetos en clases o grupos, principalmente de acuerdo a sus propiedades o características (Vazquez, 2008). Por lo general, el reconocimiento de patrones se utilizaba en aplicaciones para clasificar imágenes o señales en forma de onda. Su principal área de investigación era, hasta el año de 1960, la estadística (Kunzmann, 2005). Algunos ejemplos de su uso eran la distribución estadística, multivariada, en la que ofrecen un modelo adecuado para la variabilidad de las representaciones de patrones (Aja, 2005), otra era la Teoría de la Decisión Estadística, donde el punto es ver sí un patrón pertenece o no a una clase de patrones.

Los enfoques más populares que ha seguido el reconocimiento de patrones según (Carrasco & Martínez, 2011), (Alba & Cid, 2006) y (Yañez, 2008) son:

- Reconocimiento estadístico de patrones:

  Enfoque basado en la teoría de probabilidad y estadísticas, supone que se tiene un conjunto de medidas numéricas con distribuciones de probabilidad conocidas o estimadas y a partir de esta se comienza el reconocimiento de patrones.

- Reconocimiento sintáctico-estructural de patrones:

  Enfoque encargado de estudiar la estructura, así como la relación de los objetos a clasificar, usa teoría de lenguajes formales, gramáticas, teorías de autómatas, etc.

- Redes neuronales:

  En este enfoque se utilizan redes neuronales para el reconocimiento de patrones, las cuales son entrenadas para dar una cierta respuesta cuando se le presentan determinados valores.

- Reconocimiento lógico combinatorio:

  Este enfoque tiene como principal característica que los objetos deben ser lo más cercanos a la realidad del mismo y los objetos se describen por una combinación de rasgos numéricos y no numéricos.

Los problemas que principalmente pueden resolver el reconocimiento de patrones son las técnicas de selección de atributos y prototipos, aprendizaje supervisado y aprendizaje no supervisado (Carrasco & Martínez, 2011), aunque algunos autores como (Vazquez, 2008), ponen una cuarta clasificación el aprendizaje parcial o parcialmente supervisado.

Los elementos básicos del reconocimiento de patrones que se estudian se les conoce con el término patrón. Un patrón se describe como una descripción estructural o cuantitativa de un objeto o de alguna otra entidad de interés que involucra a los elementos de la muestra de objetos (Kittler, 2002), (Romo R., *et al.*, 2007).

Según (Alba & Cid, 2006) otro de los elementos que debe tener el reconocimiento de patrones son: el patrón, reconocimiento o clasificación, clase, clase de rechazo, extractor de características, clasificador.

Las etapas en un sistema de Reconocimiento de Patrones podrían dividirse en, la parte en que adquiere los objetos del universo, seguido de la parte donde se extraen las características y, finalmente, la parte donde se toma la decisión de clasificación del patrón (Kittler, 2002), ver figura 2.2-1.

Figura 2.2-1 Etapa de un sistema de reconocimiento de patrones

2.3 Aprendizaje

El aprendizaje, se puede explicar cómo la información que recibe un organismo, a partir de la cual se toman decisiones que permiten que se adapte a un entorno para que pueda sobrevivir (Moreno, 2004).

El aprendizaje máquina se define como el proceso de adaptación de una máquina a nuevas circunstancias (Russell & Norvig, 1996), (Bedoya, 2011) mediante el cual, un sistema mejora la realización de las tareas que realiza el humano; siendo una característica de los sistemas inteligentes (López, 2010). Por lo general se obtienen buenos resultados o mejores que los que obtendría un humano (Alvarado, 2010).

El funcionamiento es el siguiente: se le enseña que debe aprender (ejemplos), el objetivo que tiene que cumplir (que tipo de aprendizaje utilizará) y qué resultados esperamos obtener del proceso ver figura 2.3-1.

Dentro del aprendizaje máquina podemos englobar el aprendizaje en dos tipos de aprendizaje: el aprendizaje supervisado y el aprendizaje no supervisado.

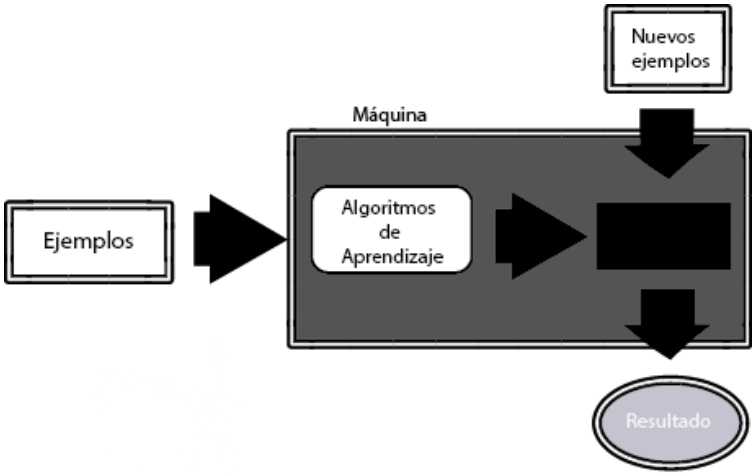


Figura 2.3-1 Proceso de aprendizaje

2.4 Tipos de aprendizaje

Aunque los principales tipos de aprendizaje que existen son los anteriormente mencionados, también existen otros tipos de aprendizaje (López, 2010), (Gómez, *et al*., 1994) (García & Gómez, 2009) importantes como son:

- Aprendizaje Hebbiano
- Aprendizaje Competitivo
- Aprendizaje Min-Max
- Aprendizaje de corrección de errores
- Aprendizaje por reforzamiento
- Aprendizaje Estocástico
- Aprendizaje Genético
- Aprendizaje Analítico
- Aprendizaje Conexionista

Otro tipo aprendizaje es el aprendizaje semi-supervisado, es un método que permite utilizar una pequeña muestra de datos etiquetados, como no etiquetados de mayor tamaño, para formar un conjunto de entrenamiento (Pascual, *et al*., 2014).

2.4.1 Supervisado

Es el aprendizaje supervisado es una técnica en la cual se asigna un nuevo objeto a un grupo ya definido. La naturaleza del aprendizaje supervisado es que sea capaz de predecir el valor correspondiente de un objeto nuevo, después de haber tenido ejemplos de referencia (datos de entrenamiento) (Alonso Romero & Calonge Cano, 2008).

2.4.2 No supervisado

El aprendizaje no supervisado no tiene datos de entrenamiento para poder ser agrupado y se asigna de acuerdo a su similitud, se utiliza cuando los patrones son cambiantes (Porta Zamorano, 2005).

2.5 Aprendizaje supervisado

El aprendizaje supervisado, o mejor conocido como clasificación supervisada (Carrasco & Martínez, 2011) cuenta con un profesor, que se encarga de proporcionar una categoría a un nuevo objeto, y lo hace para cada patrón de un conjunto de entrenamiento (Duda, et al., 2000), (Díaz, 2007). Su objetivo es clasificar nuevos objetos a partir de un conjunto de datos. Este conjunto de datos tiene dos etapas. La primera es el conjunto de entrenamiento, donde se aprenden y la otra, es la etapa de prueba (Malagón, 2003) en la que se ve que también aprendió el clasificador.

2.5.1 Algoritmos de clasificación

Dentro de las técnicas para resolver problemas de aprendizaje supervisado encontramos: los clasificadores Bayesianos, los árboles de decisión (García, 2012), redes neuronales (Corso, 2009), máquinas de soporte vectorial, algoritmos basados en instancias, algoritmos de votación, clasificadores basados en patrones, clasificadores basados en conjuntos de representaciones.

Los algoritmos basados en instancias, también conocidos como *lazy learning o memory learning* (Bedoya, 2011), almacenan objetos de entrenamiento y cuando se quiere clasificar un nuevo objeto, se recuperan los objetos más cercanos. Es decir, extrae los objetos más parecidos al nuevo objeto para clasificarlo (Morales, et al., 2008).

Siendo la clasificación el proceso que mayor tiempo consume (Morales, 2012) (García & Gómez, 2009). El ejemplo más destacado de este tipo de algoritmos es el K-Vecinos más cercano, pero existe otros como: Razonamiento basado en casos, la regresión lineal ponderada local (Bedoya, 2011).

El algoritmo de k vecinos más cercanos (k-NN), se divide en dos partes: la parte del entrenamiento y la parte de la clasificación (Alonso, *et al*., 2007), (Moreno, 2004). K-NN es una técnica que, dada una instancia a clasificar, se obtienen los k vecinos, y aplicándoles una función de distancia, determinará a qué clase pertenece de acuerdo a

los vecinos más cercanos (Witten, *et al.*, 2011). Es decir, esta técnica solo recuerda los ejemplos que se vieron en la etapa de entrenamiento. Los nuevos casos se clasifican según el comportamiento del dato más cercano (Morales, *et al*., 2008) (Alonso, *et al*., 2007).

Este algoritmo tiene tres propiedades, la primera propiedad es que se trata de un algoritmo de aprendizaje perezoso. La segunda propiedad es que clasifica nuevos objetos comparando con objetos similares e ignora los que son distintos. La tercera propiedad es representa a los objetos como puntos de valores reales en un espacio euclidiano de n dimensiones (Rodríguez, et al., 2010)

Para evitar el ruido que se puede presentar en el k-vecinos más cercanos, al clasificar nuevos objetos, se debe aumentar el número de vecinos (k), de este modo, al aumentar k, se asocia más rápido el nuevo dato a los elementos más representativos o con mayor presencia en el espacio (Mora, et al., 2008). Cabe señalar que el valor de k siempre debiera de ser un número impar (De la O, 2007) ver figura 2.5-1.

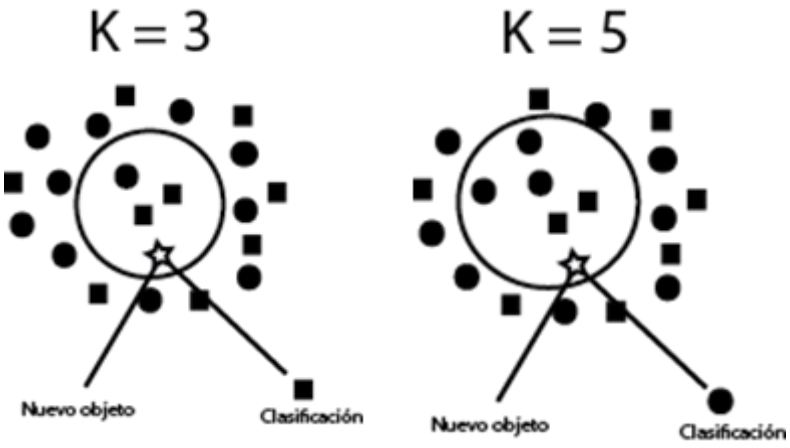


Figura 2.5-1 Ruido en clasificación

En este punto, podríamos decir que aumentar el número de vecinos o *k* al máximo no siempre es bueno, porque tiene que haber un balance entre su valor, ya que al aumentar su valor podríamos tener mayor presencia de una clase. Es decir, al algoritmo

k vecinos le afecta el desbalance de clases, lo que causa errores de confiabilidad al clasificar los objetos (Primitivo, 2011) (Hernández, 2006).

Esta clasificación también tiene sus desventajas ya que el costo de clasificar nuevos objetos suele ser muy alto (Bedregal, 2008), esto pasa por que el proceso se hace cuando se está clasificando y no cuando se está entrenando.

2.5.2 Criterios de clasificación

Se debe definir un criterio para medir la similitud entre objetos. Por ejemplo, se puede utilizar la distancia entre los elementos de la muestra, que están cercanos a un nuevo objeto, las más conocidas son: la distancia Euclidiana (Alonso Romero & Calonge Cano, 2008) (Moujahid, et al., 2008).

$$D_{Euclideana}\,(\vec{x},\vec{y}) = \sqrt{(\vec{x},\vec{y})^t(\vec{x},\vec{y})}$$

$$D_{Mahalanobis}\,(\vec{x},\vec{y}) = \sqrt{(\vec{x},\vec{y})^t \sum\nolimits_{-1} (\vec{x},\vec{y})}$$

X → $x\{x_1, x_2 \dots x_n\}$ objeto con características

Y→y$\{y_1, y_2 \dots y_n\}$ objeto con características

Figura 2.5-2 Medida para calcular distancias entre objetos

2.5.3 Medidas de evaluación

La validación cruzada evalúa y compara la eficiencia y precisión de los algoritmos de aprendizaje (Refaelzadeh, et al., 2008) , dividiendo el conjunto de datos en dos partes. La primera parte, la utiliza tanto para aprendizaje como para entrenamiento y la otra la utilizada para la validación.

La validación cruzada consiste en dividir *n* particiones del conjunto de datos en dos conjuntos, el de entrenamiento y el de prueba (Corso, 2009) (Moreno, 2004). De forma que todo el conjunto de objetos es utilizado tanto para el entrenamiento como para la prueba y es dividido de forma aleatoria y en partes del mismo tamaño (De la O, 2007).

Este proceso se realiza *n* veces lo que permite usar todo el conjunto de datos para encontrar un error de validación (Morales, et al., 2008).

El error de validación es el promedio del error de los *n* experimentos. Este procedimiento se realiza hasta el número total del conjunto de datos, con el objetivo de escoger los parámetros que tengan un menor error en la validación; siendo estos parámetros los que tendrán mejor comportamiento ante datos desconocidos (Corso, 2009) (Moreno, 2004).

2.6 Aprendizaje no supervisado

En el aprendizaje no supervisado, agrupamiento, *cluster* o *clustering* por sus siglas en inglés (Cortijo, 2001), no tiene como tal un supervisor para agrupar los nuevos objetos como pasa en el aprendizaje supervisado (Duda, *et al*., 2000). Este aprendizaje, forma grupos o clases naturales de patrones, sin información *a priori,* es decir patrones no etiquetados (Berzal, 1999) (Cortijo, 2001). Es una técnica muy útil para descubrir conocimiento oculto en conjunto de datos.

El agrupamiento es una tarea de análisis exploratorio (Bedregal, 2008), se ha utilizado en áreas de reconocimiento de patrones, minería de datos, estadística aplicada, segmentación de clientes, economía (Segmentación del mercado) biología (Agrupamiento de genes significativos), medicina (Segmentación de imágenes cerebrales mediante resonancia magnética), recuperación de información (Minería de datos), (García & Gómez, 2009), (Bedregal, 2008), (Witten, et al., 2011).

El aprendizaje supervisado consiste en dividir un conjunto de objetos en grupos, los grupos se crean de acuerdo a las características de los datos. Los grupos formados se caracterizan por contener objetos significativos, es decir, objetos que guardan relación entre sí, pero más diferentes los de otros grupos (Berzal, 1999) (García & Gómez, 2009). Para Cortijo (Cortijo, 2001) menciona prácticamente la misma definición de los autores anteriores, pero agrega que los grupos deben ser homogéneos.

También es importante que se obtenga un número pequeño de grupos, pero que el número de objetos por cada grupo sea mayor (Marin & Branch, 2008).

2.6.1 Algoritmos de agrupamiento

Los métodos de agrupamiento que existen son dos, los restringidos y los libres (Carrasco & Martínez, 2011).

- Restringidos: El algoritmo pide el número de grupos máximos que se van a formar o a encontrar del conjunto de datos (Cortijo, 2001) (Carrasco & Martínez, 2011).
- Libres: Es donde el número de grupos que se obtendrá no es conocido y el algoritmo tiene encontrar en cuántos grupos se va dividir el conjunto de datos (Carrasco & Martínez, 2011).

2.6.2 Tipos de agrupamiento

El problema del aprendizaje no supervisado ha sido muy estudiado y se han hecho un sin número de algoritmos para este aprendizaje, como: validación de agrupamiento, número de grupos, e imponer estructuras de datos (Ana, 2002).

Las clasificaciones más comunes para agrupamiento de objetos son: de particionamiento, jerárquicos y basados en densidad (Jiménez, 2011) (Pascual, et al., 2014) (Marin & Branch, 2008).

El agrupamiento jerárquico a su vez es dividido en dos tipos, algoritmos aglomérativos y divisivos (Jiménez, 2011).

Un agrupamiento jerárquico puede ser representado en forma de dendograma, el cual refleja cómo se van agrupando los objetos (Martínez, 2000) (Funes, 2008).

- El algoritmo aglomérativo o también llamado *bottom-up*, hace el agrupamiento, comenzando con un objeto por grupo, hasta que todos los objetos forman un

solo grupo (Carrasco & Martínez, 2011) (Bedregal, 2008) (Jiménez, 2011) ver figura 2.6-1

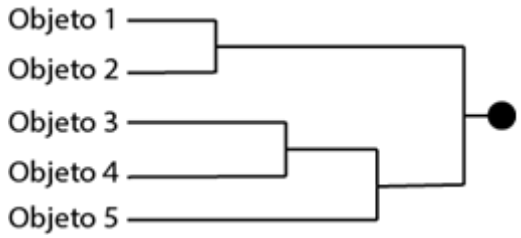


Figura 2.6-1 Dendograma Aglomerativo

- En el agrupamiento divisivo o *top-down*, están concentrados todos los objetos en un solo grupo y se van separando hasta formar grupos de un solo objeto (Martínez, 2000) (Carrasco & Martínez, 2011) (Jiménez, 2011) ver figura 2.6-2.

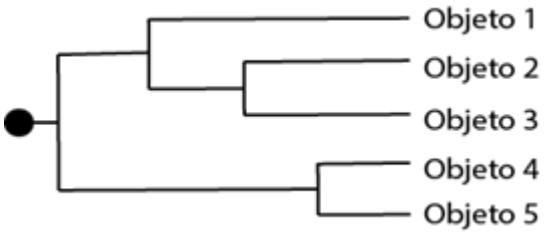


Figura 2.6-2 Dendograma Divisivo

Algunos ejemplos de este tipo de agrupamiento tenemos (Hernández, 2006): CURE, CHAMALEON, BIRCH, ROCK.

El agrupamiento particional o también llamado de optimización (Yolis, 2003) es un tipo de agrupamiento en el cual debe conocer el número de grupos que se quiere formar del conjunto de datos (Yolis, 2003), (Carrasco & Martínez, 2011). Aunque para Bedregal (Bedregal, 2008) puede o no ser especificado el valor de los grupos. A diferencia del agrupamiento jerárquico, el agrupamiento particional trabaja en un solo nivel, comienza de forma aleatoria y se va optimizando de acuerdo a una función objetivo (Jiménez, 2011), (Pascual, et al., 2007). El agrupamiento particional tiene gran ventaja en conjuntos de objetos de gran tamaño. El único inconveniente de estos algoritmos es que se debe conocer el número de grupos deseado. (Hernández, 2006).

El agrupamiento particional se define como, dado un conjunto de objetos en un espacio, determinar la partición de estos objetos en grupos, de modo que los elementos del grupo sean muy similares entre sí y diferentes de otros grupos (Bedregal, 2008) (Hernández, 2006).

Algunos algoritmos que emplean esta técnica son: CLARA (Clustering Large Applications), CLARANS (Clustering Large Applications base on Randomized), K-prototypes, K-modas y el más conocido el K-medias (Milone, et al., 2009).

Otro tipo de agrupamiento es el basado en densidad. Este tipo de algoritmo obtiene grupos basados en cuadrantes densos de objetos en un conjunto de datos que están separados por regiones de baja densidad (Bedregal, 2008) (Jiménez, 2011). Su utilidad yace en filtrar ruido y encontrar agrupamiento de diversas formas. El agrupamiento basado en densidad identifica grupos de forma arbitraria, robustos ante la presencia de ruido y escalables en un único recorrido del conjunto de datos (Bedregal, 2008). Los algoritmos basados en densidad solo pueden encontrar grupos esféricos y se les dificulta hallar grupos de formas diversas (Hernández, 2006). Estos algoritmos usan diversas técnicas para determinar los grupos que pueden ser por grafos basados en histogramas, kernels, aplicando la regla k-NN empleando los conceptos de punto central borde o ruido. (Pascual, et al., 2007).

Dentro de este tipo de agrupamiento encontramos a DBSCAN (*Density-Based Spatial Clustering of Aplications with Noise*), OPTICS (*Ordering Points to identify the Clustering Structure*), DENCLUE (*Density-Based Clustering*).

Aparte de estas categorías encontramos otras formas de agrupamiento como son: Basados en grid, basados en modelos, información categórica, difusos, redes neuronales, evolutivos, basados en entropía, (Jiménez, 2011) (Hernández, 2006) (Bedregal, 2008).

2.6.3 Medidas de evaluación

Para medir qué tan cerca está un elemento de otro y a su vez qué tan distante están de otros grupos se usa la distancia de similitud o disimilitud (Pascual, et al., 2014). Entendemos como distancia a la función que nos permiten definir la similitud entre dos objetos. Cuanto menor sea la distancia entre dos objetos, más parecidos serán los objetos (Funes, 2008).

Las medidas más conocidas para medir la disimilitud entre objetos son la distancia Euclidiana, la distancia Manhattan, la distancia Chebyshev entre otras (Funes, 2008) (Garre, et al., 2007).

Para saber qué tan válido es un grupo, se tendría que mencionar que un grupo es válido, si es compacto entre sus elementos y aislado de otros grupos (Bedregal, 2008). Para esto se utilizan los índices de validación (Gadania, et al., 2006).

Los índices de validación se pueden clasificar en índices de validación internos y externos (Gutierrez, et al., 2012). Los índices de validación internos son los que miden qué tan cerca están los elementos del grupo y los índices de validación externos son los que miden qué tan separados están de otros grupos.

2.7 Algoritmos Genéticos

Un algoritmo genético es un modelo inspirado en la evolución (Whitley Darrell, 1993), el cual esencialmente realiza un procedimiento de búsqueda y optimización y se modela según los mecanismos genéticos de la selección natural de los seres vivos (Koza, 1992).

Los algoritmos genéticos fueron desarrollados por John Holland junto a su equipo de investigación en 1975 en la universidad de Michigan (Yolis, 2003), (Gestal, 2010), (Whitley Darrell, 1993). Su funcionamiento básico consiste en evolucionar a partir de una población que representa las soluciones candidatas, estos son representados en forma de cromosoma, ver figura 2.7-1 (Cervigón, 2009) para determinar problemas,

intentando producir nuevas generaciones de soluciones mejores que las anteriores, evaluadas por una función de ajuste (Pajares & Santos, 2006).

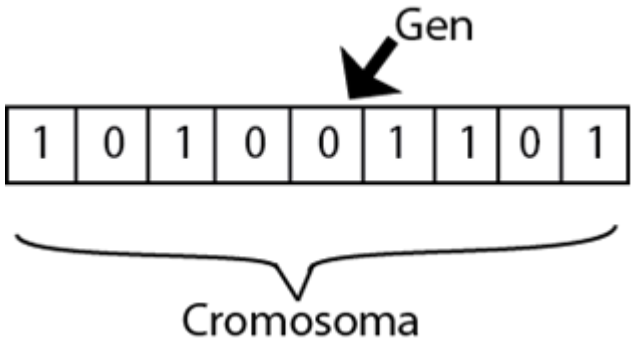


Figura 2.7-1 Representación de un Cromosoma

El gen es un conjunto de parámetros que representan una característica. Es frecuente que el gen del cromosoma utilicé un alfabeto binario 0s y 1s (Kuri & Galaviz, 2007).

