

SEP

TECNM

INSTITUTO TECNOLOGICO DE TIJUANA
DIVISIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN



ADAPTACIÓN DIFUSA DE PARAMETROS EN
ALGORITMO DE EFIMERAS Y SU APLICACIÓN A LA
OPTIMIZACIÓN DE REDES NEURONALES.

TRABAJO DE TESIS

Presentado por

ARCENIO ENRIQUE LIZÁRRAGA HERNÁNDEZ

Para obtener el Grado de

MAESTRO EN CIENCIAS EN CIENCIAS DE LA
COMPUTACION

Director de Tesis

DR. FEVRIER VALDEZ ACOSTA
Co-Director de Tesis

DRA. OSCAR CASTILLO LOPEZ

TIJUANA, BC., MARZO 2022



Tijuana, B.C., 10 de Febrero del 2022
Asunto: Se autoriza impresión de Trabajo de Tesis

C. M.C. Maribel Guerrero Luis
Jefe del Depto. de Servicios Escolares
Presente.

En lo referente al trabajo de tesis escrito, con título "**ADAPTACIÓN DIFUSA DE PARAMETROS EN ALGORITMO DE EFIMERAS Y SU APLICACIÓN A LA OPTIMIZACIÓN DE REDES NEURONALES**", presentado por el **C. ARCENIO ENRIQUE LIZARRAGA HERNANDEZ** alumno de la Maestría en Ciencias de la Computación con número de control **M20210002**, informamos a usted que después de una minuciosa revisión, de acuerdo con lo establecido en el reglamento vigente para este caso, nuestro dictamen es: Se aprueba en todas sus partes, en virtud de reunir los requisitos de un trabajo de grado de maestría y a la vez se autoriza al interesado para que proceda de inmediato a la impresión del mismo.

ATENTAMENTE

DR. FEVRIER ADOLFO VALDEZ ACOSTA
PRESIDENTE

DR. OSCAR CASTILLO LOPEZ
SECRETARIO

DRA. ELBA PATRICIA MELIN OLMEDA
VOCAL

c.c.p. Oficina de Titulación
c.c.p. División de Estudios de Posgrado e Investigación
c.c.p. Expediente
c.c.p. Interesado

EPMO/*inf





Instituto Tecnológico de Tijuana

Tijuana, Baja California,

03/marzo/2022

OFICIO No. 024/DEPI/2022

Asunto: Autorización de Impresión de Tesis

MARIBEL GUERRERO LUIS
JEFA DEL DEPARTAMENTO DE SERVICIOS ESCOLARES
PRESENTE

En lo referente al trabajo de tesis, "Adaptación difusa de parámetros en algoritmo de efímeras y su aplicación a la optimización de redes neuronales". Presentado por C. **Arcenio Enrique Lizarraga Hernández**, alumno de la Maestría en Ciencias de la Computación con numero de control **M20210002**; informo a usted que a solicitud del comité de tutorial, tengo a bien **Autorizar la impresión de Tesis**, atendiendo las disposiciones de los Lineamientos para la Operación de Estudios de Posgrado del Tecnológico Nacional de México.

Sin más por el momento le envió un cordial saludo.

A T E N T A M E N T E

Excelencia en Educación Tecnológica



GUADALUPE HERNÁNDEZ ESCOBEDO
JEFE DE DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

ccp. Archivo

GHE/lap



Calzada del Tecnológico S/N Esq. Castillo de Chapultepec y calle Cuauhtemotzin,
Fracc. Tomás Aquino C.P.22414 Tijuana, Baja California. Tel. 01 (664) 6078400
Est. 101

e-mail: dir.tijuana@tecnm.mx | tecnm.mx | Tijuana.tecnm.mx



2022 *Ricardo Flores*
Año de Magón
PRECURSOR DE LA REVOLUCIÓN MEXICANA

CARTA DECLARACIÓN DE PROPIEDAD INTELECTUAL

Tijuana, BC 28 de febrero del 2022

Yo (**Arcenio Enrique Lizárraga Hernández**) reconozco que el Trabajo de Tesis de Maestría que realice durante mis estudios en la Maestría en Ciencias de la Computación del Instituto Tecnológico de Tijuana fue parte del Proyecto de Investigación Titulado: (**Adaptación difusa de parámetros en algoritmo de efimeras y su aplicación a la optimización de redes neuronales**) que desarrolla mi director de tesis el **Dr. (Fevrier Valdez Acosta)**-y del cual es responsable del proyecto de investigación. Por esta razón, los métodos, modelos, algoritmos, y software realizados, así como datos y resultados obtenidos durante el desarrollo de mi tesis de maestría son propiedad intelectual de mi Director de Tesis, del Tecnológico Nacional de México, Instituto Tecnológico de Tijuana y del Conacyt, y No podré utilizarlos por mi cuenta durante, Ni después de terminar mi beca o estudios, excepto a solicitud escrita para poder utilizarlos bajo una colaboración directa con mi director el cual es responsable del proyecto de investigación. Por tanto, estoy de acuerdo en que No podre utilizar ni tomar modelos, ni datos utilizados en este proyecto de investigación y en el desarrollo de tesis para: presentaciones, publicaciones ni desarrollo de mi propia investigación que pudiera desarrollar una vez concluidos mis estudios.

Atentamente

Arcenio Enrique Lizarraga Hernández,



Estudiante de la Maestría en Ciencias de la Computación

DECLARACIÓN DE ORIGINALIDAD

Tijuana, BC., 28 de febrero de 2022,

Yo, (Arcenio Enrique Lizarraga Hernnandez), estudiante de la Maestría en Ciencias de la Computación, en mi calidad de autor manifiesto que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patente y similaridad. Por lo tanto, la obra realizada es de mi exclusiva autoría y no infringí en copiar el texto o imágenes, de fuentes de información por lo cual soy responsable del escrito que aquí se presenta.

Así mismo, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

En caso de presentarse cualquier reclamación o acción por parte de terceros en cuanto a los derechos de autor sobre la obra en cuestión, acepto toda la responsabilidad de tal infracción y relevo de esta a mi director de tesis, así como al Tecnológico Nacional de México, al Instituto Tecnológico de Tijuana y a sus respectivas autoridades.

Arcenio Enrique Lizarraga Hernández,



Nombre completo del estudiante y firma autografa
Estudiante de la Maestría en Ciencias de la Computación

Resumen

El presente proyecto de tesis, consistió en una investigación acerca del algoritmo de Mayfly el cual fue Inspirado en el comportamiento de vuelo y el proceso de apareamiento de efímeras, el algoritmo de Mayfly combina las principales ventajas de la inteligencia de enjambre y los algoritmos evolutivos, dando como resultado un mejor rendimiento del algoritmo PSO, por lo que analizaremos los parámetros con funciones benchmark con el fin de encontrar un rango adecuado en el que se puedan aplicar algunas reglas difusas, con el fin de mejorar la velocidad de convergencia del algoritmo y luego ser aplicado a problemas de optimización de redes neuronales debido a que los algoritmos para encontrar soluciones como PSO, DE , GA y FA han demostrado ser eficientes en términos de convergencia para este tipo de problemas, obtuvimos como resultado que los parámetros a_1 y a_2 afectan más el rendimiento del algoritmo ,variando el parámetro a_1 se observó que el parámetro alcanza su mejor desempeño en valores cercanos a 1 y el parámetro a_2 en valores entre 3 y 5, finalmente el adaptador difuso mejoro el desempeño del algoritmo de Mayfly al utilizar 9 reglas difusas y funciones de membresía trapezoidales, este adaptador difuso tendrá como entrada la iteración y la diversidad , y como salida los parámetros a_1 y a_2 .

Abstract

The present thesis project consisted of an investigation about the Mayfly algorithm which was inspired by the flight behavior and the mating process of mayflies, the Mayfly algorithm combines the main advantages of swarm intelligence and evolutionary algorithms, resulting in a better performance of the PSO algorithm, so we will analyze the parameters with reference functions in order to find a suitable range in which some fuzzy rules can be applied, in order to improve the convergence speed of the algorithm and then be applied to neural network optimization problems because the algorithms to find solutions such as PSO, DE, GA and FA have proven to be efficient in terms of convergence for these types of problems, we obtained as a result that the parameters a_1 and a_2 of the performance of the algorithm, by varying the parameter a_1 it will be executed that the parameter reaches its best performance in values ranging from 1 and the parameter a_2 in values between 3 and 5, finally the fuzzy adapter improved the performance of the Mayfly algorithm by using 9 fuzzy rules and trapezoidal membership functions, this fuzzy adapter will have iteration and diversity as input, and as output parameters a_1 and a_2 .