Los algoritmos genéticos requieren una serie de parámetros de funcionamiento, como por ejemplo el tamaño de la población, que es un grupo de individuos que pueden interactuar juntos, donde cada individuo puede ser una solución potencial al problema.

Para determinar cuáles de estos individuos corresponden a buenas propuestas de solución, es necesario calificar de alguna manera, su grado de adaptación (fitness). El fitness es un número real no negativo, entre más grande el número de la función de adaptación mejor es el resultado (Díaz, 2007). El objetivo de este número es que permita distinguir buenas propuestas de solución de aquellas que no lo son (Kuri & Galaviz, 2007).

Los algoritmos genéticos ejecutan ciertos operadores para poder generar nuevos individuos, como son: el operador de selección de la población, operador de cruza y el operador de mutación (Cervigón, 2009) (Sanjinez, 2011) (Díaz, 2010).

El operador de selección es el encargado de seccionar a los individuos mejor adaptados en parejas, con la oportunidad de reproducirse, trasmitiendo su información genética a los hijos que procrean (Gestal, 2010), (López, 2010), (Sanjinez, 2011). La

selección ocasiona que haya más individuos buenos, ya que explota el conocimiento que se ha obtenido hasta el momento (Kuri & Galaviz, 2007). Sin embargo, no se debe eliminar por completo las opciones de reproducción a los individuos menos aptos, esto es porque, si hay pocas generaciones se volverá una población homogénea. (Gestal, 2010).

Un algoritmo puede utilizar muchas técnicas diferentes para seleccionar individuos, algunos de los ejemplos más comunes de operadores de selección son: selección por torneo, selección por ruleta, selección elitista, selección jerárquica, entre otros (Gestal, 2010) (Sanjinez, 2011) (Pozas & Vázquez, 2007).

El operador de cruza, es el que, dado dos individuos seleccionados en función de su grado de adaptación, intercambian material genético que se mezclan entre sí para dar lugar a los diferentes hijos los cuales posean un código hibrido, ver figura 2.7-2. Por consiguiente, se obtienen una descendencia que comparte genes de ambos padres, existiendo la posibilidad de que los genes heredados conformen nuevos hijos mejor adaptados al medio que los anteriores. (Kuri & Galaviz, 2007) (Yolis, 2003) (López, 2010) (Sanjinez, 2011) (Pozas & Vázquez, 2007) (Salas, 2010).

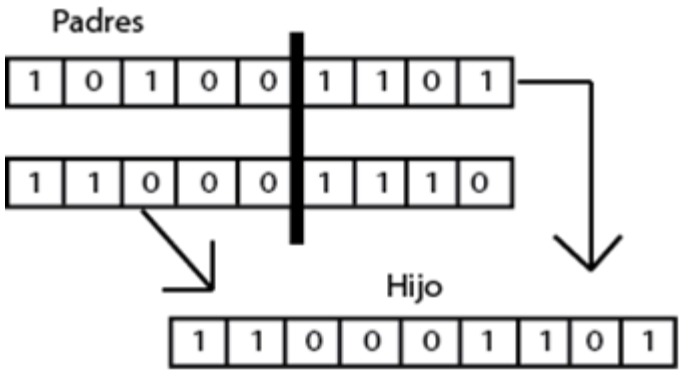


Figura 2.7-2 Ejemplo de Cruza

Para (Gestal, 2010) y (Yolis, 2003) algunos operadores de cruza son: cruza por 1 punto, cruza por 2 puntos, cruza uniforme, cruza especifica de codificaciones no binarias.

El operador de mutación es una alteración a propósito (Salas, 2010), lo que hace que se modifiquen ciertos genes de un cromosoma ver figura 2.7-3, siendo el responsable del aumento o reducción del espacio de búsqueda dentro del algoritmo genético (Arranz & Parra, 2007). El operador de mutación es quizás una solución cuando el algoritmo genético se encuentra estancado en la búsqueda de soluciones. La probabilidad de mutación que se utiliza comúnmente como parámetro, es muy baja generalmente es del 1% (Sanjinez, 2011) y las formas en que puede realizar la mutación es de forma fija o de forma variable (López, 2010).

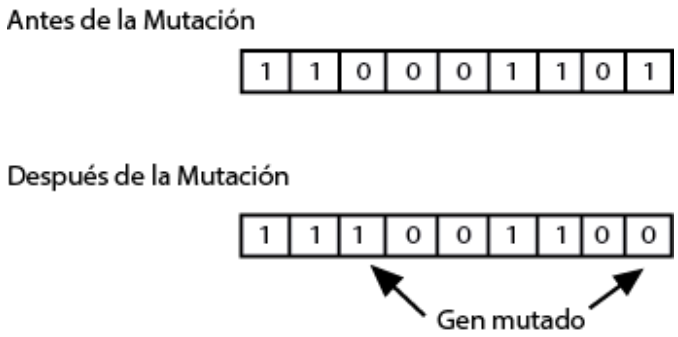


Figura 2.7-3 Ejemplo de Mutación de un punto

Para (López, 2010) algunos tipos de mutación son la mutación al azar, mutación gaussiana entre otras.

2.8 Resumen

En este capítulo se vio la definición de reconocimiento de patrones, las formas de clasificación que existen para etiquetar objetos, así como las técnicas existen para solucionar el problema del reconocimiento de patrones. También se vieron las principales formas de aprendizaje (la clasificación y el agrupamiento). No obstante, podemos ver que hay otros aprendizajes dependiendo cual es el objetivo.

Por otra parte, existe un aprendizaje que intenta utilizar las dos formas más usadas de aprendizaje como es el semi-supervisado.

También se vio que para ambos casos existen formas de validar lo que se ha aprendido tanto en la clasificación como en el agrupamiento. En el caso de la clasificación, la validación cruzada es la encargada de ver qué tan bien ha aprendido el clasificador.

En el caso de agrupamiento para validar su aprendizaje se utilizan los índices de validación tanto internos como externos. Primero se miden las distancias de los objetos del mismo grupo y después la distancia que hay entre otros grupos.

Por último, se vio en este capítulo el funcionamiento de los algoritmos genéticos, los cuales son utilizados para buscar las mejores soluciones a problemas de optimización. Para utilizar un algoritmo genético es necesario utilizar cromosomas para representar a los individuos. La forma de evaluar a los individuos es mediante la función de aptitud la cual nos indica qué tan bueno o malo es un individuo.

Para continuar el proceso del algoritmo genético es necesario utilizar ciertos operadores los cuales se encargan de generar nuevas poblaciones (soluciones) más aptas.

El primer operador es el de selección, éste selecciona a dos padres de acuerdo a la mejor función de aptitud para que estos se reproduzcan y tengan descendencia. Seguido de este operador viene el operador de cruza, donde se mezclan los genes de los padres.

Por último, se presentó el operador de mutación, aunque es un operador que se utiliza en un porcentaje muy pequeño, puede tener la capacidad de mejorar o empeorar los resultados. El operador de mutación es muy importante porque explora nuevas soluciones permitiendo que el algoritmo genético no se estanque y pueda seguir evolucionando.

CAPÍTULO III ESTADO DEL ARTE

3.1 Introducción

En la actualidad hay un sinfín de investigaciones que están buscando encontrar la mejor forma de etiquetar objetos ya sea para clasificar o para agrupar, pero estas investigaciones tienen en común la utilización de los algoritmos genéticos para llegar a mejorar la forma de etiquetarlos, es decir encontrar la mejor o la más óptima clasificación o agrupamiento con la ayuda de los algoritmos genéticos.

En este capítulo se muestran algunas investigaciones que están orientadas a etiquetar los objetos y que pretende encontrar una de las mejores formas de etiquetar estos objetos.

3.2 Trabajos relacionados

Cabe señalar que en la búsqueda del estado del arte no se encontró un trabajo que se parezca al presente proyecto de tesis. Ya que intentamos juntar dos formas de aprendizaje que son totalmente diferentes, pero si lo vemos de manera objetiva, podemos ver que el agrupamiento es un paso previo de la clasificación en algunas ocasiones.

Lo más cercano que pudimos ver es el aprendizaje semi-supervisado o también conocido como aprendizaje híbrido, en el cual se utiliza una gran muestra de objetos sin etiquetar y una pequeña muestra de objetos clasificados y a partir de esta se genera el conjunto de entrenamiento para encontrar las etiquetas para los objetos no etiquetados (Pascual, et al., 2007).

El primer trabajo que veremos es el de (Bokan, *et al*., 2011), el cual intentan resolver el cálculo del número de grupos a partir de un conjunto de objetos.

Los autores emplean el método IEKA (*Intelligent Evolutionary K-means Algorithm*), que es un método de agrupamiento no-supervisado que se basa en la unión de una solución óptima que pueden generar los Algoritmos Genéticos.

Este método genera un cromosoma que posee k centros de k grupos, que representan a los genes de dichos cromosomas, los cuales serán recalculados mediante técnicas de agrupamiento *k-means* (Kumar, et al., 2008).

Para la generación de la población del algoritmo genético, cada cromosoma contendrá k centroides aleatorios, Una vez calculados los grupos por cada cromosoma mediante *k-means*, se evalúa el grado de aceptación del grupo mediante los índices de validación; después de esto se sigue con las operaciones elementales que forman la estructura de un algoritmo genético.

Los individuos de la población tienen el mismo tamaño y son representados por dos cadenas: la cadena genotipo y la cadena mascara. En la primera se encuentran los centros del grupo y en la segunda es una cadena compuesta de 1s y 0s que representan si son genes activos o no.

La función de aptitud que utiliza para el algoritmo genético es a través de índices de validación, la cual determina el número óptimo de grupos en un conjunto de datos y para ello se emplean dos criterios de validación, el Índice de Dunn (Davies & Bouldin, 1979) y el Índice de Davis-Bouldin (Dunn, 1974).

Los operadores del algoritmo genético utilizados son: el operador de selección multi-elitismo, el operador de cruza FFD (Shinn-Ying Ho, 2009) y el operador de mutación adaptativa.

Las bases de datos que utiliza el autor para los experimentos son: la base de datos IRIS y una base de datos Sintética. La muestra de entrenamiento IRIS, consta de 150 instancias de plantas, donde cada una posee 4 atributos: largo del sépalo, ancho del sépalo, largo del pétalo y ancho del pétalo. IRIS consta de 3 clases cada una con 50

instancias, tales clases: son Iris setosa, iris versicolor e iris virginica. La muestra de entrenamiento sintética está conformada de 300 instancias, donde cada una posee 2 atributos; consta de 3 clases, de 100 instancias cada una. La base de datos sintética es creada automáticamente y consta de 300 elementos de 3 clases de 100 elementos y estos elementos constan de atributos.

El autor señala que para la base de datos IRIS, el algoritmo IEKA determió 2 grupos: uno de 100 instancias y el otro de 50 instancias. Para la Base de datos sintética, el mismo método determinó 3 grupos, cada uno con 100 instancias. Para obtener estos resultados se ha ejecutado 200 veces los algoritmos. Entonces el método IEKA es más eficiente, rápido y realiza un agrupamiento compacto donde un grupo están muy separado de otros grupos.

Otro trabajo que emplea algoritmos genéticos con agrupamiento es el de Moreno (Moreno, *et al*., 2010), en el cual busca obtener un agrupamiento de igual tamaño tomando en cuenta sus atributos. En este caso el autor propone usar un método basado en algoritmos genéticos. Para eso toma en cuenta dos aspectos, el primero es que sean iguales los grupos, y el segundo aspecto es la explosión combinatoria que va de la mano con el número de elementos totales que se tengan y la cantidad de grupos que quieren formarse. Un individuo representa una colección determinada de grupos (G).

La población que va a generar n individuos de manera aleatoria, cumpliendo con las restricciones que cada elemento debe estar en una sola de las posiciones de la población.

El gen del cromosoma contiene el identificador de un elemento, y su posición dentro de la matriz define el grupo al que pertenece.

Como función de aptitud lo que primero se realiza es obtener grupos homogéneos del total de elementos. Primero calcula el promedio de cada atributo de la totalidad de los elementos, luego para cada grupo de cada individuo se calcula el promedio de cada

atributo. Posteriormente se calcula la sumatoria entre la diferencia al cuadrado respecto al número de atributos entre cada grupo del individuo y el total de los elementos.

Los operadores que utiliza el autor para el algoritmo genético son la selección por ruleta. El operador de cruza en un punto de cruce aleatorio combina el primer segmento del primer padre con el segundo segmento, pero en el orden en que los genes correspondientes aparezcan en el segundo padre y viceversa. La modificación que sufre esta cruza es que se corta a lo largo de la columna. Para el operador de mutación es por intercambio de un gen a otro.

Los resultados que se obtienen es que a partir del método propuesto es posible obtener homogeneidad de grupos. Es decir, se logra hacer grupos bastante homogéneos para múltiples atributos, incluso cuando el número de combinaciones posibles es muy elevado y con muy poco tiempo de cómputo.

También nos menciona que este método puede ser usado en la conformación de grupos de trabajos en los cuales se busca que estos grupos de trabajo estén bien repartidos considerando varias habilidades.

Otro estudio que utiliza un algoritmo genético para agrupamiento es el de (Sabau, 2012). El problema que trata de resolver es que, en los algoritmos de agrupamiento, como el agrupamiento jerárquico y el agrupamiento de partición, siempre se requiere especificar los parámetros del algoritmo. Es decir, el usuario tiene que proporcionar los parámetros. Para especificar tales parámetros se tiene que tener un profundo conocimiento o haber trabajado con los datos. Algunos de estos son parámetros de inicialización de grupos, tamaño de los vecinos en términos de distancia, el número mínimo de objetos por grupos, etc.

El autor propone el uso de algoritmos genéticos los cuales proporcionan soluciones a problemas del agrupamiento a través de diferentes distribuciones de conjuntos de datos con una cantidad mínima de parámetros que son definidos por el usuario.

El propósito de (Sabau, 2012) es eliminar el cálculo de parámetros dados por el usuario, siendo un método original para la solución del agrupamiento. Para esto es necesario tener en cada genotipo una solución al problema del agrupamiento. El gen es definido por varios atributos basados en densidad y son muy importantes para recuperar la codificación de la partición.

Cada agrupamiento basado en densidad tiene genes definidos que sólo puede atraer objetos que aún no han sido atraídos por otros grupos, el esquema de codificación propuesto permite obtener resultados de cruza siempre válidos, con grandes variaciones, incluso cuando se utilizan operadores de cruce simples. La codificación que se utiliza, solamente codifica los grupos prototipos como genes del genotipo, donde al contrario de tener cada objeto en un gen, unos conjuntos de datos son representados por un gen. El genotipo es de tamaño variable permitiendo tener un número variable de grupos codificados.

La solución para el agrupamiento comienza al generarse la generación cero, formando grupos uno por uno en secuencias ordenadas. Al azar se seleccionan puntos de partida disponibles en el conjunto de datos, la distancia del sector y número mínimo de puntos por sector, solo se establece al azar al siguiente grupo en un espacio restringido calculando la distribución del conjunto de datos.

La distancia del sector se obtiene tomando los valores que van desde la distancia mínima de la matriz a la distancia máxima de la matriz. De una manera similar, el intervalo para el número mínimo de puntos por sector es construido. Tener un gen con un punto de partida, con la distancia del sector y con número mínimo de puntos por sector, se puede empezar a construir recursivamente el grupo con el primer punto, tomándolo como punto de inicio. En cada interacción, basado en la distancia del sector cada punto que es recién añadido correspondiente a puntos vecinos los cuales son identificados y añadidos al grupo, si la identificación de puntos vecinos excede el número mínimo de puntos por sector, la expansión recursiva del grupo se aplica de

nuevo para cada punto recién encontrado de lo contrario la expansión recursiva del grupo es detenida.

Un grupo solamente puede crecer recursivamente en puntos que no han sido tomados por otros grupos previamente. Debido a esto, el orden en que cada grupo están siendo definidos es muy importante, esto influye dramáticamente en la forma en que trabaja el operador de cruza.

Para medir la aptitud de cada individuo utilizan los índices de validación. Para el índice de validación interno, utiliza el índice de Silhouette y el índice de validación externo utiliza el índice de Rand. El índice de Silhouette no requiere ningún conocimiento previo sobre una solución correcta de agrupamiento, el índice de Rand siendo un índice de validación externo, solo puede evaluar soluciones de grupo basados en una solución correcta de agrupamiento. En este contexto el índice de Rand solo ha sido usado en orden para tener una comparación basada con otros índices de validación.

Como operador de selección (Sabau, 2012) utiliza la selección proporcional en la cual se obtiene la probabilidad de ser seleccionados directamente de acuerdo a su función de aptitud y es dividido con la suma total de toda la generación. Esto solo para el primer individuo seleccionado, para el segundo individuo se obtiene aleatoriamente, obviamente no puede reproducirse con el mismo.

Como operador de cruza, utiliza la cruza uniforme en la cual tiene una parte fija de 0.5 como máximo permitiendo para que los cromosomas de los padres puedan contribuir a nivel de genes en lugar de nivel de segmento.

Como operador de mutación, utiliza la mutación gaussiana en la cual solo altera un solo gen de los parámetros de densidad relacionada, añadiendo valores aleatorios siguiendo distribuciones gaussianas correspondientes. La posición de cada gen a nivel de cromosomas está siendo alterado por selección aleatoria de dos genes y cambia su posición.

Los experimentos que se realizaron en este trabajo fueron con un conjunto de datos sintéticos y un conjunto de datos reales con el fin de evaluar el rendimiento del algoritmo propuesto.

Tanto en el conjunto de datos sintético como en el conjunto de datos reales se cuenta con una dimensión extra en la que se encuentran las etiquetas correctas de los grupos, dimensión que sólo se utiliza para calcular el índice de Rand.

En conclusión, se observa que los experimentos que se realizaron a través de todo el análisis del conjunto de datos, ofrece un rendimiento similar al algoritmo *DBSCAN* pero, aun que es similar, es más fácil de usar y no requiere ningún conocimiento previo sobre el objetivo del conjunto de datos. Sin embargo, como era de esperarse en casos como el K-means es inferior en los resultados en el método que propone el autor.

3.3 Resumen

Como resumen se puede decir que los principales problemas de agrupamiento son: cómo encontrar los mejores centroides para un conjunto de objetos, cómo encontrar grupos homogéneos de acuerdo a sus características y cómo hacer un algoritmo genético basado en densidad que no requiera parámetros para generar los grupos y que el usuario los introduzca.

Vemos claramente que resolver este tipo de problemas con algoritmos genéticos realmente es una buena decisión, ya que al estar buscando las mejores soluciones y automatizar este proceso.

La codificación de los individuos se tiene que considerar el tamaño y el tipo de datos que se manejan. También se observó que la función de aptitud en los tres trabajos presentados es fundamental para alcanzar el objetivo y obtener buenas soluciones.

También es rescatable mencionar que las pruebas se realizan son con dos muestras de entrenamiento una sintética y una real. Con la base de datos sintética se controla el

resultado y es más sencillo ver si puede o no funcionar lo propuesto, y con la base de datos real se ve cómo se comporta el algoritmo genético.

Sin embargo, en los tres trabajos que se vieron, es claro que hay que adaptar los operadores del algoritmo genético, ya sea de selección, cruza o mutación; para obtener buenos individuos que cumplan con el objetivo de cada trabajo. Los resultados que arrojan los trabajos son buenos y llegan a cumplir con su cometido.

Sin lugar a duda, estos trabajos amplían el panorama de cómo resolver el problema que se quiere resolver y muestran que es posible adaptar operadores para resolver los problemas de agrupamiento.

CAPÍTULO IV PROPUESTA DE SOLUCIÓN

4.1 Introducción

Primero se recordará acerca de cuál es el problema de esta tesis, que en este caso es: cómo realizar aprendizaje no supervisado utilizando el mismo criterio de aprendizaje supervisado para encontrar grupos homogéneos (las clases, para formar una muestra de entrenamiento) dado un algoritmo de aprendizaje no supervisado. Retomando los trabajos del capítulo III, podemos observar que en este trabajo se quiere resolver dos sub-problemas del agrupamiento y que son tener el mejor agrupamiento pero que los grupos sean homogéneos.

En este capítulo veremos los métodos que se utilizarán para resolver el problema planteado y los métodos para lograr llevar acabo el objetivo de esta tesis, así como; los operadores que se utilizarán para el algoritmo genético y cómo se modificaron estos para buscar grupos homogéneos.

4.2 Desarrollo de la propuesta de solución

Primero hay que mencionar que el trabajo se realizó en una metodología basada en módulos, lo cual nos permite trabajar de forma más eficiente. Algunos módulos del algoritmo genético ya estaban hechos, por lo que solo se realizaron algunos módulos o en su caso se modificaron para el problema propuesto.

Los operadores que se realizaron o se adaptaron fueron: el operador de selección, el operador de cruza, el operador de mutación y el módulo de la función de aptitud. El diagrama del algoritmo genético se puede ver figura 4.2-1.

4.2.1 Creación de la población inicial

Lo primero que realiza el algoritmo genético, es generar la primera población de forma aleatoria, (como se vio en el capítulo II), la forma de codificar los individuos es de forma binaria 1s o 0s donde cada individuo será una posible solución a nuestro problema.

En el caso de nuestra propuesta el individuo será codificado mediante un arreglo de números, los cuales representarán el grupo al que pertenecen los n grupos posibles para cada elemento del arreglo.

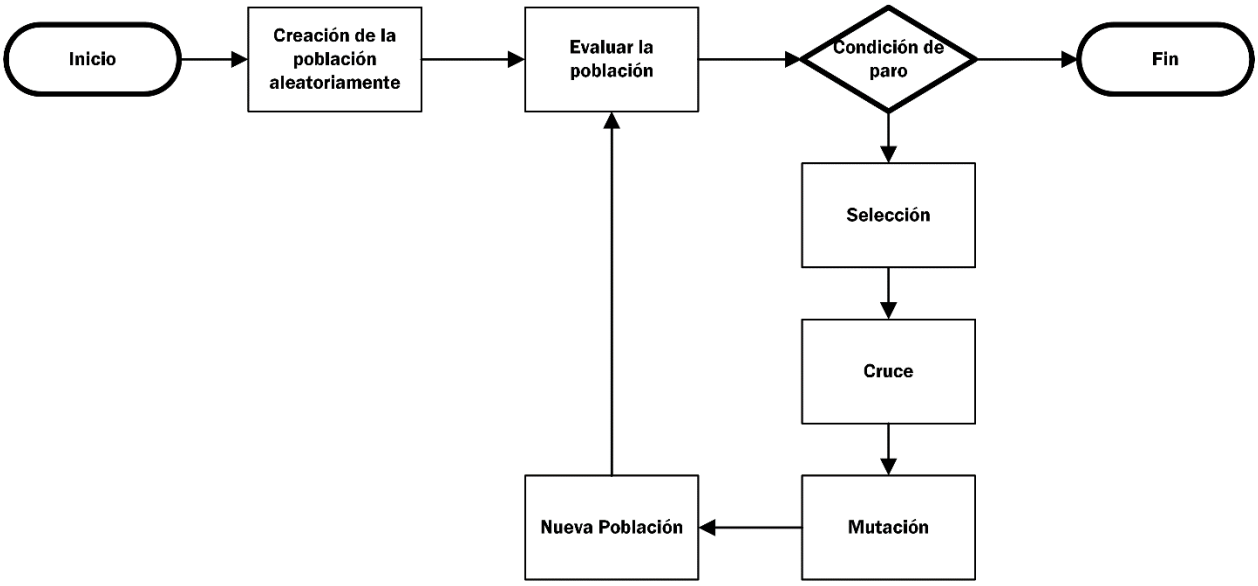


Figura 4.2-1 Algoritmo Genético

La longitud del arreglo corresponde al número de objetos que se tiene en nuestra muestra de entrenamiento.

Una forma de ejemplificarlo es que, si tenemos una muestra de 10 objetos en nuestra muestra de entrenamiento, el individuo del algoritmo genético será constituido de 10 genes, es decir, cada objeto de la muestra de entrenamiento es un gen del individuo ver figura 4.2-2.

4.2.2 Evaluación de la población

Una vez que tenemos la codificación de los individuos, el siguiente paso en el algoritmo genético es evaluar a nuestros individuos generados, ya que una parte importante de

los algoritmos genéticos es justamente la función de aptitud. La función de aptitud que utiliza el k-vecinos más cercano, el cual es evaluado con validación cruzada con n=10 interacciones. En la función de la tabla 4.2-1 se muestra como se define la función de aptitud.

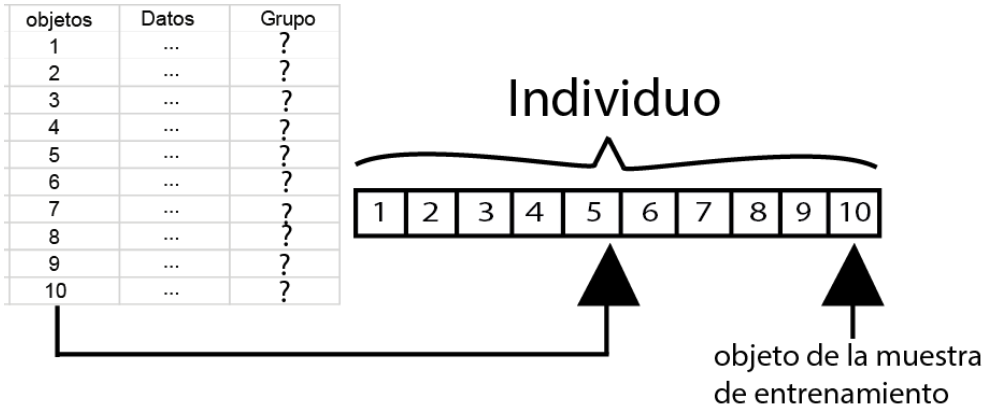


Figura 4.2-2 Representación del Individuo

Tabla 4.2-1 Función de Aptitud

$FA = \left(KNN(NfoldCV)\right)$	Donde: • KNN = K-vecinos más cercanos. • NfoldVC = Validación cruzada de 10 interacciones.

El algoritmo basado en instancias K-NN (K-vecinos más cercanos), es el algoritmo de clasificación utilizando para la propuesta. El algoritmo K-NN recibe el conjunto de prueba y el conjunto de entrenamiento, para todos los datos de prueba se calcula la similitud entre los elementos del conjunto de entrenamiento, se obtienen los k vecinos más cercanos al elemento y se clasifica con la clase a la que pertenece. La función de similitud que se utiliza es para datos de tipo numérico y se representa por la función de la tabla 4.2-2. Al finalizar se obtiene el porcentaje de las veces que los k vecinos clasificó correctamente y este valor será el porcentaje de la clasificación.

Tabla 4.2-2 Función de similitud para datos numéricos

$Similitud(x,y) = \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{\lvert y_i - x_i \rvert}{Maximo(x_i, y_i)}$	Donde: • x, y= Representan los elementos entre los cuales se está calculando la similitud • i=atributo que se está comparando

El funcionamiento de la validación cruzada es el siguiente: los datos de la muestra de objetos son divididos en *n* subconjuntos, donde uno de los subconjuntos se utiliza como prueba y el otro como entrenamiento. Esto se repite hasta que se termine con la muestra, para este caso en particular se utiliza el valor de *n* = 10 para las interacciones. Al finalizar se obtiene un promedio de clasificación. Este promedio será la función de aptitud del individuo, es decir, con este número será posible saber cuáles serán los mejores individuos para la operación de selección.

4.2.3 Condiciones de Paro

A continuación, se ve si se detiene la ejecución del algoritmo con una de las dos opciones, la primera es si se ha llegado a las generaciones deseadas y la segunda es si se ha obtenido la función de aptitud más alta. En caso de ser negativas las dos formas de paro del algoritmo genético se continúa con los operadores del algoritmo genético, en este caso se continúa con la selección.

4.2.4 Selección

Para el operador de selección se utilizan dos selecciones. La primera selección que se utilizará es la selección por ruleta y la segunda es la selección por torneo.

Lo primero que realiza la selección por ruleta, es asignar a los individuos de la población una parte proporcional de una ruleta de acuerdo a su función de aptitud, de tal forma que la suma de todos los porcentajes sea la unidad. Los individuos con mayor

función de aptitud recibirán una porción de la ruleta mayor, que la recibida por los individuos con menor función de aptitud ver figura 4.2-3.

Una vez definida la porción de importancia para cada individuo, se selecciona un individuo generando un número aleatorio con intervalo entre {0…1} y el proceso se repite para obtener otro individuo, de manera que los dos individuos se reproducirán.

La otra selección que se propone utilizar será la selección por torneo, en la cual son seleccionados de forma aleatoria un número *k* de individuos de la población y el que tenga mayor FA de los *k* individuos, es el que se selecciona.

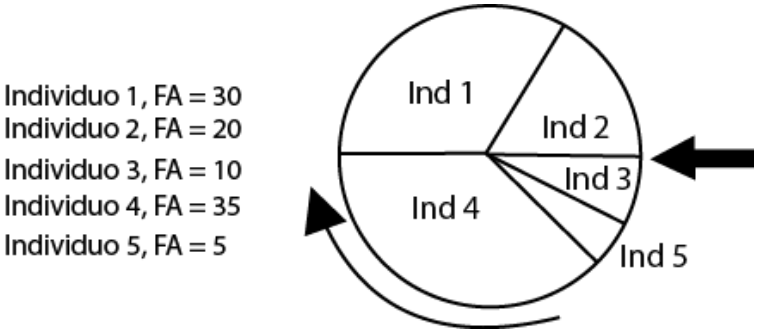


Figura 4.2-3 Ejemplo de Selección por Ruleta

Para la selección de *k* individuos, hay que tener en cuenta que la precisión de selección es un poco mayor, cuando se cuenta con más individuos en el torneo. De esta manera, los peores individuos apenas tienen oportunidad de reproducirse, que cuando el tamaño de *k* es pequeño.

4.2.5 Cruza

Teniendo los dos padres se pasa al operador de cruza. En este caso fue necesario crear una nueva cruza, llamada **"cruza por intersecciones"**. Lo que primero realiza esta nueva cruza es verificar que los individuos sean válidos, es decir, que sean homogéneos los grupos. Esto depende del número máximo y un mínimo de elementos para cada grupo del individuo. La fórmula para obtener el máximo y el mínimo se muestran a continuación en la figura 4.2-7.

El paso siguiente que realiza la cruza es obtener intersecciones de los grupos del individuo X con el individuo Y. El proceso para esto es el siguiente: el individuo X se mantiene igual, pero se obtienen los índices de cada grupo. El individuo Y, llenará una matriz con el índice de la posición en que encuentra el grupo, la dimensión de la matriz será del número de grupos por el máximo de elementos por grupos y los demás de la matriz se rellenan de -1s ver figura 4.2-4.

Tabla 4.2-3 Ejemplo de operación de cruza

$Número\ Mínimo\ (NMo) = \frac{TI}{G}/G$	Ejemplo: • TI = 100 • G = 4 100/4 = 24/4= 6 mínimo de objetos por grupo
$Número\ Máximo = \left((G^2) - G\right) * NMo$ Donde: • G = Grupos • TI = Tamaño del individuo	4^2 = 16-4=12*6 =72 máximo de objetos por grupo

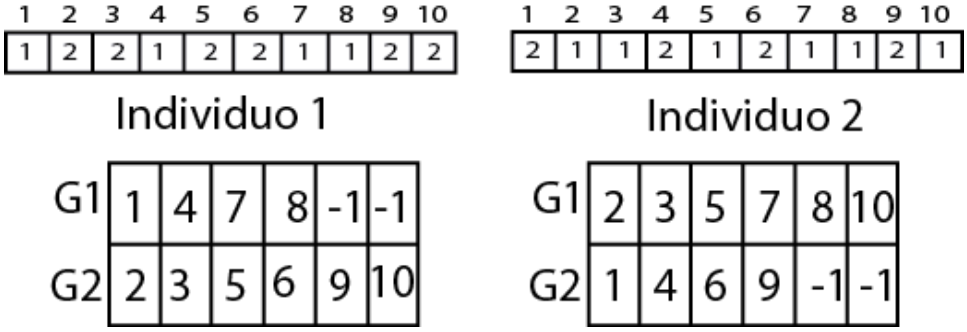


Figura 4.2-4 Obtención de Índices

Con esto se obtiene una matriz, la cual ayuda a reasignar los grupos del padre 2 ver Figura 4.2-5.

Después de obtener la matriz de intersecciones, en esta se busca el número mayor de la matriz ver figura 4.2-6. Para reasignar los grupos se toma el grupo de la columna y se reasigna el renglón; así se re-etiqueta los grupos, se elimina la columna y el renglón y se repite el proceso, ver figura 4.2-7.

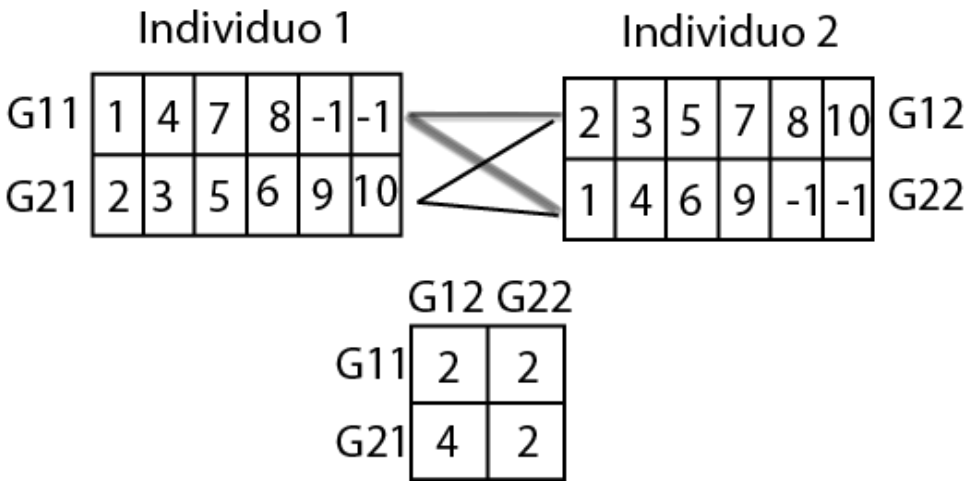


Figura 4.2-5 Obtención de matriz de intersecciones

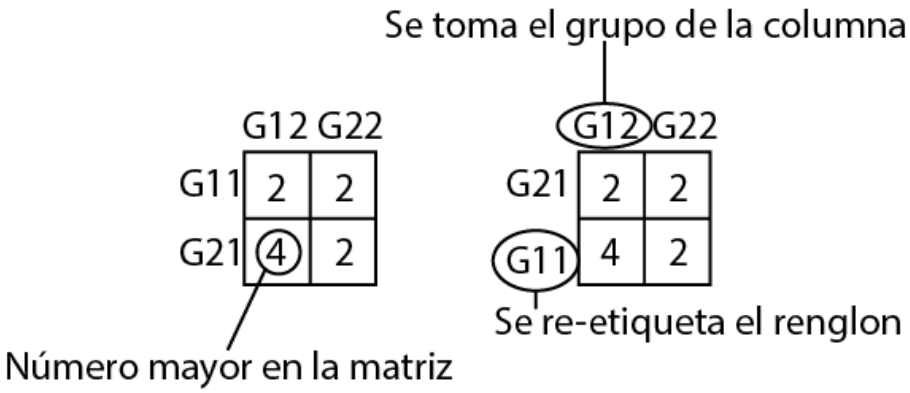


Figura 4.2-6 Buscar en número mayor

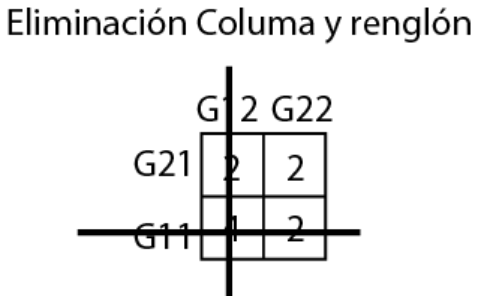


Figura 4.2-7 Eliminación de la columna y renglón

Con el padre dos y sus grupos reasignados realizar la cruza en donde se comparan los genes de los padres (1,2). En caso de que los genes de una posición de ambos padres sean iguales, pasa el gen de esa posición igual al hijo en caso contrario se genera un número aleatorio que tiene un rango de 1 a la suma de las funciones de aptitud de los padres, ver figura 4.2-8. Generado este número, se verifica si es menor a la función de aptitud del individuo 1 y se le asigna el gen del padre; en caso de ser mayor se le asigna el gen del individuo 2. Al terminar el proceso para todos los genes se realiza nuevamente la parte de validar que nuestro hijo sea válido de no ser así, se regresa a realizar de nuevo la cruza.

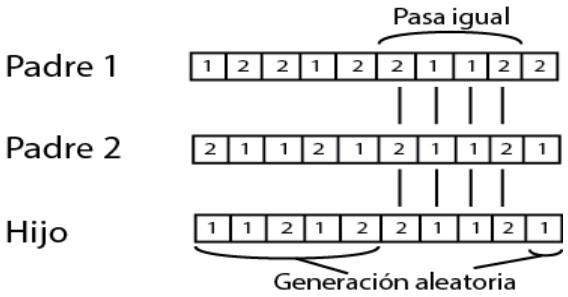


Figura 4.2-8 Cruza por Intersección

4.2.6 Mutación

Al terminar el método de cruza pasaremos al método de mutación del cual se hace de forma aleatoria en las generaciones y en un porcentaje muy bajo de la población, ciertamente lo que hace la mutación es cambiar los genes del hijo por otros.

En este caso se tomó una mutación por intercambio donde se generan dos números aleatorios, del total de genes del individuo. Los dos números aleatorios corresponden a las posiciones donde se va a intercambiar en el hijo, ver figura 4.2-9.

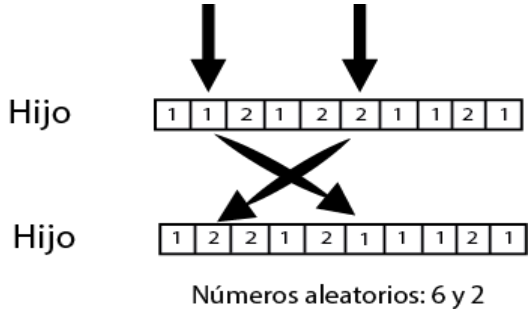


Figura 4.2-9 Mutación por Intercambio

4.3 Resumen

En este capítulo se vio el método de solución para nuestro problema. Cabe señalar que los diagramas de las diferentes partes del método propuesto que se implementó se tienen desde el anexo A1 al A12. Lo primero que realizaremos es la generación de los individuos, el cual se genera del tamaño de los elementos que se tengan en la muestra de entrenamiento. A partir de aquí se evaluará a los individuos que se genera a través del algoritmo *k-nn* y a su vez este es evaluado con *10-cross validation*. Esta evaluación representa qué tan bien ha aprendido el clasificador y a su vez nos indica el valor de la función de aptitud del individuo.

Después de haber evaluado a los individuos se ve sí se han cumplido los parámetros de paro del algoritmo. De lo contrario, el proceso continúa con la selección de individuos de acuerdo a lo antes establecido.

Habiendo seleccionado a los individuos(Padres) se ha realizado una nueva forma para cruzar a los individuos seleccionados, la cual obtienen los números máximos y mínimos por grupo, como consecuencia a esto se obtiene las intersecciones de los padres y se re-etiqueta al segundo padre para realizar la cruza. En la cual se pasan los mismos genes al hijo si los padres tienen los mismos genes, de lo contrario se genera un

número aleatorio el cual decidirá si se toma el gen del padre uno o del padre dos, siendo el padre con mayor función de aptitud el que puede pasar más genes.

El método de la mutación es una de las más simples ya que se cambian solo dos genes del hijo, este nuevo hijo se incorpora a la población y se repite el proceso hasta que se cumplan las condiciones de paro.

CAPÍTULO V EXPERIMENTACIÓN

5.1 Introducción

En este capítulo se describen los experimentos que se llevaron a cabo para demostrar que el método que se propone funciona correctamente. También se describen las muestras de entrenamiento que se utilizan para los experimentos, así como la constitución de las muestras de entrenamiento.

Como primera parte se verá cómo funciona el método propuesto con una base de datos propuesta o sintética, con el objetivo de comprobar el funcionamiento de la propuesta. Después se verá cómo se comporta el método con una base de datos real, si ésta alcanzado una buena clasificación, si obtiene los grupos naturales de la base de datos.

A partir de la utilización de estos experimentos y de acuerdo a sus resultados, se comenzará a buscar los parámetros que ayuden al algoritmo genético a que encuentre una de las mejores soluciones. Los parámetros que se van a variar son el operador de selección que puede ser (ruleta y torneo), el número de vecinos, el porcentaje de selección elite (que ayuda a mejorar la clasificación), el número de generaciones, el tamaño de población y el porcentaje de mutación.

Por último, se ve si los mejores parámetros obtenidos en todo el proceso de experimentación, podemos igualar o mejorar la clasificación utilizando los mejores parámetros.

Para comparar los resultados que se obtuvieron con el método propuesto se comparó con la muestra de entrenamiento IRIS utilizada para clasificación y con el método de clasificación de K-vecinos más cercanos.

5.2 Bases de datos utilizadas en los experimentos

Las muestras de entrenamiento que se han utilizado para este trabajo son dos: una muestra de entrenamiento sintética y una muestra de entrenamiento clásica. La muestra de entrenamiento sintético, llamado así porque tiene elementos controlados, permite saber el resultado que nos darán los experimentos. Con el objetivo de saber que tan buenos o malos son los resultados con el método que se propone.

La muestra de entrenamiento sintética está compuesta por 20 objetos, correspondientes a 2 tipos de frutas. Cada clase contiene 4 atributos que son alto, ancho, peso y diámetro, a simple vista son muy diferentes una clase de otra, el primer grupo consta de 10 manzanas y el segundo grupo de 10 melones **(Ver Anexo B2)**.

La segunda muestra de entrenamiento que se utiliza es una base de datos clásica “Iris”, la cual se emplea en trabajos de agrupamiento y clasificación, la cual es muy utilizada en el estado del arte. Consta de 150 elementos de plantas y 3 clases, donde cada una posee 4 atributos: largo de sépalo, ancho de sépalo, largo del pétalo y ancho del pétalo (Marshall, 2007)

5.3 Primer experimento

Para comenzar con los experimentos, primero se utiliza la muestra de entrenamiento sintética. El primer experimento en este caso es el más importante y crítico, ya que es el que comprueba si lo que se está proponiendo es correcto.

La base de datos sintética contiene 20 elementos con dos clases, Melón y Manzana. El algoritmo genético debería de encontrar los grupos naturales de esta base de datos, puesto que se trata de 2 tipos de objetos claramente diferentes.

5.3.1 Objetivo del primer experimento

El objetivo principal de este experimento, es encontrar los grupos naturales de la base de datos, así como ver en cuántas generaciones encuentran los grupos naturales el algoritmo genético.

Como se mencionó en la introducción, entraremos a un proceso interactivo de probar los parámetros para el algoritmo genético que mejor resultados nos den, a la hora de obtener los grupos de las bases de datos. Primero veremos cuál es el valor para k, del k vecinos más cercanos (KNN), que es mejor y nos entrega buenos resultados, los valores que se probaron para este parámetro son cuando k tiene un valor de 3, 5, 7.

Otro parámetro que se varía es la probabilidad de mutación, para ver si esto ayuda a mejorar la clasificación tomando en cuenta el mejor valor de k del KNN.

Para este primer experimento, los parámetros que se utilizan son: población de 100 individuos, máximo de 100 generaciones y probabilidad de mutación de 5%, ver Tabla 5.3-1.

Tabla 5.3-1 Primer Experimento

K-NN	Elite	Generaciones	Número de Individuos	Mutación (%)
3,5,7	2	100	100	5

5.3.2 Diferentes valores de k

En la figura 5.3-1 se puede observar que el método propuesto evoluciona conforme avanzan las generaciones alcanzando un 95% de función de aptitud con diferentes

valores de K. También se puede observar en la figura 5.3-1 que mientras que el valor de *k* sea mayor, el algoritmo evoluciona más rápido.