Dedicatoria

Esta tesis está dedicada principalmente a mi familia, a mis padres a quienes les agradezco por su apoyo, trabajo y esfuerzo a lo largo de toda mi vida a mi hija por sus palabras y mirada de aliento, a mi novia por su apoyo a lo largo de mi estancia en la maestría, por lo tanto, este trabajo es dedicado con mucho amor y cariño a ustedes.

Gracias.

Agradecimientos

Primeramente, agradeceré a mi familia por estar conmigo a lo largo de mi vida y sobre todo de mi preparación académica, por darme valor y fuerza para seguir adelante con mis sueños y no rendirme jamás.

A mi hija Michelle Nicole por darme fuerza y coraje para alcanzar mis metas, por darme su compañía y ánimos cuando realizaba mis tareas escolares, por todas las sonrisas que me motivaron a ser mejor cada día para darle un mejor futuro.

A mis padres Arcenio Lizárraga y Blanca Hernández ya que ustedes me convirtieron en la persona que soy ahora, se los agradezco de todo corazón, gracias por darme la oportunidad de estudiar y tener un mejor futuro, por su apoyo, cariño y comprensión que me han brindado desde que tengo uso de razón.

A Jocelyn por brindarme la motivación para seguir adelante con mi preparación académica y por ser un gran apoyo desde el momento en que entro a mi vida, por ayudarme a ser una mejor persona y seguir creciendo cada día.

A mi director de tesis, Dr. Fevrier Valdez Acosta por confiar en mí y por ayudarme en mi desarrollo de la tesis siendo mi guía y mentor.

A mi codirector de tesis, Dr. Oscar Castillo López por guiarme en el proceso de desarrollo de la tesis y revisar a detalle cada uno de mis avances presentados.

A la Dra. Melin por guiarme en mis tramites académicos desde que entre en el postgrado y además aconsejarme en todo lo posible durante mi desarrollo de tesis.

Agradezco al CONACYT por el apoyo económico otorgado a través de la beca durante el periodo febrero 2020 a enero 2022 para poder concluir exitosamente el primer semestre de maestría en ciencias de la computación.

Índice

Resumen.....	2
Abstract.....	3
Dedicatoria.....	4
Agradecimientos.....	5
Índice.....	6
Índice de figuras.....	8
Índice de tablas.....	10
Índice de ecuaciones.....	12
1. Introducción	13
2. Marco teórico	15
2.1 Representación de un algoritmo genético.....	15
2.1.2 Cálculo del fitness	16
2.1.3 Métodos de cruce	16
2.1.3.1 Método de cruce de un punto	17
2.1.3.2 Método de cruce de dos puntos	18
2.1.3.3 Método de cruce uniforme	18
2.1.4.1 Selección por ruleta	19
2.1.4.2 Selección elitista	20
2.1.4.3 Selección por torneo	20
2.1.4.4 Selección escalada	20
2.1.4.4 Selección jerárquica	21
2.2 Método de Mayfly.....	22
2.2.1 Movimiento de efímeras machos	22
2.2.2 Movimiento de efímeras hembras	23
2.3 Algoritmo OBL-MO.....	24
2.4 Algoritmo de Mayfly modificado.....	24
2.5 Algoritmo NMO.....	25
2.6 El algoritmo de la mosca de mayo BMA.....	26
2.7 Redes neuronales.....	27
2.7.1 Representación de una red neuronal	27
2.8 Serie de tiempo Mackey Glass.....	28

2.9 Lógica difusa.....	28
2.9.1 Reglas difusas.....	28
2.10 Hipótesis.....	29
3. Planteamiento del problema.....	29
3.1 Definición de puntos en el problema.	30
3.2 Resolución de la problemática principal.....	30
4. Desarrollo y Resultados.....	31
4.1 Estudio del impacto de parámetros.....	31
4.1.1 Graficas del estudio de parámetros.....	36
4.2 Propuesta de adaptación difusa.....	39
4.3 Diseño del adaptador difuso.....	40
4.3.1 Adaptador difuso de 1 entrada y 1 salida.....	41
4.3.2 Adaptador difuso de 2 entradas y 2 salidas.....	45
4.4 Prueba estadística.....	49
4.4.1 Prueba de hipótesis función de la esfera.....	50
4.4.2 Prueba de hipótesis función de rastrigin.....	51
4.4.3 Prueba de hipótesis función de Griewank.....	52
4.4.4 Prueba de hipótesis función de Powell.....	53
4.4.5 Prueba de hipótesis función de Sum Squares.....	54
4.4.6 Prueba de hipótesis resultados.....	55
4.5 Diseño de la red neuronal de Mackey Glass.....	56
4.6 Optimización de la red neuronal.....	58
5. Conclusiones.....	62
Referencias.....	63
Anexo 1.- Pseudocódigo de Mayfly.....	68
Anexo 2.- Pseudocódigo de Mayfly con adaptador difuso de parámetros.....	69
Anexo 3.- Pseudocódigo del cálculo de la diversidad.....	70
Anexo 4.- Generación de la red neuronal para Mackey Glass.....	71
Anexo 5 Datos utilizados para el entrenamiento de la red neuronal.....	73
Anexo 6.- Graficas de Mackey Glass.....	80
Anexo 7.- Adaptadores difusos.....	81

Índice de figuras.

Figura 2.1	Representación de un algoritmo genético	15
Figura 2.2	Determinación de aptitud de cada individuo.....	16
Figura 2.3	Cruce de un punto	17
Figura 2.4	Cruce de un punto codificación	17
Figura 2.5	Método de cruce de 2 puntos	18
Figura 2.6	Método de cruce uniforme	18
Figura 2.7	Selección por ruleta.....	19
Figura 2.8	Cromosoma mutado	21
Figura 2.9	Esquema General de hipótesis	29
Figura 3.1	Esquema general de planteamiento del problema.....	31
Figura 4.2	Impacto de parámetros función benchmark de la esfera.....	36
Figura 4.3	Impacto de parámetros función benchmark rastrigin.....	37
Figura 4.4	Impacto de parámetros función benchmark griewank	37
Figura 4.5	Impacto de parámetros función benchmark dejong1	38
Figura 4.6	Impacto de parámetros función benchmark wood	38
Figura 4.7	Diagrama de flujo del método original de Mayfly.....	39
Figura 4.8	Diagrama de flujo del método modificado de Mayfly	40
Figura 4.9	Reglas difusas	41
Figura 4.10	Adaptador difuso inicial.....	43
Figura 4.11	Funciones de membresía triangulares iniciales de entrada	43
Figura 4.12	Funciones de membresía triangulares iniciales de salida.....	43
Figura 4.13	Funciones de membresía triangular de entrada para el adaptador difuso 1	44
Figura 4.14	Funciones de membresía triangular de entrada para el adaptador difuso 1	44
Figura 4.15	Adaptador difuso final	46
Figura 4.16	Adaptador difuso final	46
Figura 4.17	Adaptador difuso final 2 variables de entrada y 2 de salida	48
Figura 4.18	Funciones de membresía de salida para el adaptador difuso final	48
Figura 4.19	Prueba cola izquierda para función benchmark de la esfera.....	50

Figura 4.20 Prueba cola izquierda para función benchmark de rastrigin	51
Figura 4.21 Prueba cola izquierda para función benchmark de griewank	52
Figura 4.22 Prueba cola izquierda para función benchmark de Powell	53
Figura 4.23 Prueba cola izquierda para función benchmark de Sum Squares	54
Figura 4.24 Serie de Mackey Glass	56
Figura 4.25 Arquitectura inicial de la red neuronal.....	57
Figura 4.26 Arquitectura de red neuronal	58
Figura 4.27 Optimización de la arquitectura de red neuronal	59
Figura 4.28 Configuración de la red neuronal.....	579
Figura 4.29 Red sin optimizar	60
Figura 4.30 Red optimizada	61
Figura 4.31 Comparación resultados de redes neuronales	61

Índice de tablas.