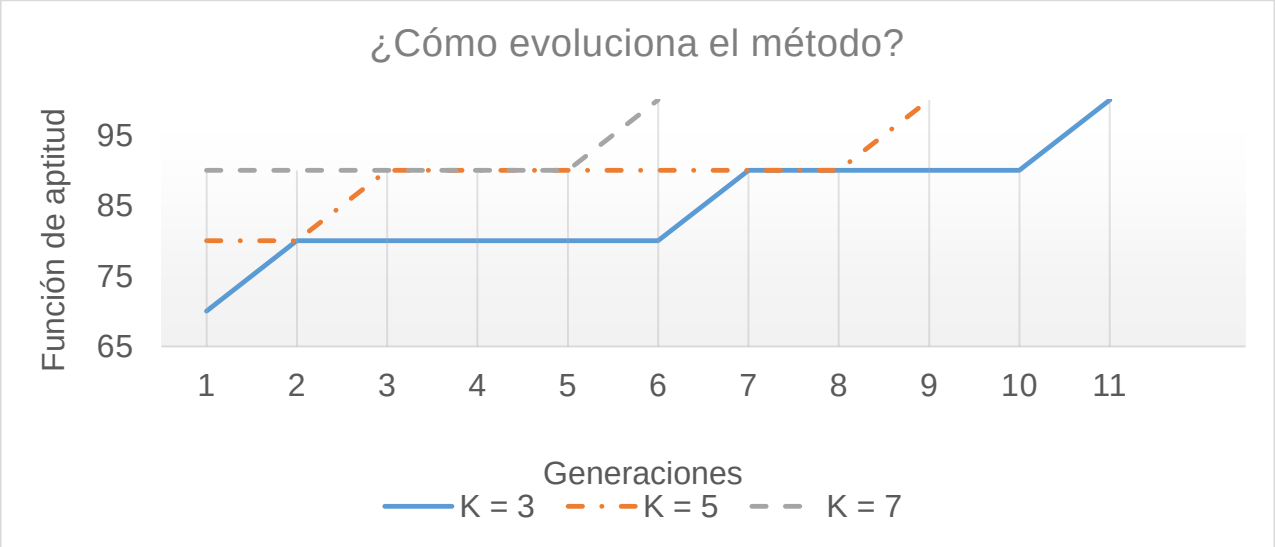


Figura 5.3-1 Gráfica de la función de aptitud con diferentes valores de k

5.3.3 Diferente mutación

Utilizando el mejor resultado del experimento anterior se emplearán 7 vecinos. Ahora se modifica el valor de la mutación con valores de 5, 10, 15 y 25. El resultado de su evolución se puede ver en la figura 5.3-2.

Al ver la gráfica de la Figura 5.3-2 se ve claramente que, a una mayor mutación, se obtiene en pocas generaciones, los grupos naturales.

5.3.4 Conclusiones del primer experimento

Como se pudo observar, en los primeros experimentos, con la muestra de entrenamiento sintética podemos notar que el método propuesto está funcionando. También se ve que al variar algunos de los parámetros del genético, está mejorando su evolución. Además, se pudo apreciar que a mayor número de *k* y de mutación, el

algoritmo encuentra más rápido los grupos, ya que estos parámetros están ayudando al algoritmo genético.

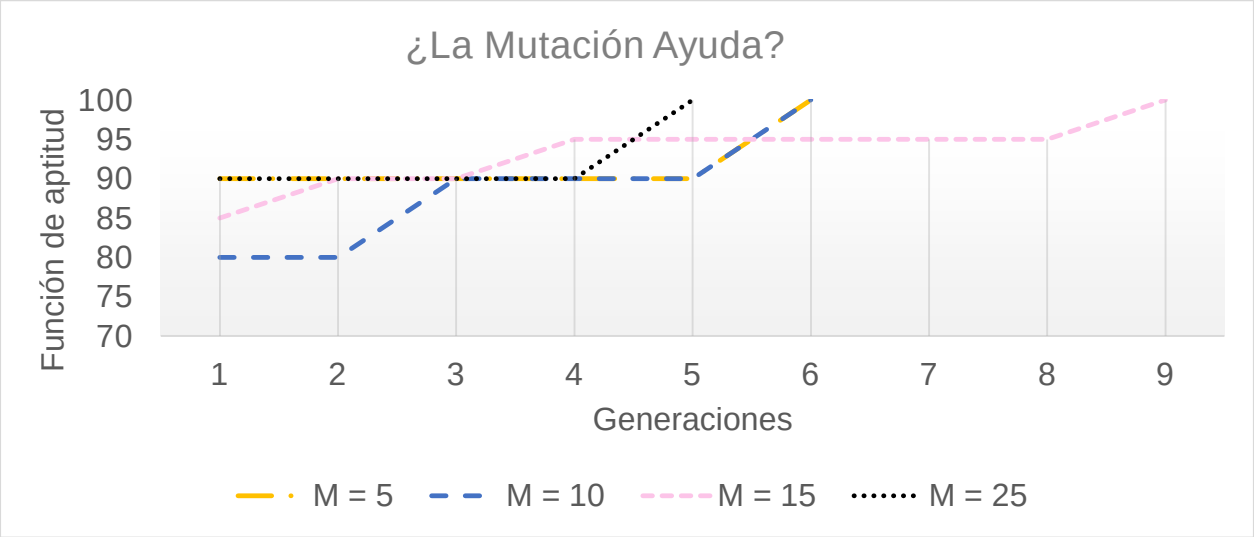


Figura 5.3-2 Gráfica de diferentes mutaciones

5.4 Segundo experimento

Como pudimos observar en el primer experimento el método que se propone si funciona y encuentra los grupos naturales de la base de datos sintética.

En este segundo experimento vamos a ver cómo se comporta con la base de datos clásica IRIS. La muestra de entrenamiento Iris, que como se mencionó en la introducción de este capítulo, es una base de datos de 150 elementos de plantas que cuenta con 3 clases y 4 atributos.

5.4.1 Objetivo del segundo experimento

El objetivo de este segundo experimento es ver ahora cómo se comporta el método con una muestra de entrenamiento clásica y tomando como referencia los mismos parámetros del primer experimento.

Veremos el comportamiento con diferentes valores del k, los datos son 3, 9, 15, 21, 27, para este experimento con la muestra de entrenamiento Iris.

Para el primer experimento los parámetros que se utilizaron son: población de 100 individuos y un máximo de 100 generaciones con probabilidad de mutación del 5%, ver tabla 5.4-1.

Tabla 5.4-1 Parámetros del segundo experimento

K-NN	Elite	Generaciones	Número de Individuos	Mutación (%)
3,9,15,21,27	2	100	100	5

5.4.2 Diferentes valores de k

Probar diferentes valores para k del K-NN, para saber cuál es el mejor valor que ayuda al algoritmo genético, para obtener buenos resultados cuando clasifica los grupos obteniendo una buena función de aptitud.

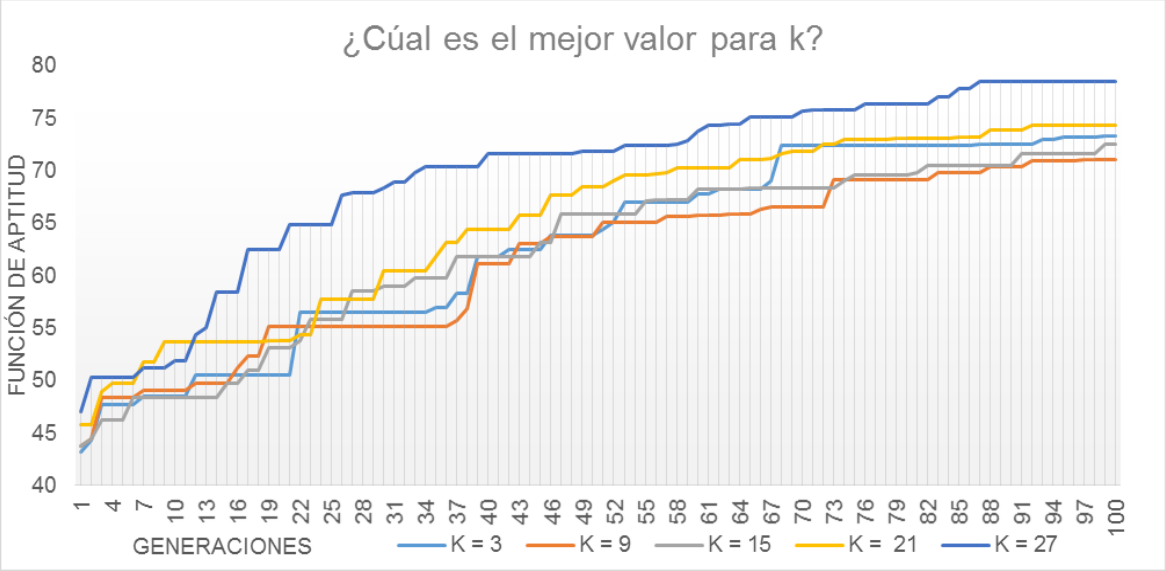


Figura 5.4-1 Gráfica de diferentes valores de k, con la base de datos Iris y selección por ruleta.

Entonces se puede decir que el mejor parámetro de k, es cuando k es igual a 27, adonde se alcanzó una función de aptitud de 78.4, cabe mencionar que estos resultados son con el operador de selección por ruleta.

5.4.3 Diferente mutación

Ahora que se ha visto el resultado con el operador de selección de ruleta, se va a probar con otro operador de selección. Lo que sigue es utilizar el operador de selección por torneo.

En la gráfica de la Figura 5.4 2 se observa los resultados de la selección por torneo y nos arrojó que después de 100 generaciones, igualmente con un valor de 27 para k, es con este valor, en el que obtiene la más alta función de aptitud para la muestra de datos Iris, en la cual se ha llegado a alcanzar una función de aptitud de 88.6 superando a la de la selección por ruleta de 78.4 teniendo un aumento de casi 10.2 en la función de aptitud.

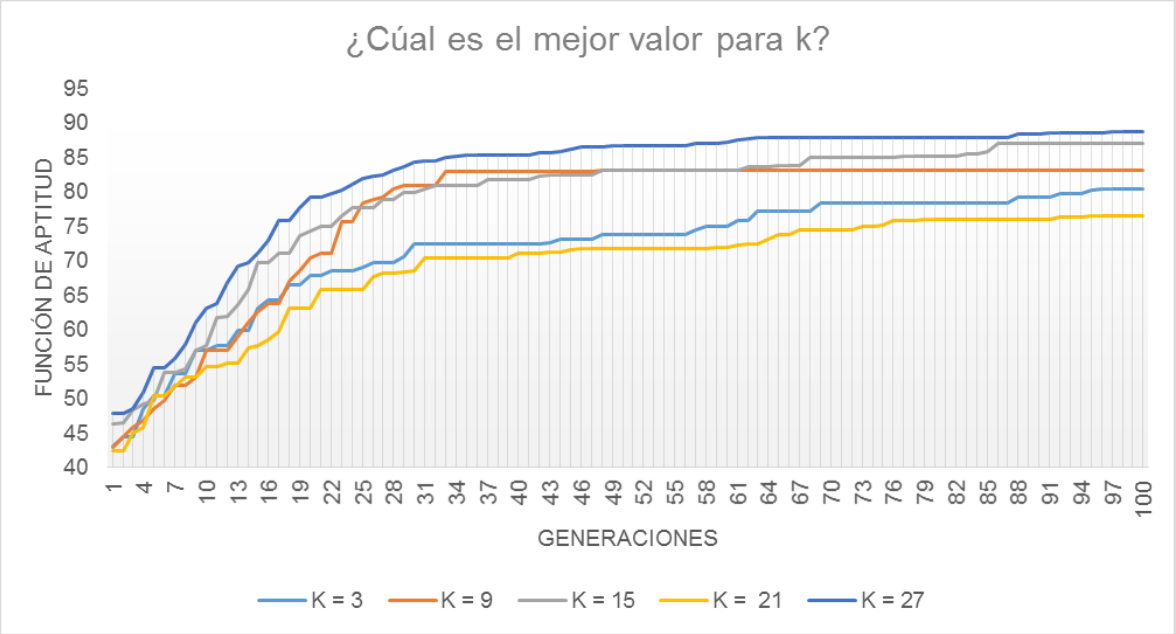


Figura 5.4-2 Gráfica de diferentes valores de k, con la base de datos Iris y selección por torneo.

5.4.4 Diferente operador de selección

Al cambiar el operador de selección se busca saber si mejora la búsqueda de solución y aumenta la función de aptitud, en la gráfica siguiente podemos ver los comportamientos para cada operador de selección.

Se puede ver en la figura 5.4-3 que efectivamente con el cambio del operador de selección se mejoran los resultados, en donde el algoritmo genético con el operador de selección por ruleta alcanzó una función de aptitud de 78.4 después de 100 generaciones. En cambio, el algoritmo genético con el operador de selección por torneo consiguió una función de aptitud de 88.6, siendo una de las mejores soluciones, ver figura 5.4-3

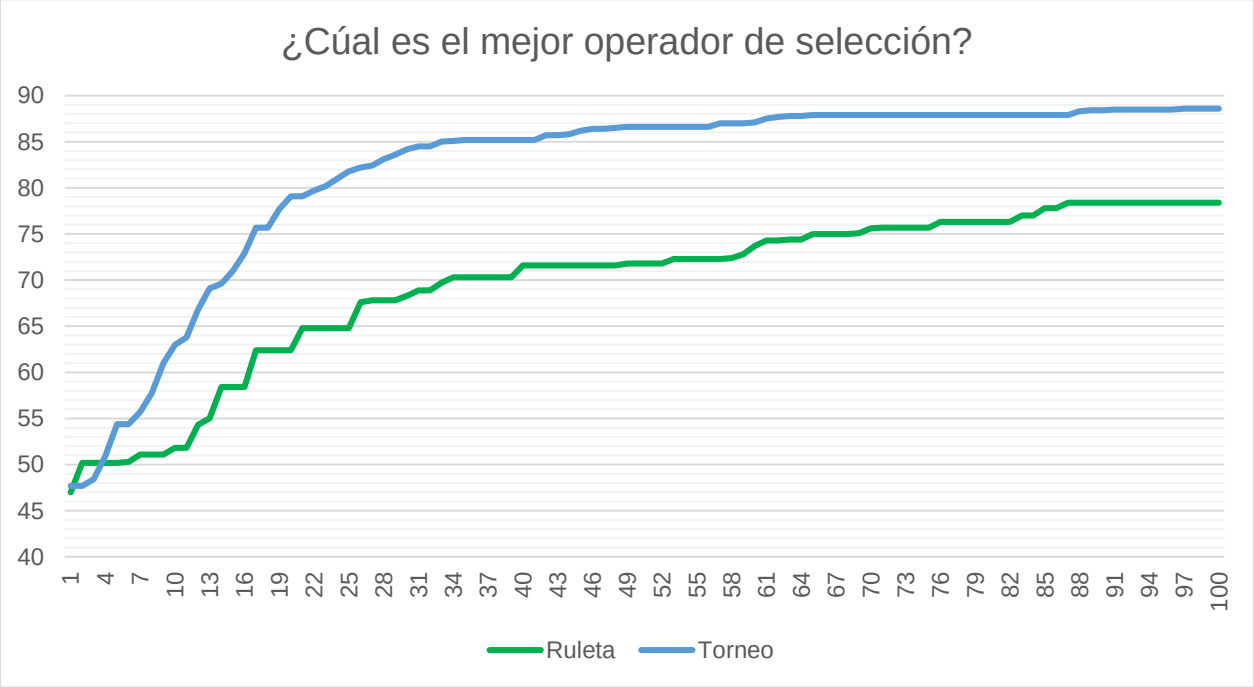


Figura 5.4-3 Gráfica de los diferentes operadores de selección

También podemos ver que la evolución del algoritmo genético es progresiva y no da saltos demasiado abruptos en el paso de una generación a otra, es decir el torneo es idóneo porque va progresivamente y evolucionando de forma paulatina.

5.4.5 Diferente Mutación

Después de los experimentos anteriores se sabe que con el operador de selección de torneo se obtienen buenos resultados, el valor de k del KNN cuando es 27 también ayuda al resultado.

En el siguiente experimento se ve si variando el valor de la mutación, se mantiene o se mejora el resultado de la función de aptitud. Para este caso se prueban diferentes probabilidades de mutación, las cuales son mutación de 5, 10, 15 y 25%.

Como podemos observar en la Figura 5.4-4; variando los valores de la mutación, el valor que da más alta la función de aptitud, es cuando la probabilidad de mutación es de 5%, ya que si se aumenta se reduce el valor de función de aptitud.

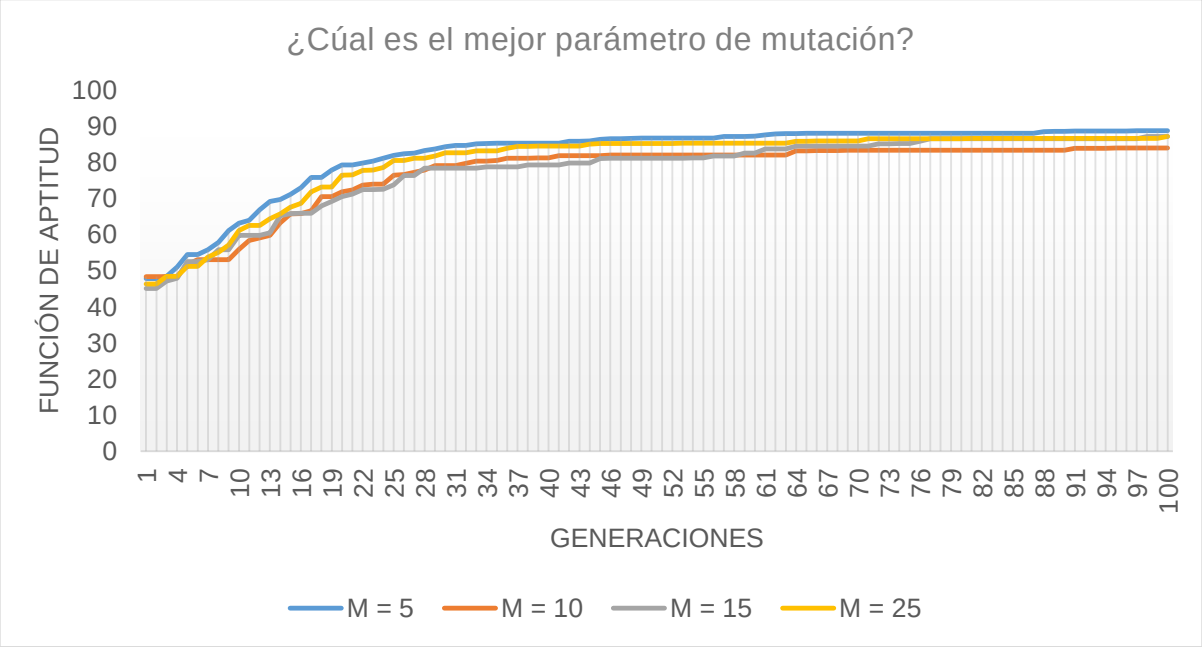


Figura 5.4-4 Gráfica variando el parámetro de mutación

5.4.6 Tamaño del operador por Torneo Selección(k)

Un parámetro fundamental de la selección por torneo, es el número de individuos (k) que se utilizan para el torneo. Las pruebas anteriores se probaron con un valor de 3. Es

decir, se toman solo 3 individuos para que se obtenga el mejor individuo, Sin embargo, ahora se quiere ver si cambiando este parámetro aumenta la función de aptitud.

Los valores que se probaran para k serán de 3, 5, 9, esperamos que variando se pueda obtener una función de aptitud aún más alta de la ahora alcanzada ver figura 5.4-5.

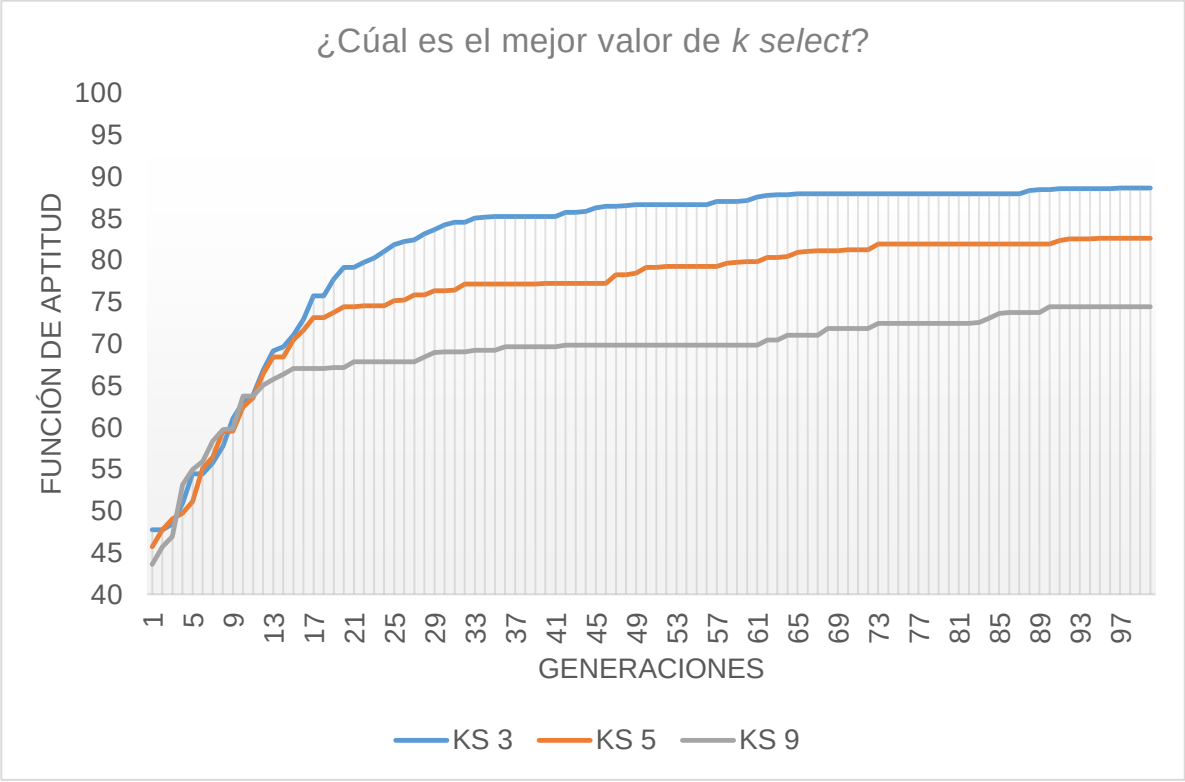


Figura 5.4-5 Gráfica con diferentes valores de k-select

Como se puede observar en la gráfica de la figura 5.4-5, el valor que mejor resultado da para k-select es cuando tiene un valor de 3.

5.4.7 Conclusiones del segundo experimento

Después de haber realizado los experimentos anteriores, podemos concluir en este segundo experimento que, usando el operador selección por torneo, tenemos mejores resultados a los obtenidos con ruleta.

Al tratar de optimizar cambiar los parámetros del algoritmo genético, se pueden obtener mejores resultados que los iniciales. Esto permite aumentar la función de aptitud del método propuesto acercándonos a la clasificación original de k vecinos de 94.6.

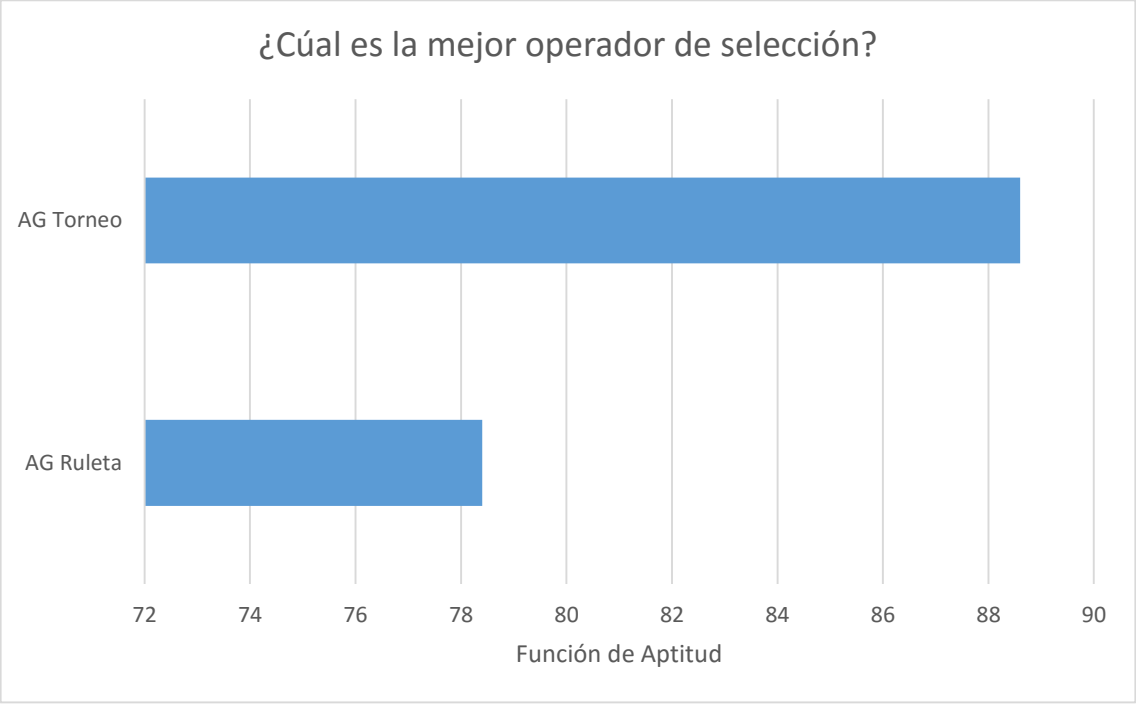


Figura 5.4-6 Gráfica con resultados de diferentes operadores de selección

Hasta este momento los parámetros que tenemos y nos han arrojado el mejor resultado son para el *k* vecinos el valor de *k* =27, para la selección de individuos el operador de selección por torneo, el *k select* del torneo cuando el valor es de 3, para la mutación cuando su valor es del 5% ver tabla 5.4-2.

Tabla 5.4-2 Mejores Parámetros para el Algoritmo Genético

K-NN	K-Select	Mutación (%)
27	**3**	**5**

5.5 Tercer experimento

5.5.1 Objetivo del tercer experimento

El objetivo de este experimento es ver qué población es mejor para el algoritmo genético con el método propuesto, en los primeros experimentos se utilizó generaciones de 100, ahora al variar los valores con el veremos si es posible llegar a un número aún mayor en nuestra función de aptitud, el valor de esta hasta ahora es 88.6 y se espera que modificando el número de generaciones aumente la función de aptitud.

Los otros valores con los que se probará el algoritmo genético son con generaciones de 150 y 200 con los parámetros que llevamos hasta el momento hemos encontrado como los mejores.

Es decir, el valor del k, del k vecinos es de 27, operador de selección con un k-select de 3 una mutación de 5% ver tabla 5.4-2.

5.5.2 Diferentes Generaciones

Los resultados que arroja al cambiar el número de generaciones es que, a las 100 generaciones, la función de aptitud que se obtiene es de 88.6%.

Los resultados que arrojan con 150 generaciones es de 81.59, en este caso disminuye la función de aptitud que se logró con 100 generaciones.

Para el experimento de 200 generaciones la función de aptitud da 86.03, En este caso aumentó con respecto al experimento de 150 generaciones, pero a un no alcanza la función de aptitud, lograda con 100 generaciones.

Como se puede observar en la gráfica de la Figura 5.5-1, con 100 generaciones es donde se encuentra el mejor resultado, después la que da mejores resultados es cuando se tiene 200 generaciones y al final 150 generaciones.

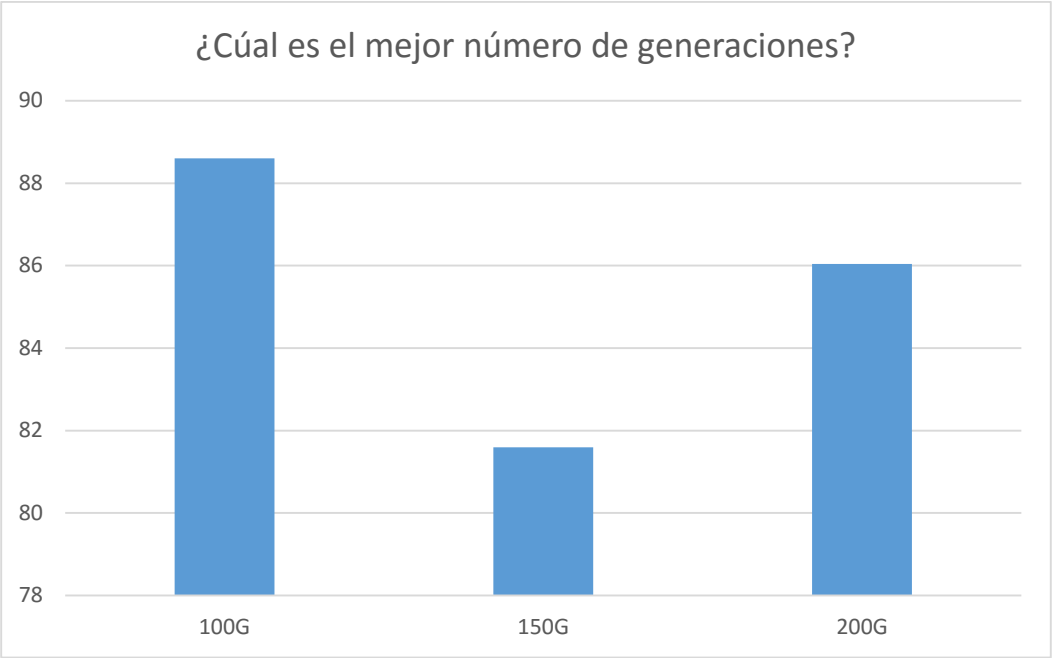


Figura 5.5-1 Gráfica con diferente número de generaciones

Al terminar con estas pruebas, obtenemos que los mejores parámetros que se han obtenido hasta ahora son los que se muestran en la Tabla 5.5-1.

Tabla 5.5-1 Mejores parámetros, Generaciones

K-NN	K-Select	Generaciones	Mutación (%)
27	27	100	5

5.5.3 Conclusiones del tercer experimento

Con este experimento se pudo ver que, si mantenemos una generación de 100, se obtiene una función de aptitud alta con respecto al principio de los experimentos. Como se puede ver al cambiar los parámetros del algoritmo genético, realmente ayuda a

mejorar el resultado, con generaciones de 100 o 200 se obtienen buenos resultados que, si ponemos un valor intermedio de estos datos, es decir 150 generaciones.

5.6 Cuarto Experimento, los mejores resultados

Hasta el momento se podría decir que estos son los mejores parámetros que se han obtenido a través del proceso de prueba y que son los mejores y que nuestro mejor resultado es de 88.6% en la función de aptitud.

Se tienen dos experimentos en los cuales podríamos decir que son aún mejores que los anteriores, que siguen el patrón de generación de 100 o 200, pero la diferencia radica en que al cambiar el número de individuos se obtienen mejores resultados de los anteriores.

5.6.1 Objetivo del cuarto experimento

En este cuarto experimento veremos los tres mejores resultados obtenidos y con qué parámetros se han obtenido, como se mencionó los parámetros con 100 y 200 generaciones nos arrojan mejores resultados hasta ahora que los de 150.

5.6.2 Parámetros de los experimentos

El primer experimento es el que se ha venido mejorando con los cambios de parámetros del algoritmo genético, en el que nos arrojó un valor dio 88.6% en la función de aptitud los valores con los que se obtuvo son los de la Tabla 5.6-1.

Con estos parámetros se obtuvo la gráfica de la Figura 5.6-1, en la cual vemos que tiene una evolución progresiva las primeras 34 generación, después de ahí se va manteniendo hasta encontrar la más alta función de aptitud, vemos que la mutación ayuda a ir mejorando en la generación 37, 40, 46, 64.

Tabla 5.6-1 Mejores parámetros para el Algoritmo Genético

K-NN	Elite	K-Select	Generaciones	Número de Individuos	Mutación (%)
27	2	3	100	100	5

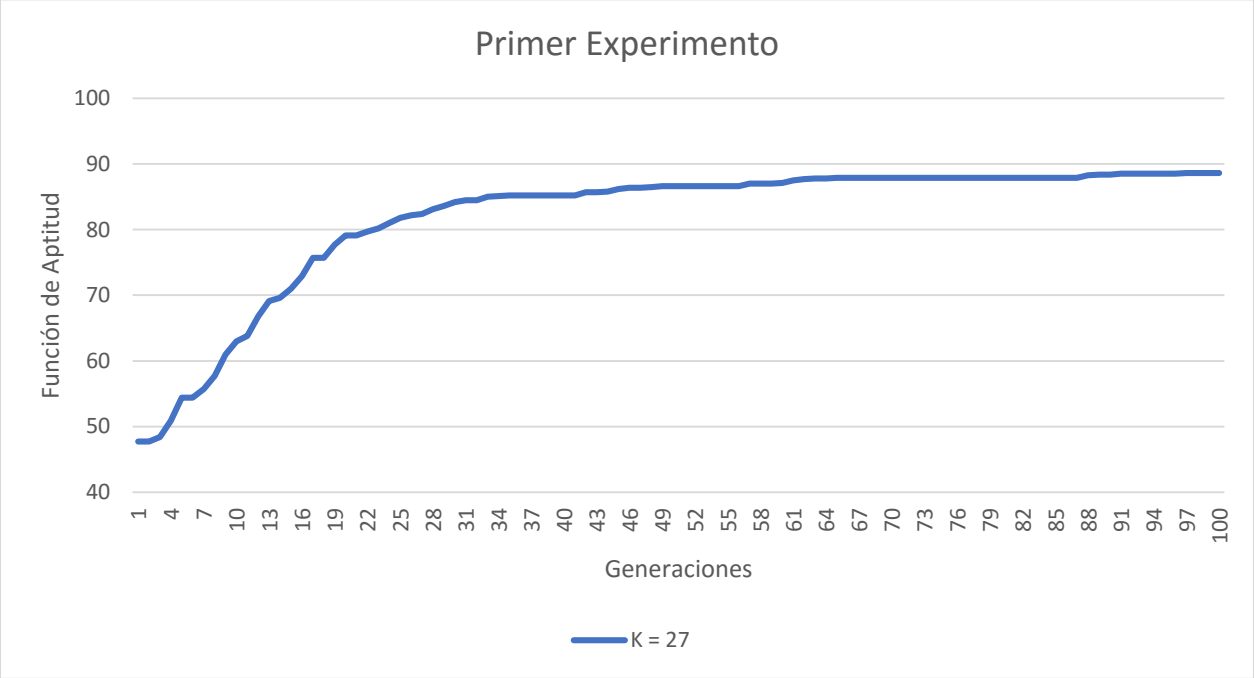


Figura 5.6-1 Gráfica del primer experimento

El segundo experimento en que se obtuvieron buenos resultados fue con los parámetros que vemos en la Tabla 5.6-2, con estos parámetros se amentó el valor de la función de aptitud con respecto al experimento uno, y el valor que alcanzó es de 89.3%.

Como podemos observar en este segundo experimento, los únicos parámetros que son iguales al experimento uno, son el número de selección elite, el valor de K-NN con k = 27. Otro parámetro que comparten es el valor para k–select el cual tiene valor de 3,

todos los demás parámetros son diferentes, en el número de individuos, las generaciones y la mutación.

Tabla 5.6-2 Parámetros del segundo experimento

K-NN	Elite	K-Select	Generaciones	Número de Individuos	Mutación (%)
27	2	3	200	50	15

Con cambios de parámetros se aumenta un 0.7% respecto al primer experimento, entonces veamos cómo es su comportamiento después de 200 generaciones Ver Figura 5.6-2.

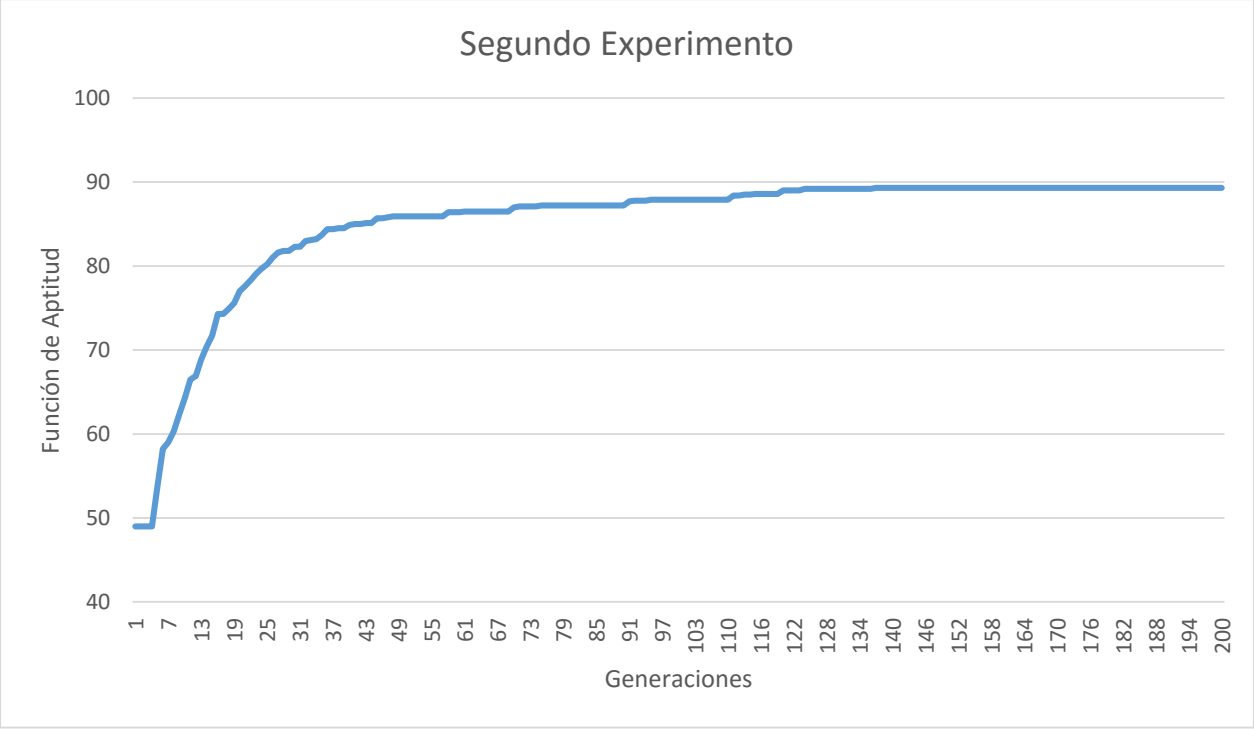


Figura 5.6-2 Gráfica del segundo experimento

En el tercer y último experimentó que veremos es en el que se obtuvo la función de aptitud más alta, con un porcentaje de 91.8, en este experimento se incrementó un 2.5% en la función de aptitud, con respecto a los experimentos dos y un 3.2 con respecto al experimento uno. Se puede ver que, en relación a los experimentos anteriores, en el primer experimento y en este tercer experimento, solo tiene relación de parámetros con el porcentaje de la mutación con un valor de 5%, con el número de generaciones con 100 los demás parámetros del algoritmo genético son diferentes (Ver tabla 5.6-3). Con respecto al tercer experimento y el segundo no tienen relación en parámetros.

Tabla 5.6-3 Parámetros del tercer experimento

K-NN	Elite	K-Select	Generaciones	Número de Individuos	Mutación (%)
9	10	3	100	150	5

Ahora si revisamos los parámetros del tercer experimento se puede ver que, como se mencionó anteriormente, que con generaciones de múltiplos de 100 se obtenían buenos resultados y en los tres casos se ve claro esto aún más en el caso del tercer experimento se cumple está condición; caso contrario en el número de los individuos, donde con valores de 50, 100, 150 se obtuvieron buenos resultados para el parámetro de los individuos.

También se puede ver en el tercer experimento, que la selección élite aumenta su valor a 10, al igual que el número de individuos. A diferencia de estos paramentos, la mutación se redujo un 5%, el valor de KNN a un menor valor de 9 y en el número de generaciones a 100. En la gráfica de la Figura 5.6-3 se puede ver su comportamiento.

En la gráfica de este tercer experimento, podemos ver que desde el principio empieza a evolucionar y la mutación realmente le ayuda, puesto que después de 48 generaciones

ya está alcanzando una función de aptitud de 87.7, si seguimos la línea de la gráfica después de 88 generaciones ya tiene un 89.1 superando a los otros experimentos después de este punto se empieza a estabilizarse, evolucionando un poco más lento y hasta la generación 150 encuentra la más alta función de aptitud de 91.8%.

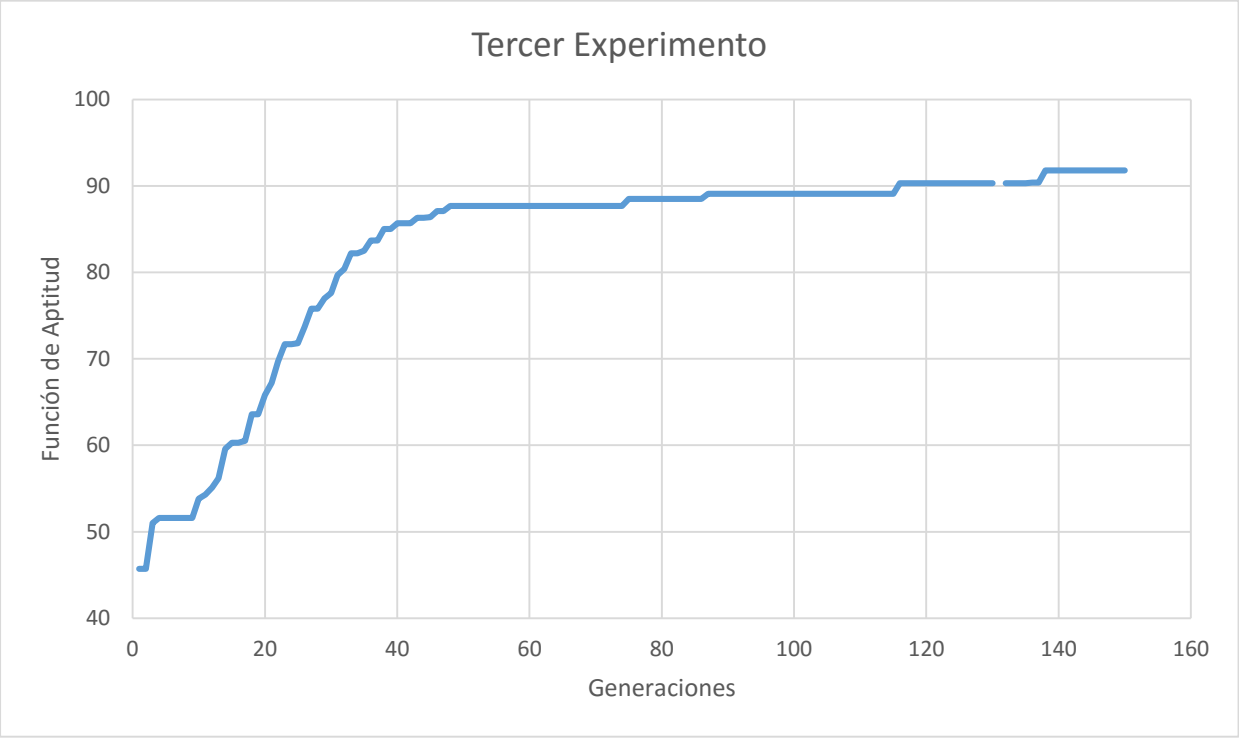


Figura 5.6-3 Gráfica del tercer experimento

5.6.3 Conclusiones del tercer experimento

Como conclusiones a este tercer experimento podemos decir que los mejores parámetros se obtienen con una generación de 150, y un número de individuos de 200; minimizando el parámetro de KNN. También podemos, ver también que, si se aumenta la selección elite, este parámetro nos ayuda demasiado en la búsqueda de la solución, como se había visto hasta el momento con el valor de 2 se obtuvieron buenos resultados, hasta que se encontró que con una elite de 10 nos da un resultado aún mucho.

También se puede concluir que en la mayoría de los experimentos se utiliza una mutación de 5% y da buenos resultados, en el caso de tercer experimento nos dio aún una función de aptitud mucho mayor, haciendo que este experimento sea el experimento que más se acerca al resultado que arroja la clasificación original de la muestra de entrenamiento de IRIS con KNN.

5.7 Resumen

En este capítulo se desarrollaron experimentos para demostrar que el método que se propone funciona, como ya se vio a lo largo de los experimentos. Se variaron los parámetros para ver si al ir cambiando los parámetros del algoritmo genético para obtener se podría obtener buenos resultados

Los experimentos se realizaron con dos muestras de entrenamiento, una sintética y una clásica, los resultados que nos dio con la muestra de entrenamiento sintética es que el método realmente encuentra los grupos naturales de esta muestra de entrenamiento encontrándolos en 6 generaciones los grupos naturales.

Después de los primeros experimentos se prosiguió con la base de datos clásica (Iris), la cual con los parámetros del primer experimento se obtuvo un 78.4%, con el operador de selección por ruleta. A partir de la utilización de estos experimentos, de acuerdo a los resultados obtenidos, se comenzará a buscar los parámetros que ayudaron al algoritmo genético a que encuentre una de las mejores soluciones.

El primer parámetro que se cambió fue el operador de selección, el mejor operador de selección es por torneo, el cual nos dio una clasificación de 88.6, superando por mucho a la clasificación por ruleta. Dentro de este operador de selección también había que buscar cuál era el mejor valor para el número de individuos que se seleccionarían en el torneo y el valor que mejor dio resultados fue cuando vale 3.

Con este resultado se variaron otros parámetros, uno de ellos es el valor de k vecinos, en el cual se comenzó con un valor alto de k=27, al final de los experimentos se

encontró que con un valor de 9 da muy buenos resultados, superando por mucho a lo que arroja con k 27.

El parámetro que más ayudó a mejorar a la hora de obtener el resultado fue el valor de la selección elite, al inicio de los experimentos se tenían un valor de 2; siendo un buen valor, pero al reducir el valor del k vecinos se reducía la clasificación con el valor de 2, aumentando este parámetro ayudó demasiado para encontrar el valor de 91.8%.

El número de generaciones y el número de los individuos, se vio que para el caso de las generaciones con 100 y 200 nos arrojó buenos resultados, pero con una reducción entre estos valores de 150 nos arrojó el valor más alto, en cuanto a los individuos un valor de 50 fue el más adecuado.