Tabla 4.1 Representación de las funciones benchmarks	31
Tabla 4.2 Evaluación de las funciones benchmark para 10 dimensiones variando el parámetro beta	32
Tabla 4.3 Evaluación de las funciones benchmark para 10 dimensiones variando el parámetro a_1	32
Tabla 4.4 Evaluación de las funciones benchmark para 10 dimensiones variando el parámetro a_2	33
Tabla 4.5 Evaluación de las funciones benchmark para 20 dimensiones variando el parámetro beta	33
Tabla 4.6 Evaluación de las funciones benchmark para 20 dimensiones variando el parámetro a_1	34
Tabla 4.7 Evaluación de las funciones benchmark para 20 dimensiones variando el parámetro a_2	34
Tabla 4.8 Evaluación de las funciones benchmark para 50 dimensiones variando el parámetro beta	35
Tabla 4.9 Evaluación de las funciones benchmark para 50 dimensiones variando el parámetro a_1	35
Tabla 4.10 Evaluación de las funciones benchmark para 50 dimensiones variando el parámetro a_2	36
Tabla 4.11 Funciones de membresía de entrada en el adaptador difuso inicial	42
Tabla 4.12 Funciones de membresía de salida en el adaptador difuso inicial	42
Tabla 4.13 Comparación de resultados con adaptador difuso para a_2 y a_1	45
Tabla 4.14 Funciones de membresía de entrada en el adaptador difuso de 2 entradas y 2 salidas	47
Tabla 4.15 Funciones de membresía de salida en el adaptador difuso de 2 entradas y 2 salidas.	47
Tabla 4.16 Resultados de los adaptadores difusos.....	49

Tabla 4.17 Resultados prueba de hipótesis.....55

Índice de ecuaciones

Ecuación 2.1	22
Ecuación 2. 2	23
Ecuación 2.3	23
Ecuación 2.4	23
Ecuación 2.5	24
Ecuación 2. 6	25
Ecuación 2.7	25
Ecuación 2.8	26
Ecuación 2. 9	27
Ecuación 2.10	29
Ecuación 4.1.....	41
Ecuación 4.2.....	47
Ecuación 4. 3	48
Ecuación 4.5	49

1. Introducción

La optimización es un proceso para encontrar la mejor solución de una función (ya sea su valor mínimo o máximo). Una gran cantidad de problemas del mundo real se representan como problemas de optimización, para la minimización de un solo objetivo.

Inspirado en el comportamiento de vuelo y el proceso de apareamiento de efímeras, el algoritmo propuesto combina las principales ventajas de la inteligencia de enjambre y algoritmos evolutivos. Para evaluar el desempeño del algoritmo se emplearon 10 funciones de benchmark [1].

Efímera es una especie de insectos del cual sus crías vivirían en el agua durante varios años y cuando estos maduren se descortezarán y volarán por el aire. Sin embargo, estos insectos maduros solo tienen el fin de reproducirse ya que su periodo de vida es corto [2].

De hecho, ofrece una potente estructura algorítmica híbrida, basada en el comportamiento de efímeras, para investigadores que intentan mejorar el rendimiento del algoritmo PSO utilizando técnicas tales como crossover y búsqueda local ya que se ha demostrado que PSO necesita algunas modificaciones [1].

La lógica difusa o lógica multivaluada se basa en la teoría de conjuntos difusos que nos ayuda a modelar el conocimiento mediante el uso de reglas difusas si-entonces las cuales proporcionan un cálculo sistemático para tratar la información lingüística y mejora el cálculo numérico mediante el uso de etiquetas lingüísticas estipuladas por funciones de pertenencia [17].

Los algoritmos de búsqueda de soluciones tales como PSO, DE, GA y FA han demostrado ser eficientes en términos de velocidad de convergencia para este tipo de problemas, por lo que esperamos que el algoritmo de Mayfly al ser una mejora del algoritmo PSO funcione de manera eficiente aplicado a problemas de optimización de redes neuronales.

Una vez analizamos los parámetros beta, a_1 y a_2 con funciones benchmark, diseñaremos un adaptador difuso con los parámetros que resulten afectar más en el desempeño del algoritmo de Mayfly, para así mejorar su rendimiento y optimizar la arquitectura de una red neuronal.

La estructura de este trabajo es de la siguiente manera, en la Sección 2 tenemos el marco teórico, donde hablaremos de los antecedentes del algoritmo de Mayfly así como de las variantes que han surgido del mismo hasta el momento, luego tenemos la Sección 3, en esta Sección se mostrará el planteamiento del problema así como la hipótesis, en la Sección 4 hablaremos acerca del desarrollo de esta tesis, empezando con el estudio de impacto de parámetros en el rendimiento del algoritmo de Mayfly, el diseño del adaptador difuso, pruebas estadísticas y la optimización de la red neuronal para la serie Mackey Glass, a continuación se encuentra la Sección y finalmente la Sección de conclusiones donde podemos saber con exactitud si el algoritmo de Mayfly con adaptación difusa resultó mejorar el rendimiento en comparación con su versión original y que tan eficiente resultó aplicado a la optimización de arquitecturas de red para series de tiempo

2. Marco teórico

Primeramente, comenzaremos mencionando que los algoritmos genéticos son principalmente utilizados para resolver problemas en los cuales los algoritmos clásicos supondrían un gran costo computacional, dichos algoritmos se basan en la evolución genética de la teoría de Darwin por lo tanto iniciamos con un conjunto de cromosomas donde cada uno representa una posible solución al problema dado, este cromosoma se genera aleatoriamente con las dimensiones que consideremos adecuadas para el problema específico que busquemos resolver, así como el número de población que tendremos.

2.1 Representación de un algoritmo genético

A continuación, se mostrará paso a paso como se representa un algoritmo genético y cómo evoluciona para encontrar la solución en un conjunto llamado población, que se compone de cromosomas y este a su vez de genes.

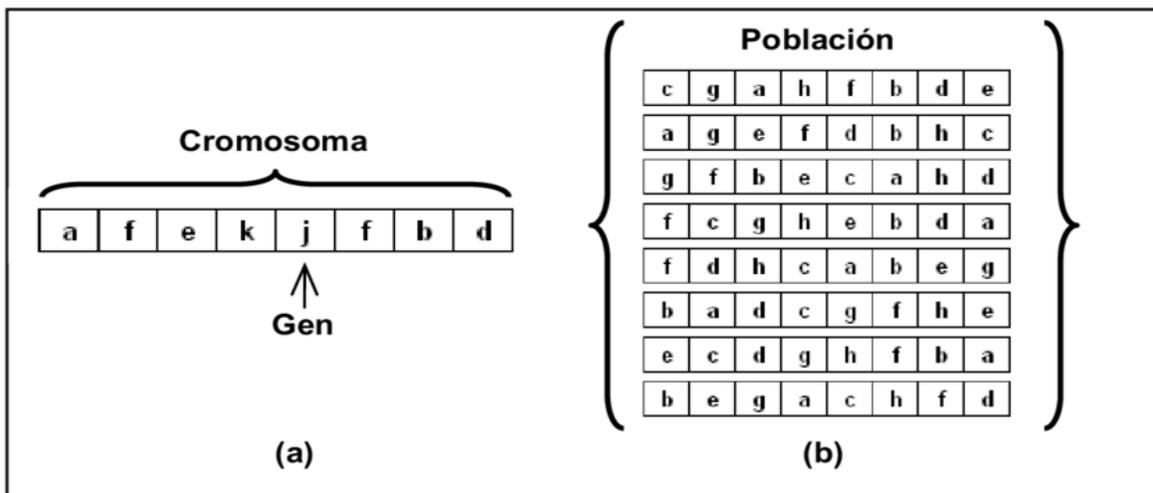


Figura 2.1 Representación de un algoritmo genético

En la Figura 2.1 podemos apreciar la representación general de un algoritmo genético, donde tenemos un cromosoma y un conjunto de ellos para formar una población de soluciones, dicho cromosoma se compone de genes los cuales irán cambiando con el objetivo de irnos acercando a la solución, esto será así conforme hagamos la reproducción de los mismos también llamada cruce y la mutación.

2.1.2 Calculo del fitness

El fitness es de vital importancia en los algoritmos genéticos ya que es la manera en que estos determinan si una solución es buena, es decir evalúan cada uno de los cromosomas de la población para determinar quiénes son los mejores individuos y a partir de aquí tomar a los mejores para ir creando descendencia con ellos, el fitness siempre será diferente para cada problema que se busque resolver.