En cuanto a la mutación de todos los parámetros que se cambiaron es el único que prácticamente se mantiene en casi todos los experimentos, en el caso más visible donde ayudó es en el último experimento.

Entonces podemos concluir que la variación en los parámetros ayuda mucho, a encontrar el resultado.

En la Tabla 5.8, se muestra la clasificación que le dió el método, en la parte izquierda el método original y en el lado derecho lo que encontró el método propuesto.

Tabla 5.7-1 Resultados del Método Propuesto

NG	GO	GM
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1

Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	1

Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	1

Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	2
Iris-setosa	1	1
Iris-setosa	1	2
Iris-setosa	1	2
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	3
Iris-versicolor	2	2
Iris-versicolor	2	2

Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	3
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	3
Iris-versicolor	2	1
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	2
Iris-versicolor	2	1
Iris-versicolor	2	1
Iris-versicolor	2	2
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	2
Iris-virginica	3	1
Iris-virginica	3	3

Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	2
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	2
Iris-virginica	3	2
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	2
Iris-virginica	3	2
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	1
Iris-virginica	3	3
Iris-virginica	3	3
Iris-virginica	3	1
Iris-virginica	3	3

Donde:

NG = nombre del grupo

GO = Grupo original

GM = Grupo del método

En la tabla podemos apreciar que se está cumpliendo con el método hizo lo esperado, al parecer la clase 3 es el más pequeña, pero se está cumpliendo con la validación de los máximos y mínimo para cada grupo. Donde para este caso el mínimo es de 16 por grupo y máximo por grupo puede tener 96, en este caso los valores son: 1s = 62, 2s =70 y 3s =18.

CAPÍTULO VI CONCLUSIONES, APORTACIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

Revisando los objetivos que se planeaban al inicio de este trabajo podemos concluir lo siguiente.

Se pudo realizar el algoritmo de agrupamiento basado en un algoritmo de genético, que este dirigido por un algoritmo de clasificación, en este caso se utilizó el k-vecinos más cercanos para que funcionará como la función de aptitud del algoritmo genético.

Con base en lo anterior, se puede ver que si es posible utilizar el mismo criterio tanto para agrupar como para clasificar.

Con el método propuesto no se intentó igualar la clasificación que realiza KNN con la muestra de entrenamiento original que es de 94.6% de precisión. Con el método propuesto se alcanzó 91.8% sin embargo el criterio de clasificación y de agrupamiento es el mismo. Podemos ver que el método de agrupar y clasificar con el mismo criterio está casi satisfecho mientras que weka obtiene una clasificación para IRIS de 94,6 de clasificación y el método propuesto está obteniendo un 91.8 de clasificación con los mejores parámetros que se encontraron algoritmo genético.

6.2 Aportaciones

La aportación que se hace con este trabajo es que se abre un panorama para utilizar los criterios para los dos tipos de aprendizaje tanto supervisado como no supervisado.

Aunque ya hay trabajos que intentan clasificar para agrupar, o intentan hacer clasificación semi-supervisada, el método nos ayuda a utilizar desde el principio el

mismo criterio, es decir desde que se tienen los datos sin etiqueta el algoritmo lo va llevando de la mano hasta encontrar los grupos y luego clasificarlas. En cada interacción del algoritmo genético va buscando los mejores resultados, con esto podemos iniciar en el camino de probar más clasificadores y más agrupamientos para encontrar buenos resultados que sirvan en las dos áreas de clasificación.

6.3 Trabajos futuros

Como trabajos futuros para este trabajo se propone probar otros criterios para agrupar y clasificar, es decir otros clasificadores, viendo que un algoritmo de k-vecinos más cercanos funciona bien.

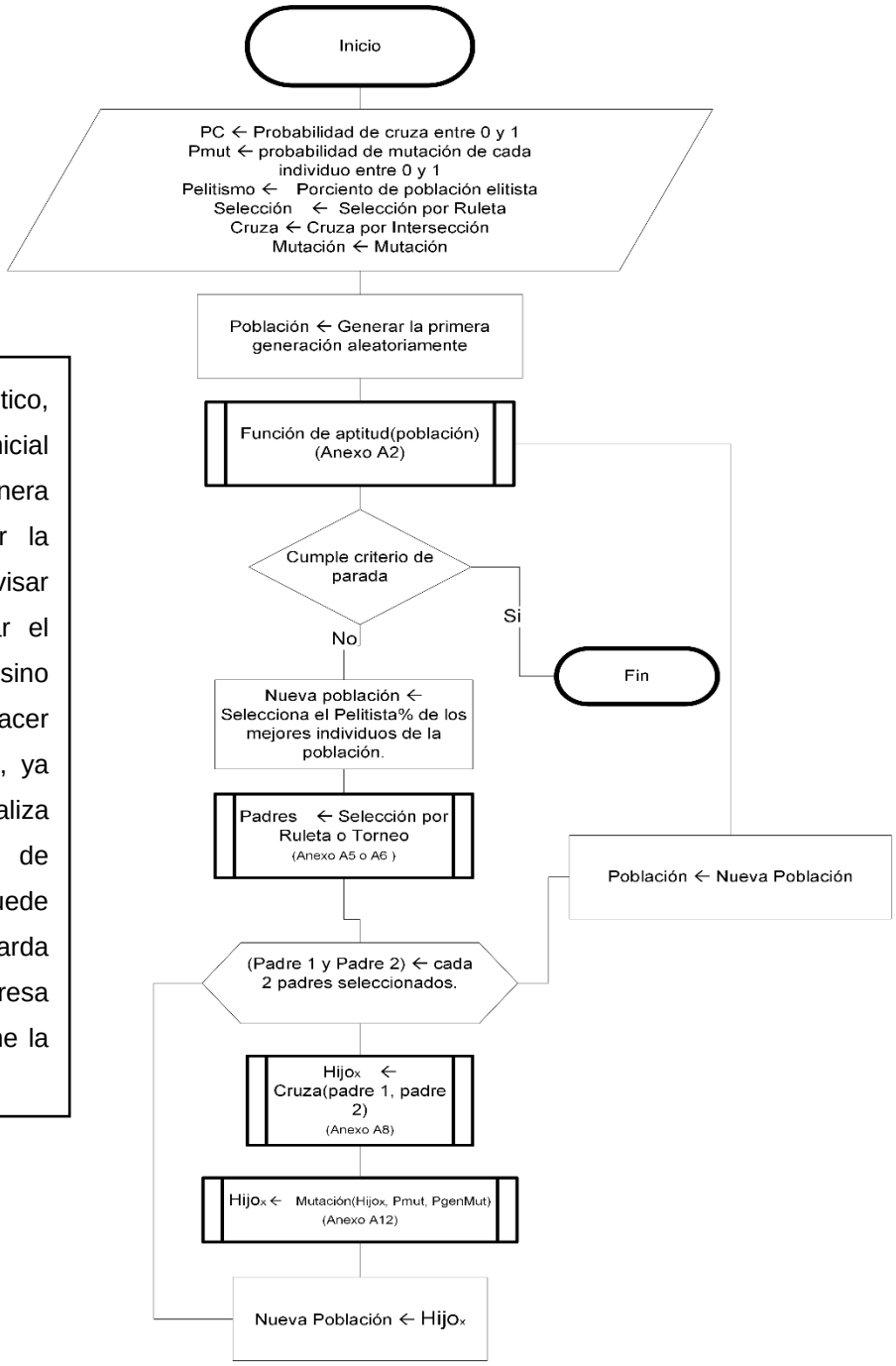
Como trabajo futuro es utilizar este método para el problema del desbalance de clases, ya que el método que se desarrolló, al poder especificar, el número de grupos que debe realizar dependiendo de los objetos de la muestra de entrenamiento y la restricción de números mínimos y máximos para cada grupo que se genera, ayuda a el desbalance de clases, puesto que, al utilizar la restricción de elementos por grupo, reduce el problema y mantendría o podría aumentar la confiabilidad del clasificador.

Otro trabajo futuro podría ser que el algoritmo ya no necesite de parámetros dados por el usuario, si no que el algoritmo pueda tomar la decisión de cuantos grupos debe generar y qué paramentos hay que variar.

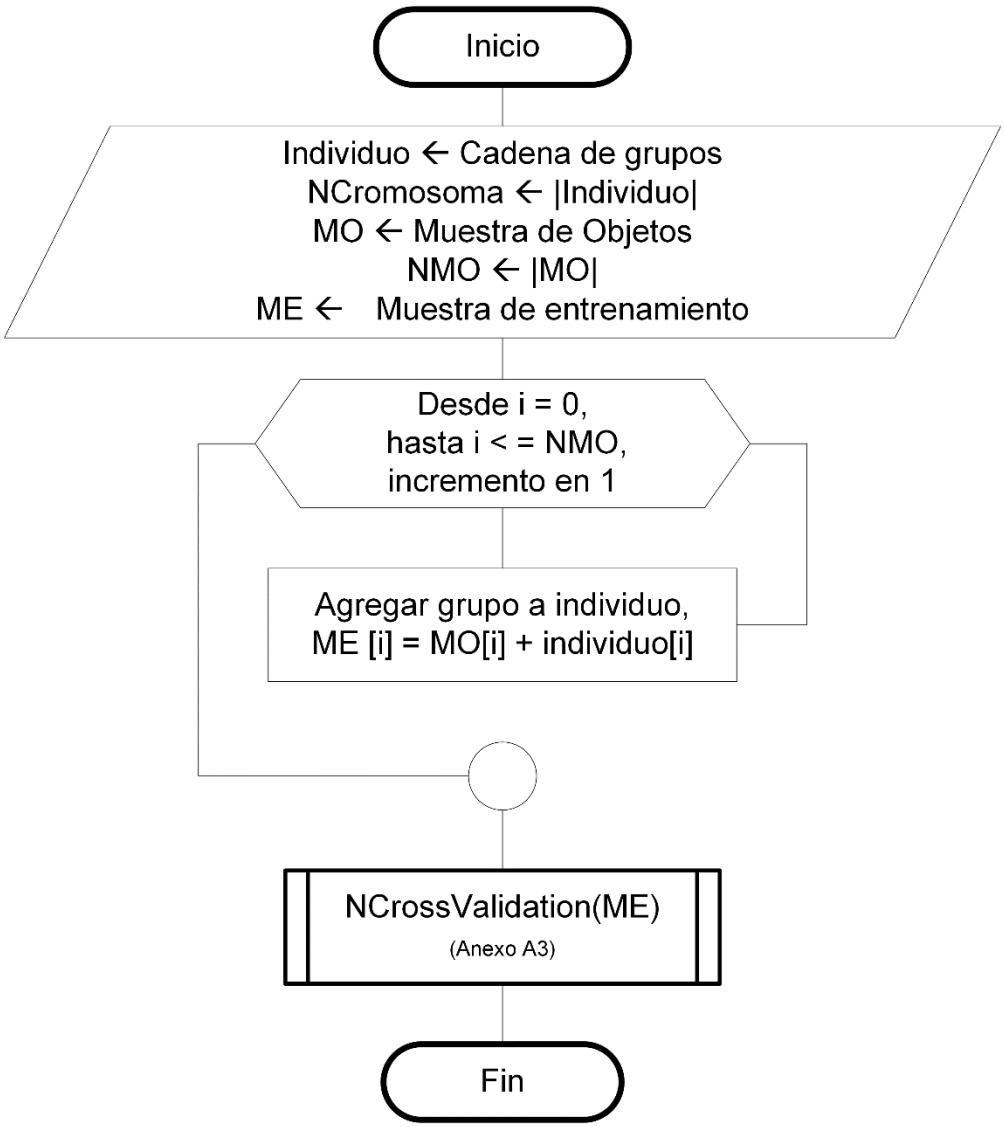
ANEXOS

Anexo A1: Diagrama de flujo del Algoritmo General

Pasos del algoritmo genético, obtener la población inicial generándola de manera aleatoria, después evaluar la aptitud de la población, revisar las condiciones para parar el algoritmo genético, sino continuar, a continuación, hacer la selección de los padres, ya seleccionado los padres, realiza la cruza de individuo y de acuerdo a la probabilidad puede o no haber mutación, se guarda el nuevo individuo y se regresa al punto de ver si se detiene la ejecución.



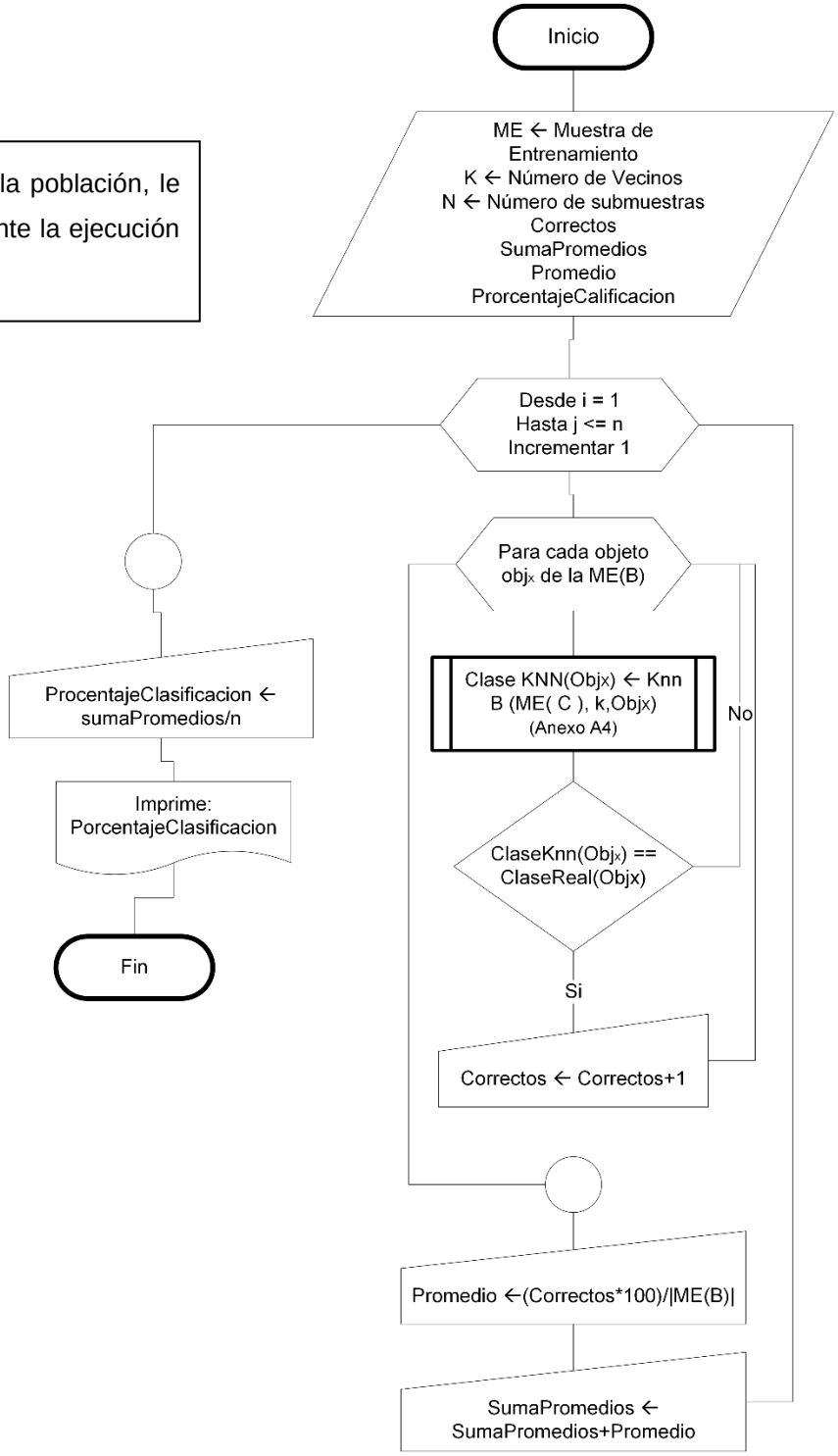
Anexo A2: Diagrama de flujo de la Función de Aptitud



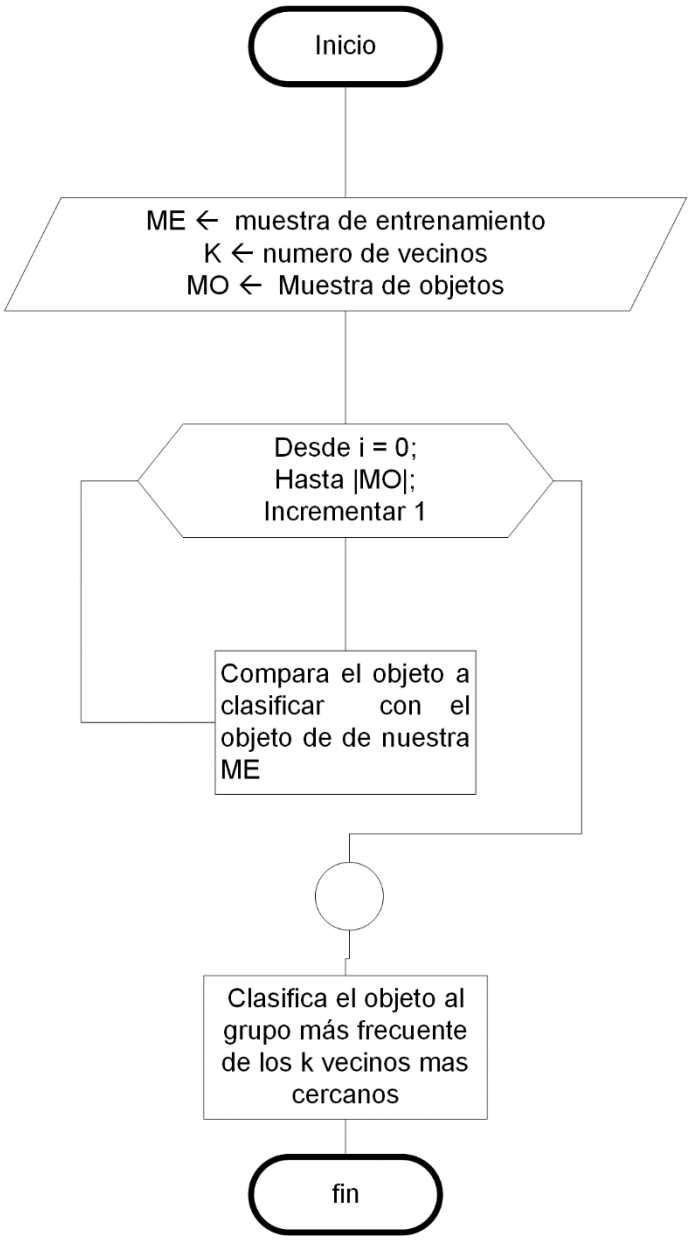
Obtiene la función de aptitud de para todos los objetos de la muestra de entrenamiento.

Anexo A3: Diagrama de flujo de la Validación Cruzada *(Cross Validation)*

Evaluar a todos los individuos de la población, le da un valor de clasificación mediante la ejecución de un algoritmo de K-NN.

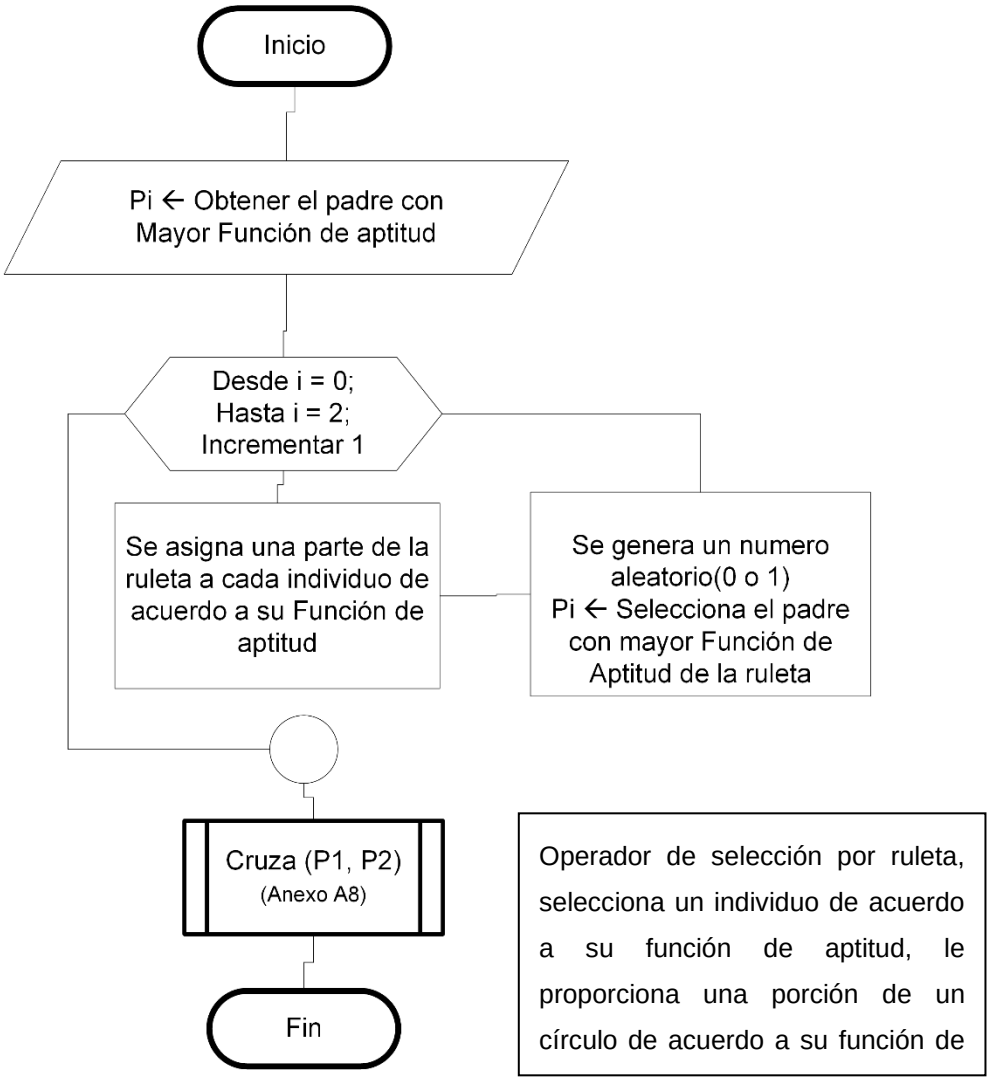


Anexo A4: Diagrama de flujo de los K-vecino más cercanos



Se obtiene la muestra de entrenamiento y se carga la muestra, se extraen todos objetos, se compara con la muestra de objetos con las clases originales y clasifica, se clasifica al objeto al que más se parezca y se realiza para todos los objetos de la

Anexo A5: Diagrama de flujo del Operador de Selección por Ruleta



Anexo A6: Diagrama de flujo del Operador de Selección por Torneo

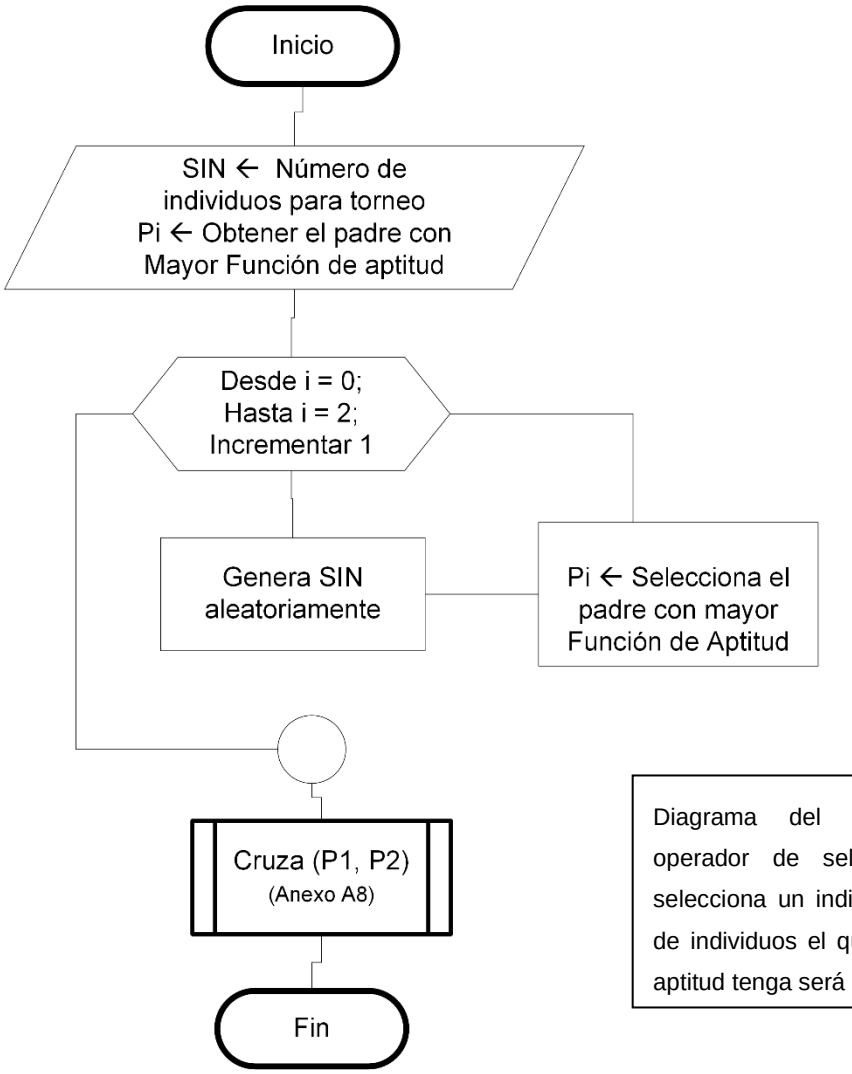
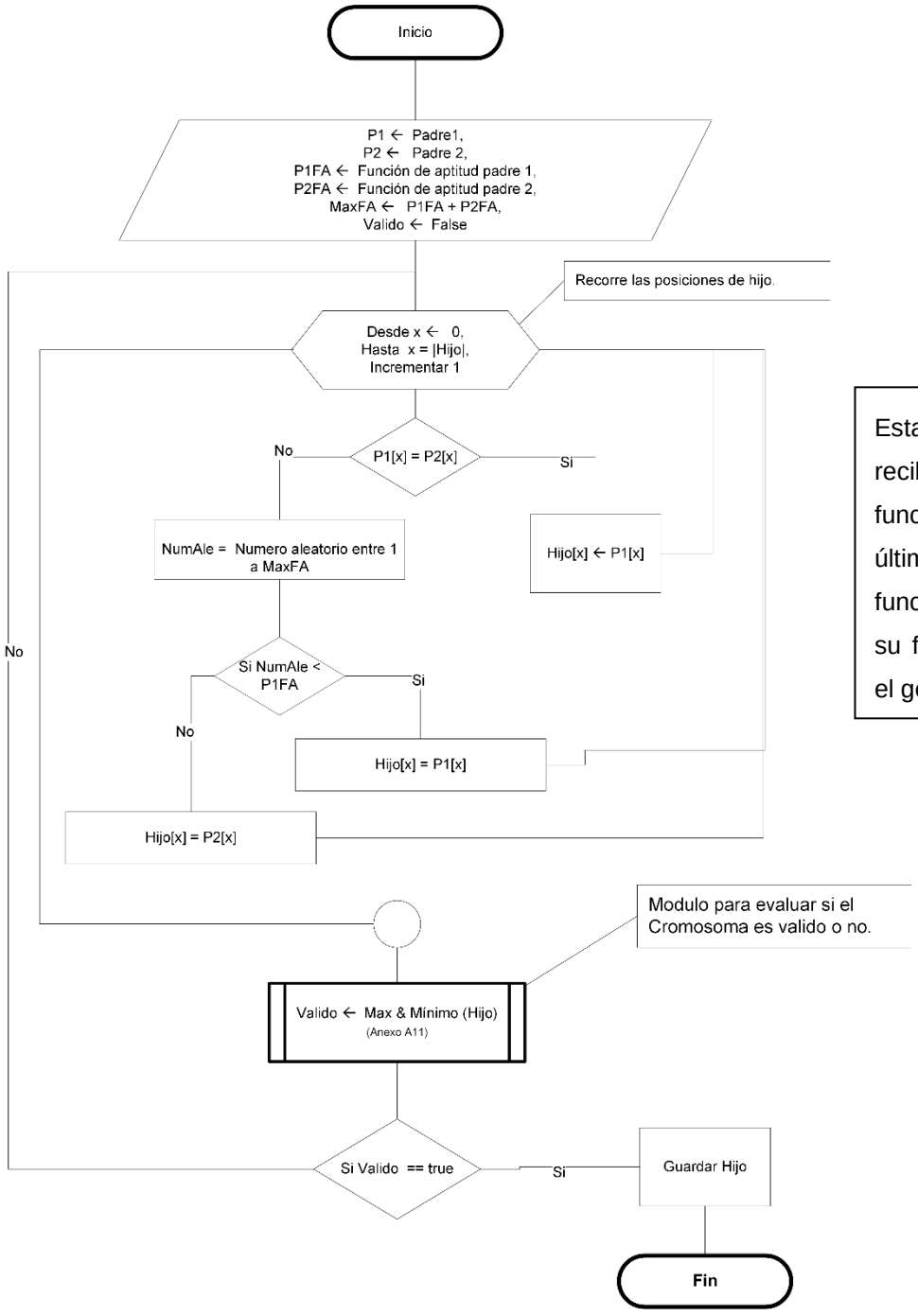


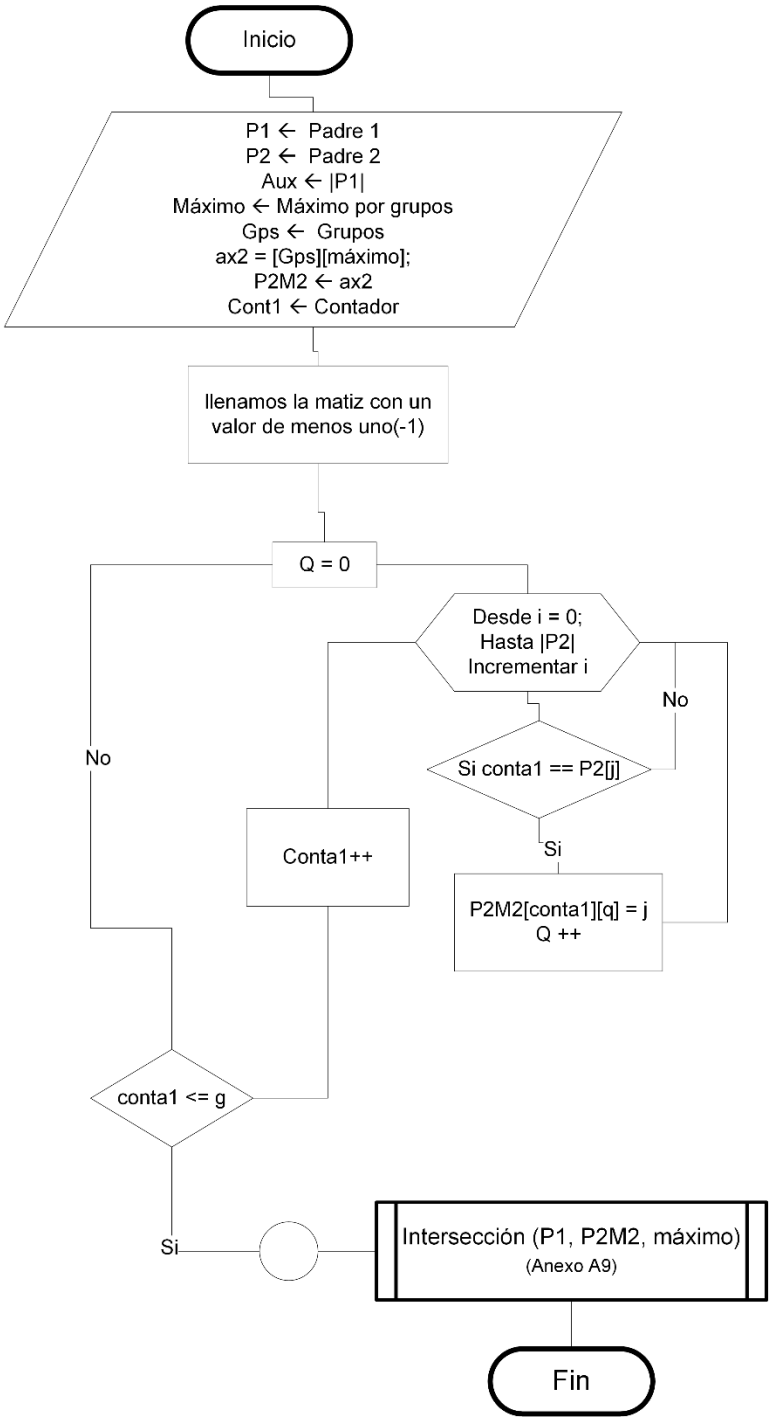
Diagrama del funcionamiento del operador de selección por torneo, selecciona un individuo de un número de individuos el que mayor función de aptitud tenga será el elegido.

Anexo A7: Diagrama de flujo del Operador de Cruza por Intersección



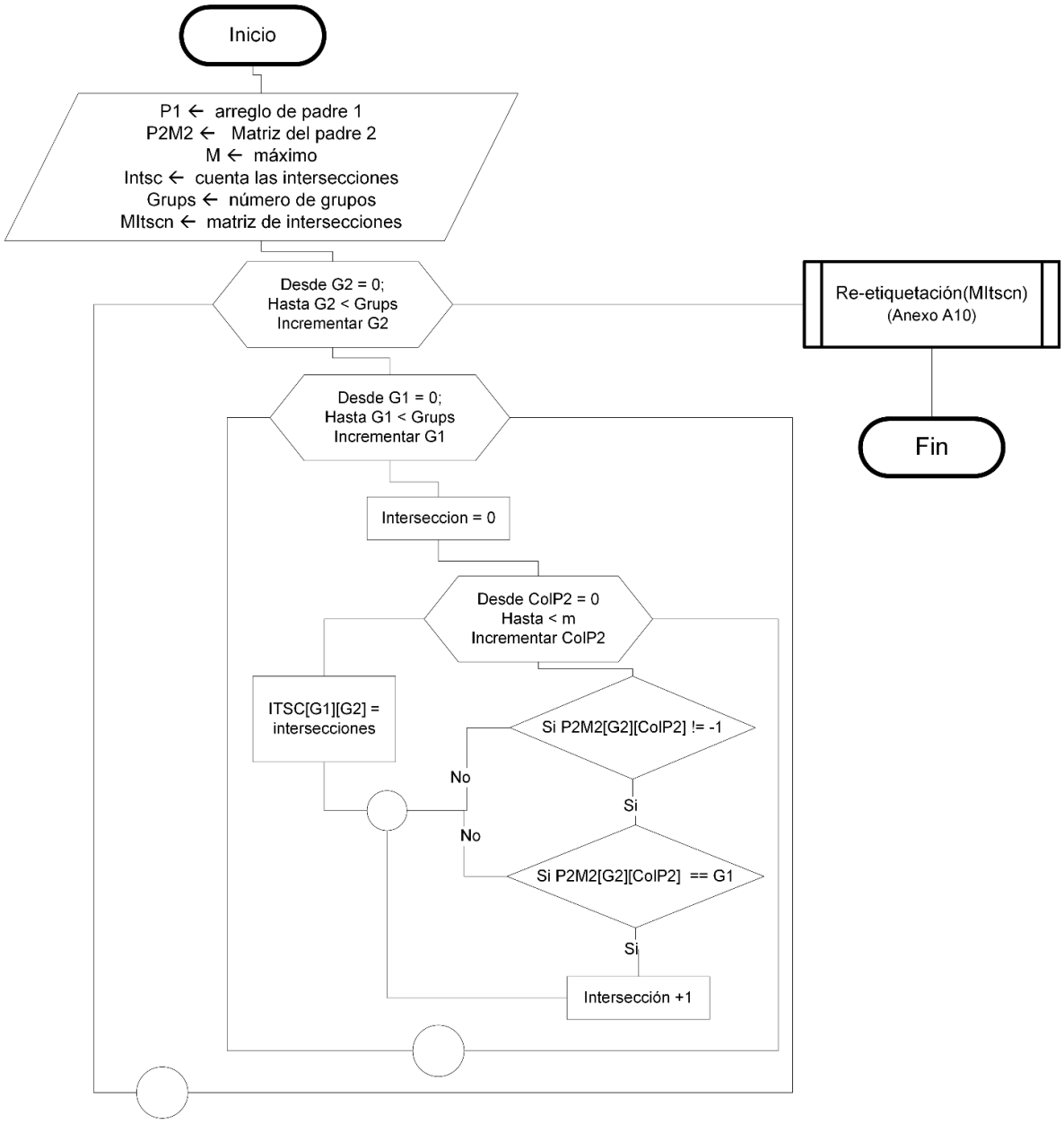
Esta clase realizara la cruza es recibir los padres y sus respectivas funciones de aptitud (FA), con esta última generara la suma de las dos funciones de aptitud de acuerdo a su función de aptitud se le pasara el gen de acuerdo al padre.

Anexo A8: Diagrama de flujo del Métodos de Cruza Intersección



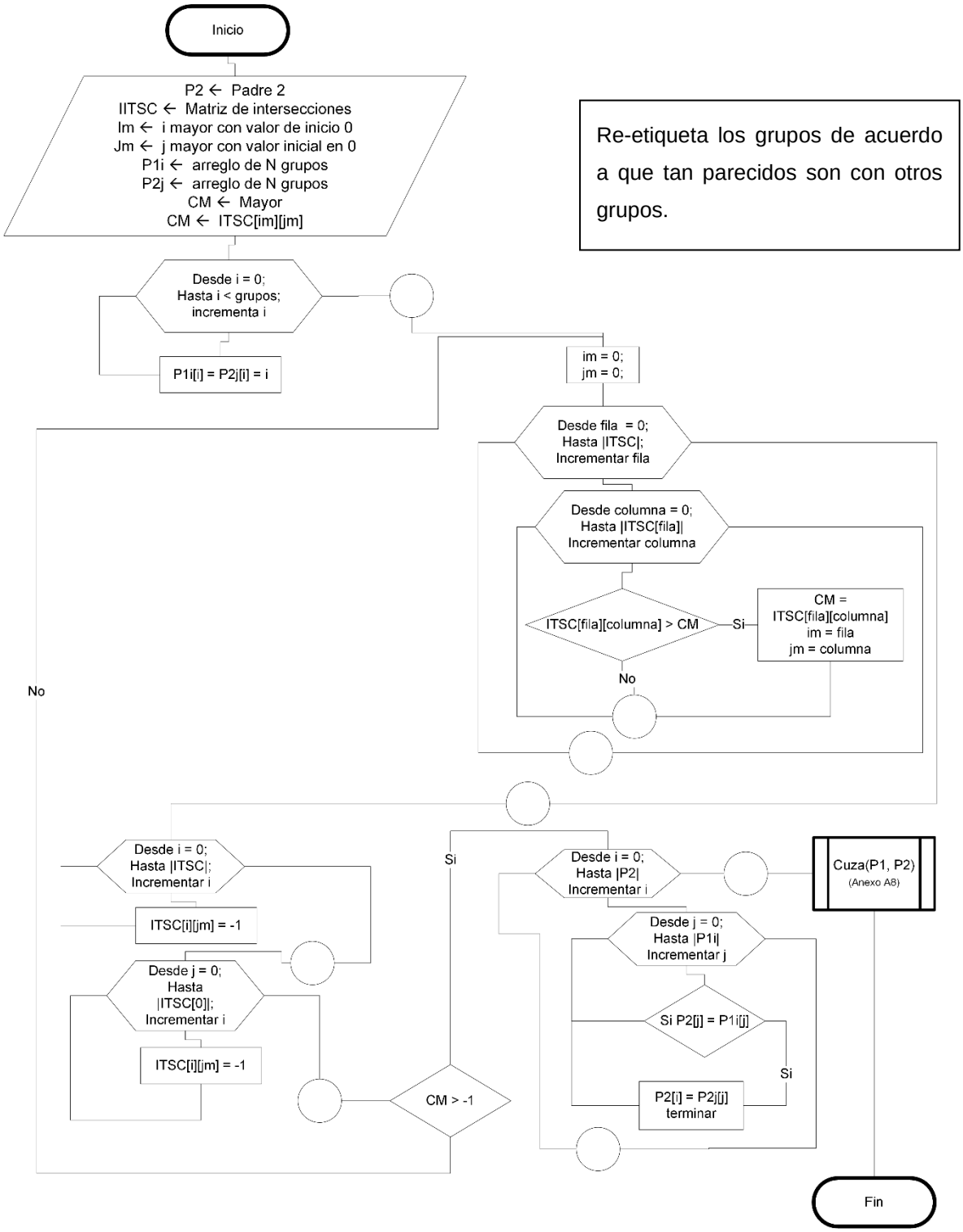
Este método solo genera una matriz con las posiciones en la que se encuentran los grupos de un menor a mayor y por defecto está llena de -1s.

Anexo A9: Diagrama de flujo de Intercambio de Grupos

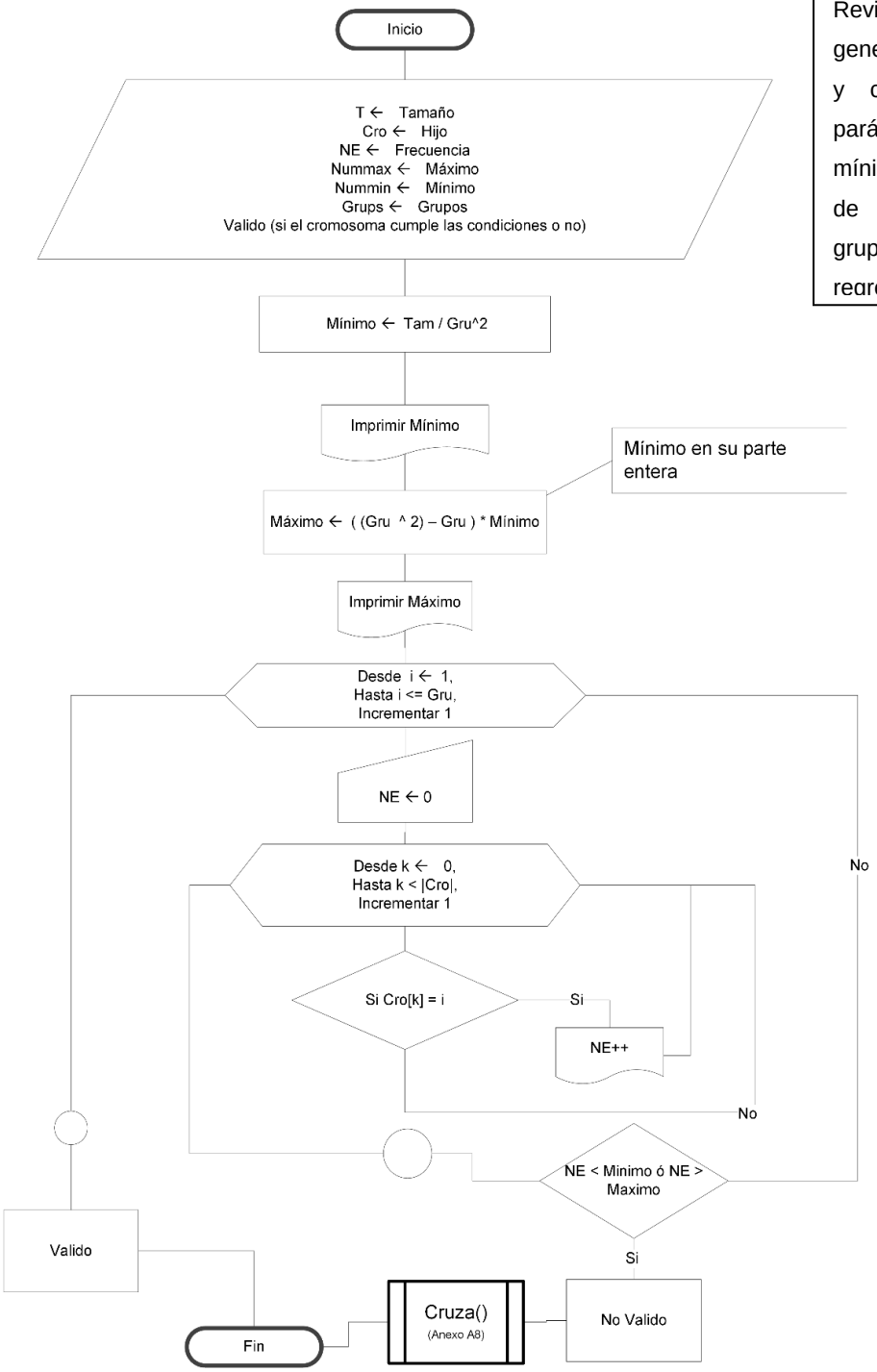


El método necesita a PA y la Matriz del PB. Para ver que tanto se parece el grupo 1 del padre 1 con el grupo 1 del padre 2 y así para cada uno de los grupos de los padres. Como resultado obtendremos una nueva matriz con los valores que se interceptan entre cada grupo.