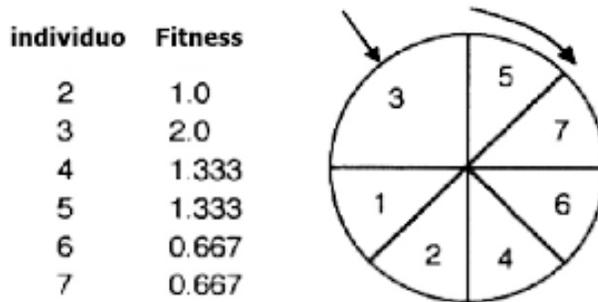


Figura 2.2 Determinación de aptitud de cada individuo

En la Figura 2.2 se puede apreciar el cálculo del fitness para diferentes individuos de la población, en este ejemplo observamos que el individuo número 6 y el 7 tienen el menor fitness lo que significa que son los peores individuos, en cambio el 3 y el 4 son los mejores por lo cual serán usados para crear descendencia y el resto de las soluciones serán eliminadas.

2.1.3 Métodos de cruce.

Comenzaremos explicando que un cruce de cromosoma consiste en seleccionar a 2 cromosomas dentro de la población generalmente a los que tienen un mejor fitness y estos se denominan padres, de estos cromosomas se seleccionan genes de ambos para formar un cromosoma nuevo, los genes que se seleccionaran depende del método de cruce que se haya elegido y esto dependerá del algoritmo genético en cuestión.

2.1.3.1 Método de cruce de un punto.

Una vez seleccionados los dos cromosomas padres estos se cortan por un punto. Se copia la información genética de uno de los padres desde el inicio hasta el punto de cruce y el resto se copia del otro progenitor, a continuación, se mostrará un ejemplo de cómo se realiza este tipo de cruce.

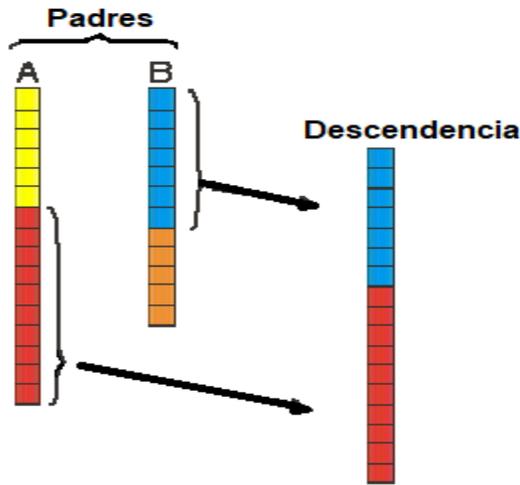


Figura 2.3 Cruce de un punto

En la Figura 2.3 tenemos 2 padres llamados A y B los cuales se combinarán para crear una nueva descendencia y usaran el método de cruce de un solo punto, el cual nos dice que seleccionaremos hasta qué punto tomaremos los genes del primer progenitor para posteriormente tomar el resto de genes del 2do.

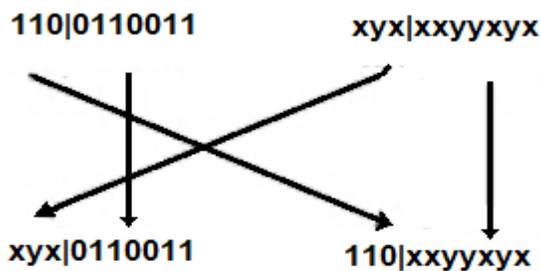


Figura 2.4 Cruce de un punto codificación

En la Figura 2.4 podemos apreciar gráficamente el método de cruce de un solo punto ya que es así como normalmente se representa un cromosoma, aquí apreciamos que

los genes señalados en 'x' e 'y' se combinan para formar al nuevo individuo o solución al problema específico para el cual este siendo aplicado el algoritmo genético.

2.1.3.2 Método de cruce de dos puntos.

Este método de cruce es muy parecido al método anterior, sin embargo, en este caso como su nombre lo indica sus padres se cortan por dos puntos. Se copiará al descendiente los genes de un cromosoma progenitor desde el principio hasta el primer punto de cruce, los genes del otro progenitor desde el primer punto de cruce hasta el segundo y del segundo punto de cruce hasta el final se copiará del otro progenitor.

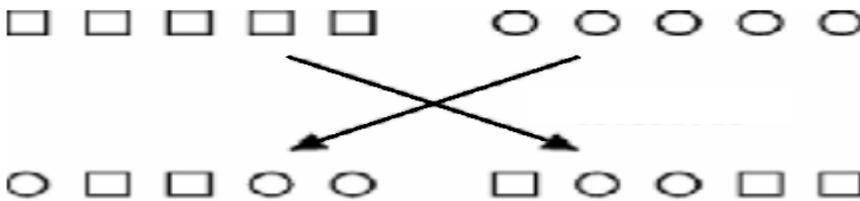


Figura 2.5 Método de cruce de 2 puntos

En la Figura 2.5 ilustramos el método de cruce de 2 puntos en el cual apreciamos de manera muy gráfica la combinación de ambos progenitores de tal manera que se hagan 4 combinaciones y puedan crear descendencia más diversificada.

2.1.3.3 Método de cruce uniforme.

Cada gen del descendiente se obtiene de cualquiera de los padres de forma aleatoria. Una opción es generar un número aleatorio. Si este número supera un cierto umbral se elegirá un padre determinado y si no lo supera se elige al otro.



Figura 2.6 Método de cruce uniforme

En la Figura 2.6 ilustramos la combinación de 2 cromosomas por medio del cruce uniforme en el cual dicha combinación será siempre diferente y manejado con un umbral personalizado por el programador, ya que la posibilidad de tomar genes de un padre o de otro es aleatoria.

2.1.4 Métodos de selección

Como su nombre lo indica un método de selección se encargará de elegir de alguna manera a los individuos con mejor fitness, ya que después estos serán usados en la reproducción o apareamiento con el fin de transmitir las mejores características a la nueva generación y logrando así acercarnos a la mejor solución del problema, para este fin contamos con diferentes métodos de selección sin embargo aquí analizaremos solo algunos de ellos.

2.1.4.1 Selección por ruleta

Este método consiste en que cada cromosoma contendrá una probabilidad de ser seleccionado para el apareamiento sin embargo esta probabilidad es directamente proporcional al fitness que contenga cada uno de estos, sin embargo tendremos problemas utilizando este tipo de selección si hay un cromosoma con un nivel muy superior de fitness ya que esto significaría que habrá una reducción de diversidad genética considerable y por lo tanto no exploraremos lo suficiente para encontrar una solución aceptable al problema.

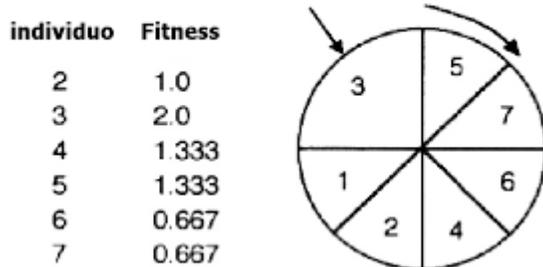


Figura 2.7 Selección por ruleta

En la Figura 2.7 tenemos un total de 6 individuos con sus respectivos fitness, como podemos ver el individuo numero 3 contiene un fitness mayor por lo que su probabilidad de ser seleccionado en la ruleta incrementa con respecto a los demás, esto porque crear una descendencia con este individuo podría traer mejores generaciones y acercarnos más a la

solución óptima del problema, sin embargo siempre es importante tener en cuenta a los que tienen un fitness menor aunque con una probabilidad más baja para así favorecer la exploración del algoritmo y no solo la explotación, caso que se nos presentaría si solo tomáramos en cuenta los de fitness mayor.

2.1.4.2 Selección elitista

Este método consiste básicamente en conservar siempre uno o 2 cromosomas de mayor fitness de una generación a otra sin alterar sus genes ya sea con mutación o con el cruce, ya que estos se copian tal cual generación tras generación, esto se hace porque en algunas ocasiones podría ocurrir que perdamos al mejor cromosoma o solución de nuestro problema y gracias a este tipo de selección podemos prevenir ese problema.

2.1.4.3 Selección por torneo

Este método como su nombre lo indica realiza torneos entre los diferentes cromosomas, es decir, selecciona individuos aleatoriamente y compara sus fitness, el que tenga mayor aptitud será el ganador, aquí es importante mencionar que entre más individuos entren al torneo la probabilidad de que un cromosoma con fitness bajo sea seleccionado disminuirá, por lo que es importante realizar este tipo de selección con una variedad importante de cromosomas para tener mejores resultados.

2.1.4.4 Selección escalada

Este método no es recomendable de usarse al comenzar un algoritmo genético debido a que la discriminación selectiva es más alta y eso quiere decir que se hace una mayor discriminación de individuos lo cual puede llevar a que el algoritmo tenga poca exploración, por lo tanto, es comúnmente usado cuando todos los individuos tengan ya una aptitud elevada, para trabajar la explotación de las soluciones dentro del algoritmo.

2.1.4.4 Selección jerárquica

Este tipo de selección es muy peculiar ya que atraviesa diferentes tipos de etapas, en las primeras generaciones suele ser menos discriminativas y conforme estas van incrementándose se hace más rigurosa, este tipo de selección tiene la ventaja de ocupar un costo computacional menor ya que en las primeras etapas descarta las opciones menos viables y enfoca los recursos en los individuos con mayor aptitud.

2.1.5 Mutación

Una vez elegido el tipo de cruce es de vital importancia establecer el porcentaje de mutación que tendrá la descendencia esto se hace con el principal objetivo de sacar el algoritmo de mínimos locales en caso de que estos se presenten, sin embargo debemos tener cuidado con el factor de mutación que decidamos elegir ya que esto podría afectar enormemente los resultados esperados y siempre será mejor mantener un factor bajo de mutación debido a que si lo exageramos tendremos como resultado simplemente un algoritmo de búsqueda de soluciones aleatorias lo cual rompe el paradigma para lo que fueron diseñados los algoritmos genéticos y dejaría de considerarse como tal, el porcentaje más usado comúnmente es de 0.1 pero este suele variar a valores muy cercanos a cero en algunos problemas, ya que habrá algunos en los cuales podría ser contraproducente, ahora que conocemos la importancia de la mutación mencionaremos en que consiste, la mutación es la probabilidad de que algún gen cambie a la inversa es decir de 0 a 1 o de 1 a 0 dichos genes serán la combinación de los padres en el proceso de apareamiento o también denominado cruce, entonces se elige un gen al azar y se aplica la mutación pudiendo así generar mejores soluciones en la mayoría de los casos.

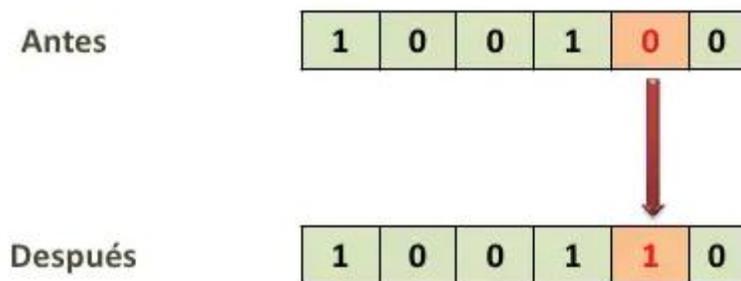


Figura 2.8 Cromosoma mutado

En la Figura 2.8 ejemplificamos como afecta la mutación a un cromosoma aleatorio, aquí podemos ver que el gen 2 de derecha a izquierda ha sido mutado ya que originalmente se encontraba con un valor de 0 y en el nuevo individuo este valor cambiara a 1, sin embargo esto no significa que solo un gen pueda mutar, es decir, pueden ser varios genes al mismo tiempo pero no es recomendable ya que podríamos caer en exceso de mutación y la respuesta que buscamos sería difícil de encontrar debido al efecto de aleatoriedad excesiva que tendríamos en el algoritmo.

2.2 Método de Mayfly

Mayfly es una especie de insectos del cual sus crías vivirían en el agua durante varios años y cuando maduren se descortezarán y volarán por el aire. Sin embargo, estos insectos maduros solo tienen el fin de reproducirse ya que su periodo de vida es corto [1].

Al observar a las efímeras machos y hembras corriendo en busca de pareja, se propuso el algoritmo MA. Desarrollado a partir del algoritmo PSO, las efímeras también actualizan sus posiciones. Sin embargo, convergen más rápido al óptimo por lo tanto el algoritmo de Mayfly es una mejora del algoritmo PSO [1].

2.2.1 Movimiento de efímeras machos

Asumiendo que x_i^t es la posición actual de Mayfly en el espacio de búsqueda en el paso de tiempo t , la posición se cambia agregando una velocidad v_i^{t+1} a la posición actual;

Esto se puede formular como:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2.1)$$

Teniendo en cuenta que las efímeras machos están siempre a unos metros sobre el agua realizando la danza nupcial, suponemos que no pueden desarrollar grandes velocidades y que se mueven constantemente. Como resultado, la velocidad de una efímera macho se calcula como

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-\beta r_p^2} (pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2} (gbest_j - x_{ij}^t) \quad (2.2)$$

Donde v_{ij}^t es la velocidad de efímera i en la dimensión $j=1-n$ al paso de tiempo t , x_{ij}^t es la posición de la efímera i en la dimensión j al paso de tiempo t , a_1 y a_2 son constantes de atracción positiva que se utilizan para escalar la contribución del componente cognitivo y social, respectivamente, $pbest$ la mejor posición local, $gbest$ la mejor posición global y β es un coeficiente de visibilidad fijo [21].

2.2.2 Movimiento de efímeras hembras

Asumiendo que y_i^t es la posición actual de Mayfly hembra i en el espacio de búsqueda en el paso de tiempo t , la posición se cambia agregando una velocidad v_i^{t+1} a la posición actual; Esto se puede formular como:

$$y_i^{t+1} = y_i^t + v_i^{t+1} \quad (2.3)$$

El proceso de atracción puede ser manejado aleatoriamente sin embargo en este algoritmo se empleó una secuencia determinista en la cual la mejor hembra debe ser atraída por el mejor macho, la segunda mejor hembra por el segundo mejor macho y así sucesivamente, sus velocidades pueden calcularse con la siguiente formula:

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^t + a_2 e^{-\beta r_{mf}^2} (x_{ij}^t - y_{ij}^t), & \text{if } (f(y_i) > f(x_i)) \\ v_{ij}^t + flr, & \text{if } (f(y_i) \leq f(x_i)) \end{cases} \quad (2.4)$$

Donde V_{ij}^t es la velocidad de las efímeras hembras i en la dimensión $j = 1-n$ al paso de tiempo t , y_{ij}^t es la posición de la efímera hembra i en la dimensión j al paso de tiempo t , a_2 es una constante de atracción positiva, β es un coeficiente de visibilidad fijo, mf es la distancia cartesiana entre efímeras machos y hembras, fl es un coeficiente de caminata aleatorio, utilizado cuando una hembra no es atraída por un macho, por lo que vuela aleatoriamente y r es un valor aleatorio entre $(-1,1)$ [21].

2.3 Algoritmo OBL-MO

Las investigaciones demostraron que no solo los mejores candidatos o las mejores trayectorias históricas funcionarían bien para guiar a las personas en enjambres para encontrar la mejor solución, sino que las peores y peores trayectorias históricas también funcionarían bien al hacerlo. Tales situaciones podrían tratarse directamente como pares de oposiciones y satisfacer la antigua filosofía china Yin-Yang la cual nos muestra que habría una fuerte relación entre los lados Yin y Yang, o bien y mal, negativo y positivo, fuego y agua, seco y húmedo, entonces, se propuso la regla de aprendizaje basado en la oposición (OBL) [4].

Este algoritmo es una variante del algoritmo de Mayfly en el cual al igual que su predecesor se dividirá en dos tipos, la efímera macho y la efímera hembra, ambos actualizarán sus posiciones con sus velocidades $v_i(t)$ [4].

$$v_i^{t+1} = \begin{cases} gv_i^t + a_1 e^{-\beta r_p^2} [x_i^t - y_i^t] & f(y_i^t) > f(x_i^t) \\ gv_i^t + flr_1 & f(y_i^t) \leq f(x_i^t) \end{cases} \quad (2.5)$$

Donde $f(x)$ es la función de aptitud, g y fl son ponderaciones que se reducirían de su valor máximo al mínimo. a_1 y β son constantes. r_1 es el número aleatorio en distribución uniforme con el intervalo de -1 y 1. r_m representa la distancia cartesiana entre la pareja femenina y masculina [4].

Teniendo en cuenta la filosofía Yin-Yang, podríamos encontrar que, si la posición actual estuviera muy lejos del mejor, entonces, la posición opuesta se acercaría al mejor en un dominio de definición dado. Por tanto, podemos calcular lo contrario a la posición actual [4].