Anexo A10: Diagrama de flujo de Re-etiquetación

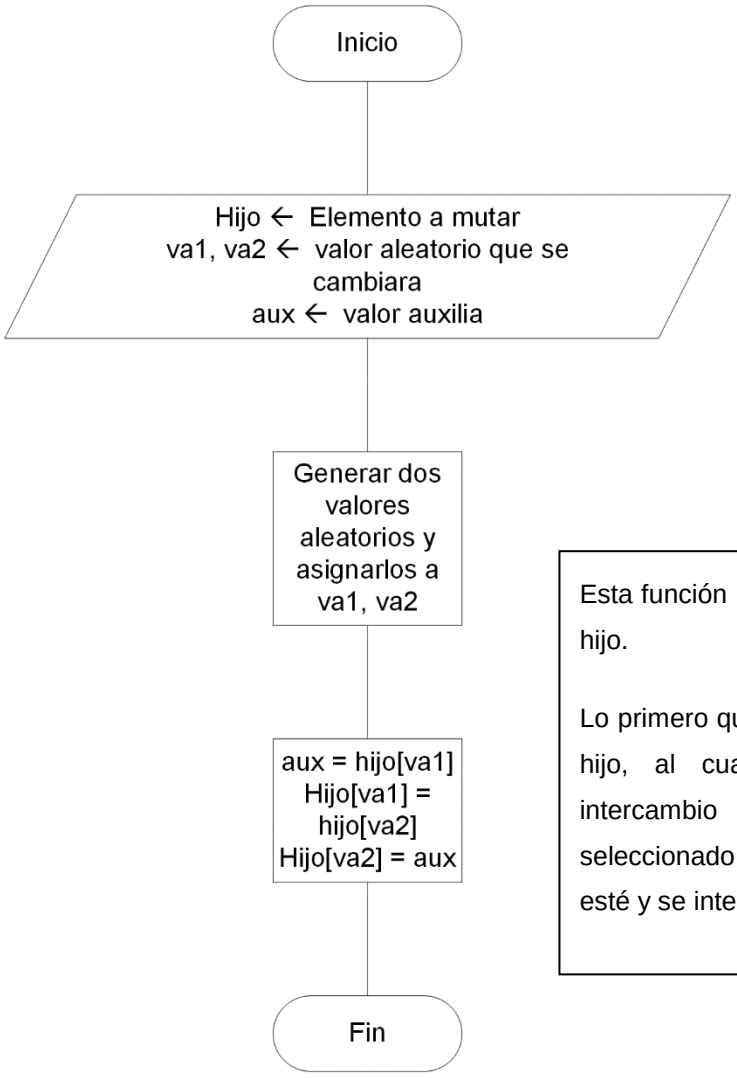


Anexo A11: Diagrama de flujo de la Validación



Revisa si los individuos generados son válidos y cumplen con los parámetros para un mínimo y un máximo de elementos por grupo de no ser así regresa a generar uno

Anexo A12: Diagrama de flujo del Operador de Mutación por Intercambio



Esta función realiza la mutación de un hijo.

Lo primero que hace es selecciona un hijo, al cual se le realizara un intercambio de genes, se seleccionado dos puntos al azar de esté y se intercambian los valores.

Anexo B1: Base de Datos sintética

	sepal_ length	sepal_ width	petal_ length	petal_ width	class
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5	3	1.6	0.2	Iris-setosa
27	5	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa
29	5.2	3.4	1.4	0.2	Iris-setosa
30	4.7	3.2	1.6	0.2	Iris-setosa
31	4.8	3.1	1.6	0.2	Iris-setosa
32	5.4	3.4	1.5	0.4	Iris-setosa
33	5.2	4.1	1.5	0.1	Iris-setosa
34	5.5	4.2	1.4	0.2	Iris-setosa
35	4.9	3.1	1.5	0.1	Iris-setosa
36	5	3.2	1.2	0.2	Iris-setosa
37	5.5	3.5	1.3	0.2	Iris-setosa
38	4.9	3.1	1.5	0.1	Iris-setosa
39	4.4	3	1.3	0.2	Iris-setosa
40	5.1	3.4	1.5	0.2	Iris-setosa
41	5	3.5	1.3	0.3	Iris-setosa
42	4.5	2.3	1.3	0.3	Iris-setosa
43	4.4	3.2	1.3	0.2	Iris-setosa
44	5	3.5	1.6	0.6	Iris-setosa
45	5.1	3.8	1.9	0.4	Iris-setosa
46	4.8	3	1.4	0.3	Iris-setosa
47	5.1	3.8	1.6	0.2	Iris-setosa
48	4.6	3.2	1.4	0.2	Iris-setosa
49	5.3	3.7	1.5	0.2	Iris-setosa
50	5	3.3	1.4	0.2	Iris-setosa
51	7	3.2	4.7	1.4	Iris-versicolor
52	6.4	3.2	4.5	1.5	Iris-versicolor
53	6.9	3.1	4.9	1.5	Iris-versicolor
54	5.5	2.3	4	1.3	Iris-versicolor
55	6.5	2.8	4.6	1.5	Iris-versicolor
56	5.7	2.8	4.5	1.3	Iris-versicolor
57	6.3	3.3	4.7	1.6	Iris-versicolor
58	4.9	2.4	3.3	1	Iris-versicolor
59	6.6	2.9	4.6	1.3	Iris-versicolor
60	5.2	2.7	3.9	1.4	Iris-versicolor
61	5	2	3.5	1	Iris-versicolor
62	5.9	3	4.2	1.5	Iris-versicolor
63	6	2.2	4	1	Iris-versicolor
64	6.1	2.9	4.7	1.4	Iris-versicolor
65	5.6	2.9	3.6	1.3	Iris-versicolor
66	6.7	3.1	4.4	1.4	Iris-versicolor
67	5.6	3	4.5	1.5	Iris-versicolor
68	5.8	2.7	4.1	1	Iris-versicolor
69	6.2	2.2	4.5	1.5	Iris-versicolor
70	5.6	2.5	3.9	1.1	Iris-versicolor
71	5.9	3.2	4.8	1.8	Iris-versicolor
72	6.1	2.8	4	1.3	Iris-versicolor
73	6.3	2.5	4.9	1.5	Iris-versicolor
74	6.1	2.8	4.7	1.2	Iris-versicolor
75	6.4	2.9	4.3	1.3	Iris-versicolor

76	6.6	3	4.4	1.4	Iris-versicolor
77	6.8	2.8	4.8	1.4	Iris-versicolor
78	6.7	3	5	1.7	Iris-versicolor
79	6	2.9	4.5	1.5	Iris-versicolor
80	5.7	2.6	3.5	1	Iris-versicolor
81	5.5	2.4	3.8	1.1	Iris-versicolor
82	5.5	2.4	3.7	1	Iris-versicolor
83	5.8	2.7	3.9	1.2	Iris-versicolor
84	6	2.7	5.1	1.6	Iris-versicolor
85	5.4	3	4.5	1.5	Iris-versicolor
86	6	3.4	4.5	1.6	Iris-versicolor
87	6.7	3.1	4.7	1.5	Iris-versicolor
88	6.3	2.3	4.4	1.3	Iris-versicolor
89	5.6	3	4.1	1.3	Iris-versicolor
90	5.5	2.5	4	1.3	Iris-versicolor
91	5.5	2.6	4.4	1.2	Iris-versicolor
92	6.1	3	4.6	1.4	Iris-versicolor
93	5.8	2.6	4	1.2	Iris-versicolor
94	5	2.3	3.3	1	Iris-versicolor
95	5.6	2.7	4.2	1.3	Iris-versicolor
96	5.7	3	4.2	1.2	Iris-versicolor
97	5.7	2.9	4.2	1.3	Iris-versicolor
98	6.2	2.9	4.3	1.3	Iris-versicolor
99	5.1	2.5	3	1.1	Iris-versicolor
100	5.7	2.8	4.1	1.3	Iris-versicolor
101	6.3	3.3	6	2.5	Iris-virginica
102	5.8	2.7	5.1	1.9	Iris-virginica
103	7.1	3	5.9	2.1	Iris-virginica
104	6.3	2.9	5.6	1.8	Iris-virginica
105	6.5	3	5.8	2.2	Iris-virginica
106	7.6	3	6.6	2.1	Iris-virginica
107	4.9	2.5	4.5	1.7	Iris-virginica
108	7.3	2.9	6.3	1.8	Iris-virginica
109	6.7	2.5	5.8	1.8	Iris-virginica
110	7.2	3.6	6.1	2.5	Iris-virginica
111	6.5	3.2	5.1	2	Iris-virginica
112	6.4	2.7	5.3	1.9	Iris-virginica
113	6.8	3	5.5	2.1	Iris-virginica
114	5.7	2.5	5	2	Iris-virginica
115	5.8	2.8	5.1	2.4	Iris-virginica
116	6.4	3.2	5.3	2.3	Iris-virginica
117	6.5	3	5.5	1.8	Iris-virginica
118	7.7	3.8	6.7	2.2	Iris-virginica
119	7.7	2.6	6.9	2.3	Iris-virginica
120	6	2.2	5	1.5	Iris-virginica
121	6.9	3.2	5.7	2.3	Iris-virginica
122	5.6	2.8	4.9	2	Iris-virginica
123	7.7	2.8	6.7	2	Iris-virginica
124	6.3	2.7	4.9	1.8	Iris-virginica
125	6.7	3.3	5.7	2.1	Iris-virginica
126	7.2	3.2	6	1.8	Iris-virginica
127	6.2	2.8	4.8	1.8	Iris-virginica
128	6.1	3	4.9	1.8	Iris-virginica
129	6.4	2.8	5.6	2.1	Iris-virginica
130	7.2	3	5.8	1.6	Iris-virginica
131	7.4	2.8	6.1	1.9	Iris-virginica
132	7.9	3.8	6.4	2	Iris-virginica
133	6.4	2.8	5.6	2.2	Iris-virginica
134	6.3	2.8	5.1	1.5	Iris-virginica
135	6.1	2.6	5.6	1.4	Iris-virginica
136	7.7	3	6.1	2.3	Iris-virginica
137	6.3	3.4	5.6	2.4	Iris-virginica
138	6.4	3.1	5.5	1.8	Iris-virginica
139	6	3	4.8	1.8	Iris-virginica
140	6.9	3.1	5.4	2.1	Iris-virginica
141	6.7	3.1	5.6	2.4	Iris-virginica
142	6.9	3.1	5.1	2.3	Iris-virginica
143	5.8	2.7	5.1	1.9	Iris-virginica
144	6.8	3.2	5.9	2.3	Iris-virginica
145	6.7	3.3	5.7	2.5	Iris-virginica
146	6.7	3	5.2	2.3	Iris-virginica
147	6.3	2.5	5	1.9	Iris-virginica
148	6.5	3	5.2	2	Iris-virginica
149	6.2	3.4	5.4	2.3	Iris-virginica
150	5.9	3	5.1	1.8	Iris-virginica

Anexo B2: Base de Datos sintética

altura	ancho	peso	diámetro	Clase
49	52	100	26	Manzana
54	47	100	23.5	Manzana
60	59	145	29.5	Manzana
60	63	165	31.5	Manzana
54	56	120	28	Manzana
60	55	120	27.5	Manzana
55	56	115	28	Manzana
55	56	110	28	Manzana
55	55	105	27.5	Manzana
56	57	105	28.5	Manzana
151	148	2500	74	Melón
145	140	2000	70	Melón
180	162	3000	81	Melón
140	142	1500	71	Melón
144	143	1500	71.5	Melón
120	122	1000	61	Melón
121	123	1000	61.5	Melón
145	143	1500	71.5	Melón
143	143	1500	71.5	Melón
146	140	2000	70	Melón

REFERENCIAS

Aja, S., 2005. *Reconocimiento de Patrones.* México: UNAM.

Alba, J. L. & Cid, J., 2006. *Reconocimiento de Patrones.* [En línea] Available at: http://www.gts.tsc.uvigo.es/pi/Reconocimiento.pdf

Alonso Romero, D. L. & Calonge Cano, D. T., 2008. *Redes Neuronales y Reconocimiento de Patrones.* Valladolid: Dpto. de Informática y Automática.

Alonso, J. I., Gómez, J. A., García, I. & Martínez, J., 2007. *Autolocalización inicial para robots móviles usando el método de K-NN.* Albacete: Articulo.

Alvarado, P. A., 2010. *Algoritmos de Clasificación: Comparación del Algoritmo Naive Bayes con otras Metodologías para la Clasificación de Correo Electrónico no deseado.* Loja: Articulo.

Ana, F., 2002. Similarity Measure and Clustering of String Patterns. En: *Pattern Recognition and String Matching.* Wisconsin: Kluwer Academic Publishers, pp. 155-193.

Arranz, J. & Parra, A., 2007. *Algoritmos Genéticos.* Madrid: Practicas de Asignación.

Bedoya, J. A., 2011. *Aplicación de distancias entre terminos para datos planos y jerárquicos.* Valencia: Tesis.

Bedregal, C. E., 2008. *Agrupamiento de Datos utilizando técnicas MAM-SOM.* s.l.:s.n.

Berzal, F., 1999. *Metodos de agrupamiento.* s.l.:s.n.

Bokan, A., Patiño, R. & Túpac, Y., 2011. *Validación de Clusters usando IEKA y SL-SOM.* San Paulo(Peru): s.n.

Carrasco, J. A. & Martínez, J. F., 2011. Reconocimiento de Patrones. *Komputer Sapiens,* pp. 5-9.

Cervigón, L. A. y. C., 2009. *Algoritmos Evolutivos.* Madrid España: RA-MA.

Chapelle, O., Schölkopf , B. & Zien, A., 2006. *Semi-Supervised Learning.* London: The MIT Press.

Corso, C. L., 2009. *Aplicación de algoritmos de clasificacion supervisada usando weka.* Argentina(Córdova): Universidad Tecnológica Nacional, Facultad Regional Córdoba.

Cortijo, F. J., 2001. *Tecnicas no supervisadas Métodos de agrupamiento.* s.l.:s.n.

Davies, D. L. & Bouldin, D. W., 1979. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions,* Abril, 2(PAMI-1), pp. 224-227.

De la O, J. R., 2007. *Interfaz Cerebro-Computadora para el control de un cursor Basado en Ondas Cerebrales.* México: s.n.

Desgraupes, B., 2013. *Clustering Indices.* Paris: s.n.

Díaz, C., 2007. *Clasificacion no Supervisada.* [En línea] Available at: http://clustering.50webs.com/supervisadovsnosupervisado.html [Último acceso: 6 Diciembre 2013].

Díaz, J. C., 2010. *Un algoritmo Genético con codificación real para la evolucion de Trasformaciones Lineales.* Laganes: s.n.

Duda, R. D., Stork, D. G. & Hart, P. E., 2000. *Pattern Classification.* Second ed. California: Wiley.

Dunn, J. C., 1974. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics,* Issue 4, pp. 95-104.

Funes, A., 2008. *Agrupamiento Conceptual Jerárquico Basado en Distancias.* Valencia: s.n.

Gadania, A., Daniel, L., Roney, B. & Wu, E., 2006. *Implementation of and Experimenting with a clustering tool.* s.l.:s.n.

García, C. & Gómez, I., 2009. *Algoritmos de aprendizaje KNN & KMEANS.* Madril: s.n.

García, G., 2012. *Descubrimiento de factores que inciden en la eficiencia terminal de estudiantes de educación superior con arboles de desición.* Texcoco(México): s.n.

Garre, M. y otros, 2007. *Comparacion de diferentes algoritmos de clustering en la estimacion de costes en el desarrollo de software.* Madrid: s.n.

Gestal, M., 2010. *Introduccion a los Algoritmos Geneticos.* [En línea] Available at: http://sabia.tic.udc.es/mgestal/cv/index.php [Último acceso: 02 Julio 2014].

Gómez, F. J., Fernández, M. Á., López, M. T. & Díaz, M. A., 1994. Aprendizaje con Redes Neuronales Artificiales. En: *Revista Ensayos.* s.l.:UCLM, pp. 169-180.

Gutierrez, A. E., García, M. & Martínez, J. F., 2012. *Algoritmo de agrupamiento basado en patrones utilizado árboles de desición no supervisados.* Puebla: s.n.

Haykin, S., 1998. *Neural Networks A Comprehensive foundation.* Segunda ed. Michigan: Pretice Hall.

Hernández, E., 2006. *Algoritmo de clustering basado en entropía para descubrir grupos en atributos de tipo mixto.* México: s.n.

Igel, C., 2002. *Machine Learning: Kernel-based Methods.* Copenhagen: s.n.

Ingaramo, D. A., Errecalde, M. I. & Rosso, P., 2007. Medidas internas y externas en el agrupamiento de resumenes científicos de dominio reducidos. *Procesamiento del Lenguaje Natural,* pp. 55-62.

Jiménez, D. D., 2011. *Algoritmo de clustering paralelos en sistemas de recuperacion de informacion distribuidos.* Valencia: s.n.

Kittler, J., 2002. *Notas del Seminario de Reconocimiento de Patrones.* s.l.:s.n.

Koza, J. R., 1992. *Genetic Programming. On the Programming of Computers by Means of Natural Selection..* s.l.:The MIT Press.

Kumar, V. y otros, 2008. Top 10 algorithms in data mining. *Knowledge Information Systems 14,* pp. 1-37.

Kunzmann, K., 2005. *Reconocimiento de Patrones en Imagenes y Videos Endocospicos Utilizando Redes Neuronales Artificiales.* s.l.:s.n.

Kuri, A. F. & Galaviz, J., 2007. *Algoritmos Genéticos.* México: SMOIA.

Llobet, R., 2006. *Aportaciones al Diagnostico de Cáncer Asistido por Ordenador.* Valencia: s.n.

López, J. C., 2010. *Un Algoritmo Genético con codifcación para la Evolución de Trasformaciones Lineales.* Madrid: s.n.

Malagón, C., 2003. *Clasificadores Bayesianos. El algoritmo Naïve Bayes.* s.l.:s.n.

Marin, A. & Branch, J. W., 2008. *Aplicacion de dos nuevos algortimos para agrupar resultados de búsquedas en sistemas de catálogos públicos.* Colombia: s.n.

Marshall, M., 2007. *Machine Lerning Repository.* [En línea] Available at: https://archive.ics.uci.edu/ml/datasets/Iris [Último acceso: 7 Abril 2014].

Martínez, C. D., 2000. *La cadena media y su aplición en Reconocimiento de Formas.* s.l.:s.n.

Milone, D. y otros, 2009. Métodos de agrupamiento no supervisado para la integracion de datos genómicos y metabólicos de múltiples líneas de introgresión. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial,* 13(44), pp. 56-66.

Ming-Hseng Tseng, Chang-Yun, C., Ping-Hung, T. & Hui-Ching, W., 2010. *A Study on cluster validity using intelligent evolutionary k-means approach.* s.l.:s.n.

Mora, J., Morales, G. & Barrera, R., 2008. *Evaluación del clasificador basado en los k vecinos más cercanos para la localización de la zona en falla en los sistemas de potencia.* Colombia: s.n.

Morales, E. F., 2012. *Aprendizaje Basado en Instancias.* [En línea] Available at: http://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/node69.html [Último acceso: noviembre 2013].

Morales, G., Mora, J. & Vargas, H., 2008. Estrategia de regresión basada en el método de los k vecinos más cercanos para la estimación de distancia de falla en sistemas radiales. *Revista Facultad de Ingeniería Universidad de Antioquia,* pp. 100-108.

Moreno, A. y otros, 1994. *Aprendizaje Automatico.* Primera ed. Bacelona: Edicions UPC.

Moreno, F., 2004. *Clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más cercano.* Alicante(Alicante): s.n.

Moreno, J., Rivera, J. C. & Ceballos, Y. F., 2010. *Agrupamiento Homogéneo de elementos con múltiples atributos meiante algoritmos genéticos.* Medellin: Redalyc.

Moujahid, A., Inza, I. & Larrañaga, P., 2008. *Clsificadores K-NN.* Pais-Vasco: s.n.

Osorio, N., 2013. *Identificación de Factores que Influyen en la Desercion Escolar de Estudiantes Universitarios Usando Algoritmos Genéticos y K Vecinos más Cercanos.* Tianguistenco(México): s.n.

Pajares, G. & Santos, M., 2006. *Inteligencia Artificial e Ingenieria del Conocimiento.* Madrid: RaMa.

Pascual, D., Pla, F. & Sánchez, S., 2007. *Algoritmos de Agrupamiento.* Santiago de Cuba: s.n.

Pascual, D., Vázquez, F. D., Sánchez, S. & Pla, F., 2014. *Detección de ruido en aprendizaje semi-supervisado con el uso de flujo de datos.* Santiago: Revista Facultad de Ingeniería Universidad de Antioquia.

Porta Zamorano, J., 2005. Clasificación de patrones:Metodos No Supervisados. Issue IULA-UPF, pp. 1-10.

Pozas, J. C. & Vázquez, N., 2007. *Algoritmos Genéticos Aplicación al Juego de las N Reinas.* Madril: s.n.

Primitivo, L., 2011. *Analisis de la complejidad de los datos y su efecto en las redes neuro artificiales.* Texcoco(México): s.n.

Refaelzadeh, P., Tang, L. & Liu, H., 2008. *Cross Validation.* Arizona: s.n.

Rodríguez, J. E., Barrera, H. A. & Bautista, S. P., 2010. *Software para el filtrado de paginas web pornograficas basado en clasificador KNN - UDWEBPORN.* Medellín: s.n.

Romo R., H. A., Ramírez M., F. & B, V., 2007. *Detección del Bacilo Mycobacterium Tuberculosis mediante Reconocimiento de Patrones.* s.l.:s.n.

Russell, S. & Norvig, P., 1996. *Inteligencia Artificial un Enfoque moderno.* s.l.:Prentice Hall.

Sabau, A. S., 2012. *Variable Density Based Genetic Clustering.* Pitesti: s.n.

Salas, R. E., 2010. *Algoritmo genético para la solucion del problema de optimización combinatoria y desición secuencial en el juego "But who´s couting".* Bogota: s.n.

Sanjinez, V. H., 2011. *Optimizacion del costo del enlaces en una red.* Pamplona: s.n.

Shinn-Ying Ho, L.-S. S. ,. J.-H. C., 2009. Intelligent evolutionary algorithms for large parameter optimization problems. *IEEE Transactions Evolutionary Computation,* Volumen 8, pp. 522-541.

Talavera, I. & Rodríguez, J. L., 2008. *Reporte tecnico.* La habana: CENATAV.

Theodoridis, S. & Koutroumbas, K., 2003. *Pattern Recognition second edition.* Greece: Elsevier.

Vazquez, F., 2008. *Caracterización e Interpretación de Descripciones Conceptuales en Dominios poco Estructurados.* México: s.n.

Whitley Darrell, 1993. *A Genetic Algorithm Tutorial.* Colorado: Department of Computer Science.

Witten, I., Frank, E. & Hall, M., 2011. *Data Mining Practical Machine Learning Tools and Techniques.* United States: Morgan Kaufmann Publications.

Yañez, D. C., 2008. *Reconocimiento de patrones.* [En línea] Available at: http://148.204.64.169/cornelio/RecPat.htm

Yolis, E., 2003. *Algoritmos Géneticos Aplicados a la Categorización Automática de Documentos.* Buenos Aires: s.n.