2.4 Algoritmo de Mayfly modificado

En este algoritmo, los individuos en enjambres se identificarían específicamente como efímeras machos y hembras. Y ambos realizan diferentes comportamientos de actualización. Sin embargo, en la versión original del algoritmo de Mayfly, si las posiciones

actuales estuvieran lejos del mejor candidato o de las mejores trayectorias históricas, los individuos correrían hacia la mejor posición con menor velocidad. Por el contrario, si las posiciones actuales están cerca del mejor candidato global o de las mejores trayectorias históricas, entonces los individuos actuarían con velocidades más rápidas [3].

El algoritmo de Mayfly se dividirá en dos tipos, el macho de corta vida para el i -ésimo individuo en la iteración actual t , y la hembra de corta duración. Ambos tipos actualizarán sus posiciones con sus velocidades en la iteración actual.

$$v_i^{t+1} = gv_i^t + a_1 e^{-\beta r_p^2} [x_{hi} - x_i^t] + a_2 e^{-\beta r_g^2} [x_g - x_i^t] \quad (2.6)$$

Donde, g es una variable disminuida linealmente desde el valor máximo a uno más pequeño. a_1 , a_2 y β son dos constantes para equilibrar los valores. r_p y r_g son dos variables que se utilizan para determinar la distancia cartesiana entre los individuos y su mejor posición histórica, la mejor posición global en enjambres [3].

Las efímeras machos siempre serían fuertes y, en consecuencia, funcionarían mejor en la optimización, al igual que los individuos en enjambres del algoritmo PSO, los individuos en el algoritmo de Mayfly actualizarían las posiciones de acuerdo con sus posiciones actuales $p_i(t)$ y velocidad $v_i(t)$ en la iteración actual.

$$p_i(t+1) = p_i(t) + v_i(t+1) \quad (2.7)$$

2.5 Algoritmo NMO

Es un algoritmo de optimización de efímeros negativos. En el cual los machos de las efímeras actualizarían sus velocidades de acuerdo con los peores candidatos junto con sus peores trayectorias. A diferencia de la interpretación positiva normal, en el algoritmo NMO, las efímeras machos huirían de sus peores trayectorias y los peores candidatos globales, las efímeras hembras tienen el deber de reproducirse, por lo tanto, todas tienen prisa por aparearse. Sus velocidades se actualizarían de acuerdo con sus compañeros. Es decir, la velocidad de la i -ésima efímera hembra se guiaría para actualizarse de acuerdo con su valor

de aptitud f [$y_i(t)$] y su pareja f [$x_i(t)$], dicha velocidad será actualizada por la siguiente ecuación [5].

$$v_i^{t+1} = \begin{cases} gv_i^t + a_1 e^{-\beta r_{mf}^2} x_i^t - y_i^t & f(x_i^t - y_i^t) > f(x_i^t) \\ gv_i^t + flr_1 & f(y_i^t) \leq f(x_i^t) \end{cases} \quad (2.8)$$

Donde, g y fl son pesos que se reducirían de su valor máximo al mínimo. a_1 y β son constantes. r_1 es el número aleatorio en distribución uniforme con el intervalo de -1 y 1. r_p representa la distancia cartesiana entre la pareja femenina y masculina [5].

Una vez que se actualizaron las velocidades y posiciones, todos los individuos se clasificarían nuevamente y la mitad superior mejor sería renombrada como efímeras machos y otra mitad como efímeras hembras, los experimentos de simulación mostraron que los algoritmos NMO y MO(Mayfly) funcionarían bien optimizando las funciones de referencia unimodal o multimodal, incluso para la no simétrica. Sin embargo, para las funciones de referencia unimodal, el algoritmo de Mayfly funcionaría mejor que el NMO [5].

2.6 El algoritmo de la mosca de mayo BMA

El algoritmo ha sido diseñado mediante la concepción explicada y el modelado de tres algoritmos, el algoritmo de optimización de enjambre de partículas (PSO), el algoritmo de optimización de luciérnagas (FOA) y el algoritmo genético (GA) que lo hicieron tan eficiente debido al uso de las tres populares ventajas de los métodos. Con la hibridación de los algoritmos, la explotación del algoritmo PSO se combina con las características de exploración de los otros algoritmos para lograr una mayor eficiencia en el modelado de las efímeras, especialmente el modelo de reproducción. Este modelo ayuda a las efímeras más aptas a sobrevivir después de la eclosión de los huevos sin tener en cuenta su vida útil. Primero buscamos el macho efímero con el menor valor de costo en la población, luego seleccionamos la cantidad de semillas, seguido de esto las hembras efímeras atraen a los machos elite para generar una nueva descendencia usando esto como radio de crecimiento, la actualización de su velocidad es dada por la siguiente ecuación [6].

$$v_{ij}^{t+1} = v_{ij}^t + b_1 \exp(-\beta r_p^2) (x_1^{ij} - x_{ij}^t) + b_2 \exp(-\beta r_g^2) (x_g^j - x_{ij}^t) \quad (2.9)$$

Donde x_{ij}^t y v_{ij}^t describen la ubicación y la velocidad de i-esima mosca de mayo en la dimensión j, β describe un coeficiente definido para hacer que la efímera sea visible para otros, r_p y r_g representan la distancia cartesiana restringida en el rango, x_1^i y x_g^j describen el Mayfly con la posición optimiza para la efímera b_1 y b_2 describen las constantes de atracción positiva y factores, sociales respectivamente [6].

2.7 Redes neuronales

Para entender una red neuronal artificial primero hay que saber que es una red neuronal biológica debido que está basada en los mismos principios, una red neuronal biológica es un conjunto de neuronas interconectadas entre sí que comparten algún tipo de información con el fin de proporcionar aprendizaje en conjunto esto lo hacen a través de la sinapsis, ahora bien la red neuronal artificial es una representación computacional de un conjunto de nodos conectados entre sí, que manejan pesos denotados por ' w_i ', los cuales se van modificando conforme la red va aprendiendo.

2.7.1 Representación de una red neuronal

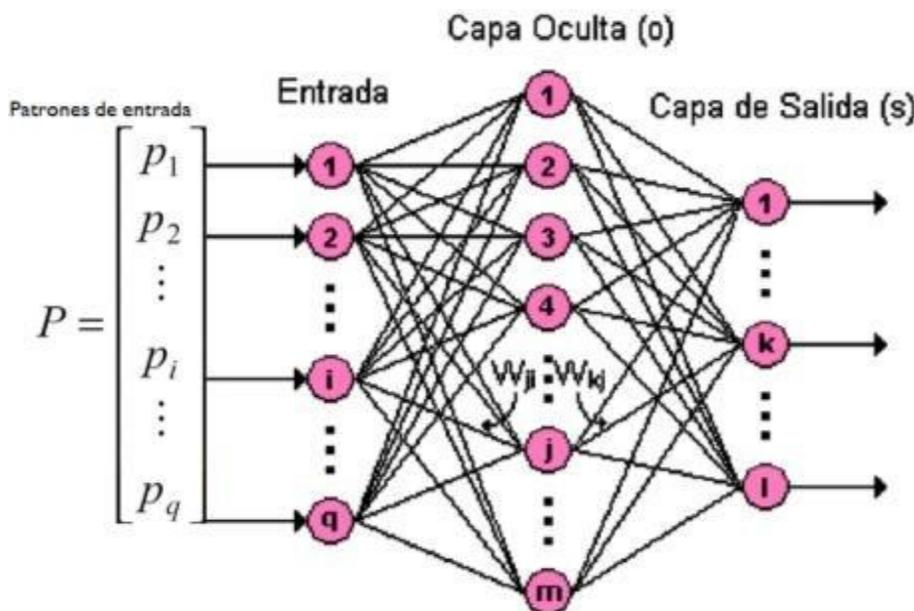


Figura 2.9 Red neuronal

En la Figura 2.9 tenemos un esquema general de lo que es una red neuronal, empezando con los patrones de entrada, estos contendrán los datos de entrenamiento, en este caso tenemos una red neuronal de 3 capas, en la primera recibiremos los datos de entrada, y estos a su vez serán pasados a las neuronas en la capa oculta para por ultimo llegar a las neuronas de la capa de salida donde obtendremos una interpretación final por parte de la red neuronal.

2.8 Serie de tiempo Mackey Glass

Esta serie caótica es ampliamente usada para evaluar el desempeño de una red neuronal, fue inventada por los matemáticos Michael Mackey y León Glass, es también denominada una ecuación diferencial y utilizada con retardos sin embargo tiene un comportamiento caótico en $\tau=17$ que será el principio utilizado para evaluar nuestra red neuronal y será calculada con la siguiente ecuación.

$$\frac{dx(t)}{dt} = \frac{0.2x(-t - \tau)}{1 + X^{10}(t - \tau)} - 0.1x(t) \quad (2.10)$$

2.9 Lógica difusa

Este tipo de lógica es utilizada cuando el proceso en cuestión es muy complicado o costoso computacionalmente, a diferencia de la lógica tradicional de decisión aquí no hay blanco y negro o si y no, esto se trata más bien de valores discretos entre 0 y 1, logrando con esto tener una perspectiva diferente para resolver problemas de control entre otros, este tipo de lógica se compone de varios elementos.

2.9.1 Reglas difusas

Son aquellas que indican un resultado dependiendo de un conjunto de condiciones previamente especificadas, la primera condición de la oración contendrá las distintas entradas del sistema denominadas también como funciones de membrecía, la 2da parte contendrá la acción que debe llevar a cabo.

2.9.2 Funciones de membresía

Este tipo de funciones representa el grado de pertenencia de un elemento dado sin embargo hay una variedad de formas que pueden adquirir estas, entre las cuales se encuentran las funciones tipo triangulares, trapezoidales y gaussianas.

2.10 Hipótesis

El desempeño del algoritmo de Mayfly puede verse favorecido al aplicar un adaptador difuso de parámetros en la optimización de una arquitectura de red.

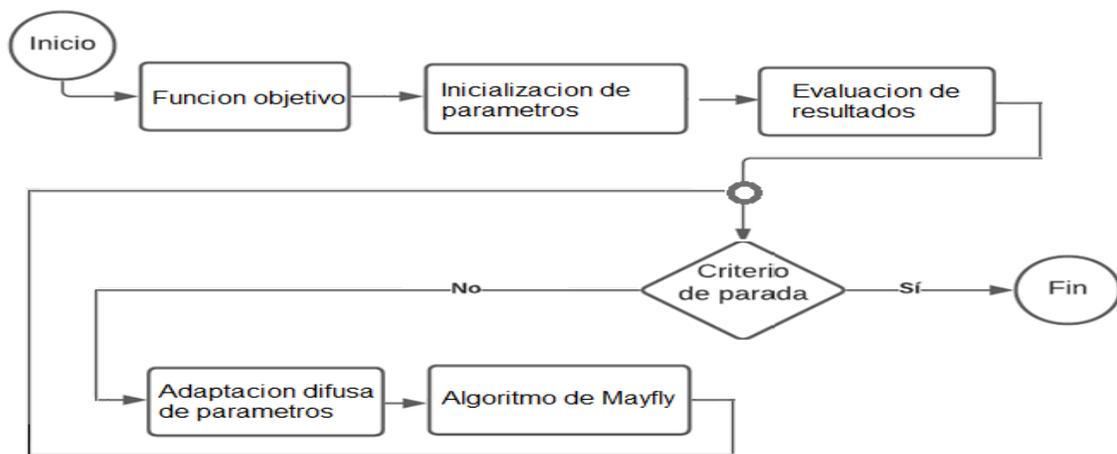


Figura 2.10 Esquema General de hipótesis

Como podemos apreciar en la Figura 2.10 tenemos un panorama general del método propuesto para optimizar el algoritmo de Mayfly, debido a que la adaptación difusa ha demostrado mejorar la eficiencia de algunos otros algoritmos como PSO, esperamos que resulte benéfico al aplicarse en este caso al algoritmo de Mayfly.

3. Planteamiento del problema

En el presente trabajo de tesis se ha planteado hacer una mejora del algoritmo de Mayfly aplicado a optimización de arquitectura de redes neuronales, dicha mejora se pretende lograr al aplicar una adaptación difusa de parámetros en base al impacto en el desempeño del algoritmo de Mayfly que estos representen.

3.1 Definición de puntos en el problema.

Evaluar el impacto de los parámetros principales, los cuales son a_1 , a_2 y β resulta tardado ya que debemos analizar las diferentes dimensiones y funciones benchmark y posterior a eso graficar resultados para apreciar los resultados, esto con el fin de utilizar los parámetros que tengan un impacto relevante en el desempeño del algoritmo.

Para diseñar el adaptador difuso de parámetros debemos pasar por varias fases de prueba, esto significa que no podemos realizar un adaptador complejo desde el inicio debido a que podríamos hacer cálculos innecesarios o más complejos sin que mejoren nuestros resultados, por lo tanto, debemos empezar con un adaptador de 1 entrada y 1 salida, así como funciones de membresía triangulares.

Otro inconveniente a la hora de hacer la adaptación difusa es que las funciones de membresía no se pueden optimizar completamente a prueba y error, debido al excesivo tiempo que llevaría probar todas, por su complejidad de variantes y el tiempo de ejecución de algoritmo.

3.2 Resolución de la problemática principal.

Una vez analizada la problemática se pretende lograr un mejor desempeño en el algoritmo de Mayfly por medio de la adaptación difusa de parámetros en los parámetros de mayor impacto.

De tal manera que podamos aprovechar el uso de este nuevo algoritmo modificado para optimizar una arquitectura de red neuronal, enfocándonos en el número de neuronas óptimo para distintas aplicaciones, en nuestro caso será empleado en una aplicación para predecir el valor de la serie caótica de Mackey Glass.

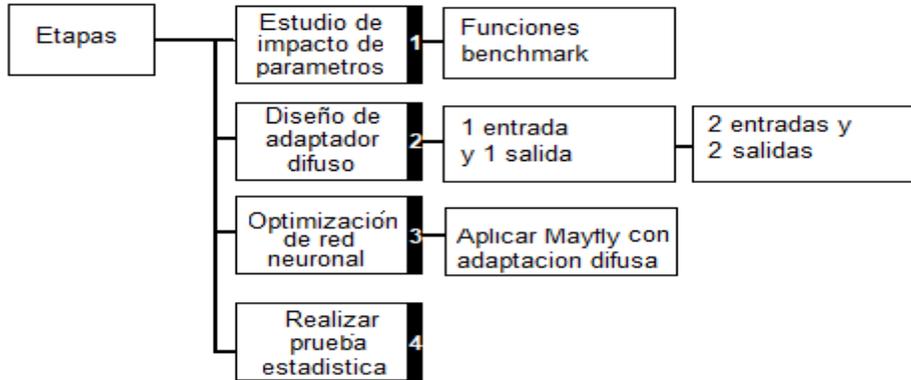


Figura 3.1 Esquema general de planteamiento del problema

En la Figura 3.1 tenemos las etapas por las que atravesó el desarrollo del adaptador difuso de parámetros para el algoritmo de Mayfly

4. Desarrollo y Resultados

El desarrollo de este trabajo de tesis fue llevado a cabo en diferentes etapas, cada una de estas contribuyendo en el avance de la investigación con el fin de mejorar el algoritmo de Mayfly mediante el uso de un adaptador difuso, primeramente, se realizó una prueba del impacto de parámetros, la cual tenía como objetivo encontrar el o los parámetros de mayor impacto en el rendimiento del algoritmo, así como el rango óptimo de valores para dichos parámetros.

4.1 Estudio del impacto de parámetros

Como se mencionó anteriormente el primer paso en el desarrollo de esta tesis consistió en el análisis del impacto de parámetros, para lo que fue necesario utilizar funciones benchmark y diferentes dimensiones, en este caso utilizamos 10,20 y 50 dimensiones con el fin de encontrar los parámetros a los cuales será relevante aplicarle una adaptación difusa.

Tabla 4.1 Representación de las funciones benchmark

Abreviación	Nombre de la función
F1	Sphere
F2	Rastrigin
F3	Ackley
F4	Dejong1
F5	Griewank
F6	Schwefel

F7	Styblinski tank
F8	Wood
F9	Hyper-ellipsoid
F10	Powell

Tabla 4.2 Evaluación de las funciones benchmark para 10 dimensiones variando el parametro beta

Función	parámetro beta para 10 dimensiones			
		1	3	10
F1	\bar{x}	1.231×10^{-238}	1.042×10^{-244}	2.146×10^{-249}
	s	0	0	0
F2	\bar{x}	3.200×10^{-2}	0	9.950×10^{-2}
	s	1.817×10^{-1}	0	3.036×10^{-1}
F3	\bar{x}	2.980×10^0	2.980×10^0	2.980×10^0
	s	1.817×10^{-1}	0	3.036×10^{-1}
F4	\bar{x}	2.087×10^{-242}	9.334×10^{-253}	2.127×10^{-245}
	S	1.355×10^{-15}	1.355×10^{-15}	1.355×10^{-15}
F5	\bar{x}	4.400×10^{-3}	2.100×10^{-3}	1.700×10^{-3}
	s	7.400×10^{-3}	4.700×10^{-3}	4.200×10^{-3}
F6	\bar{x}	4.150×10^3	-3.864×10^2	4.150×10^3
	s	6.826×10^{-1}	6.920×10^0	2.775×10^{-12}
F7	\bar{x}	-3.817×10^2	-3.839×10^2	-3.856×10^2
	s	9.206×10^0	8.776×10^0	9.080×10^0
F8	\bar{x}	7.530×10^{-2}	1.140×10^{-30}	1.422×10^{-30}
	s	4.125×10^{-1}	2.374×10^{-30}	5.363×10^{-30}
F9	\bar{x}	9.632×10^{-258}	4.645×10^{-244}	6.415×10^{-253}
	s	0	0	0
F10	\bar{x}	2.0482×10^{-98}	3.1343×10^{-103}	7.2248×10^{-104}
	s	1.1218×10^{-97}	1.7167×10^{-102}	3.9263×10^{-103}

Tabla 4.3 Evaluación de las funciones benchmark para 10 dimensiones variando el parámetro a₁

Función	parámetro a ₁ para 10 dimensiones			
		1	3	10
F1	\bar{x}	5.5873×10^{-256}	8.2493×10^{-214}	7.9061×10^{-217}
	s	0	0	0
F2	\bar{x}	3.3200×10^{-2}	6.6300×10^{-2}	3.3200×10^{-2}
	s	1.8170×10^{-1}	2.5240×10^{-1}	1.8170×10^{-1}
F3	\bar{x}	2.9805×10^0	2.9805×10^0	2.9805×10^0
	s	1.3550×10^{-15}	1.3550×10^{-15}	1.3550×10^{-15}
F4	\bar{x}	6.7385×10^{-243}	2.3152×10^{-221}	1.5242×10^{-213}
	S	0	0	0
F5	\bar{x}	3.3000×10^{-3}	0.0001×10^{-3}	2.0000×10^{-3}
	s	5.2000×10^{-3}	4.7000×10^{-3}	3.7000×10^{-3}
F6	\bar{x}	4.1504×10^3	6.8000×10^{-3}	4.1503×10^3
	s	2.7751×10^{-12}	8.7000×10^{-3}	5.4460×10^{-1}
F7	\bar{x}	-3.8233×10^2	-3.8459×10^2	-3.860×10^2
	s	1.1342×10^1	8.9023×10^0	8.7454×10^0
F8	\bar{x}	3.6016×10^{-31}	3.2762×10^{-30}	5.6805×10^{-30}
	s	1.1646×10^{-30}	6.1005×10^{-30}	1.0670×10^{-29}
F9	\bar{x}	9.5773×10^{-241}	4.0063×10^{-223}	5.8795×10^{-212}
	s	0	0	0

F10	\bar{x}	2.6545×10^{-17}	1.7270×10^{-16}	3.9018×10^{-16}
	s	9.0270×10^{-17}	2.7811×10^{-16}	6.2289×10^{-15}

Tabla 4.4 Evaluación de las funciones benchmark para 10 dimensiones variando el parámetro a_2

Función	parámetro a_2 para 10 dimensiones			
		1	3	10
F1	\bar{x}	5.3682×10^{-139}	2.1765×10^{-13}	1.7390×10^{-1}
	s	2.9375×10^{-138}	1.1867×10^{-12}	1.439×10^{-1}
F2	\bar{x}	0	2.3220×10^{-1}	9.9703×10^0
	s	0	5.6550×10^{-1}	4.1971×10^0
F3	\bar{x}	2.9805×10^0	2.9805×10^0	3.2146×10^0
	s	1.3550×10^{-15}	1.3550×10^{-15}	1.0280×10^{-1}
F4	\bar{x}	1.4992×10^{-140}	3.4933×10^{-12}	1.4790×10^{-1}
	S	7.8799×10^{-140}	1.9039×10^{-11}	1.3640×10^{-1}
F5	\bar{x}	5.5000×10^{-3}	8.5000×10^{-2}	1.6300×10^{-2}
	s	7.4000×10^{-3}	1.7500×10^{-1}	1.4600×10^{-2}
F6	\bar{x}	4.1504×10^3	4.1504×10^3	4.1504×10^3
	S	2.7751×10^{-12}	2.7751×10^{-12}	2.5400×10^{-2}
F7	\bar{x}	-3.8555×10^2	-3.8553×10^2	-3.7200×10^2
	s	8.7980×10^0	8.0342×10^0	1.3761×10^1
F8	\bar{x}	2.4568×10^{-30}	4.1500×10^{-2}	1.1230×10^{-1}
	s	8.3848×10^{-30}	2.2740×10^{-1}	1.0210×10^{-1}
F9	\bar{x}	2.7140×10^{-136}	4.1774×10^{-14}	4.1250×10^{-1}
	s	1.4145×10^{-135}	1.9507×10^{-13}	4.4690×10^{-1}
F10	\bar{x}	1.0525×10^{-158}	1.1838×10^{-8}	2.0000×10^{-2}
	s	5.7455×10^{-158}	3.6954×10^{-8}	3.2000×10^{-3}

Tabla 4.5 Evaluación de las funciones benchmark para 20 dimensiones variando el parámetro beta

Función	parámetro beta para 20 dimensiones			
		1	3	10
F1	\bar{x}	2.2389×10^{-85}	1.4310×10^{-83}	5.7410×10^{-85}
	s	6.2447×10^{-85}	6.4675×10^{-83}	2.5457×10^{-84}
F2	\bar{x}	2.1891×10^0	2.5936×10^0	2.1701×10^0
	s	3.6250×10^0	2.3445×10^0	1.2814×10^0
F3	\bar{x}	2.9826×10^0	2.9826×10^0	2.9856×10^0
	s	1.1600×10^{-2}	1.1160×10^{-2}	1.6600×10^{-2}
F4	\bar{x}	1.4083×10^{-86}	1.9740×10^{-83}	5.3884×10^{-83}
	S	5.32111×10^{-86}	1.0654×10^{-82}	2.9432×10^{-82}
F5	\bar{x}	1.5000×10^{-3}	1.1000×10^{-3}	1.3000×10^{-3}
	s	3.0000×10^{-3}	2.8000×10^{-3}	3.0000×10^{-3}
F6	\bar{x}	8.3013×10^3	8.3010×10^3	8.3013×10^3
	S	1.2925×10^0	9.4850×10^{-1}	1.2925×10^0
F7	\bar{x}	-7.4279×10^2	-7.4047×10^2	7.3714×10^2
	s	2.1879×10^1	2.3545×10^1	2.3164×10^1
F8	\bar{x}	3.7865×10^{-31}	1.0871×10^{-30}	1.0398×10^{-30}
	s	7.2171×10^{-31}	1.8127×10^{-30}	1.8174×10^{-30}
F9	\bar{x}	1.6499×10^{-84}	1.4959×10^{-82}	5.9297×10^{-80}
	s	7.3705×10^{-84}	7.8305×10^{-82}	3.2474×10^{-79}

F10	\bar{x}	7.7396×10^{-108}	9.4023×10^{-81}	1.9654×10^{-104}
	s	4.2390×10^{-107}	5.1499×10^{-80}	1.0765×10^{-103}

Tabla 4.6 Evaluación de las funciones benchmark para 20 dimensiones variando el parámetro a_1

Función	Parámetro a_1 para 20 dimensiones			
		1	3	10
F1	\bar{x}	3.7018×10^{-85}	1.1700×10^{-18}	1.3335×10^{-76}
	s	1.7365×10^{-84}	1.1056×10^{-77}	6.1097×10^{-76}
F2	\bar{x}	2.1226×10^0	2.5272×10^0	8.6230×10^{-1}
	s	2.1345×10^0	2.8146×10^0	9.3240×10^{-1}
F3	\bar{x}	2.9805×10^0	2.9805×10^0	2.9805×10^0
	s	1.3550×10^{-15}	1.3550×10^{-15}	1.3550×10^{-15}
F4	\bar{x}	7.3290×10^{-82}	5.7774×10^{-79}	3.6134×10^{-79}
	S	4.0143×10^{-81}	3.1329×10^{-78}	1.3513×10^{-78}
F5	\bar{x}	1.1000×10^{-3}	7.0250×10^{-80}	4.93707×10^{-4}
	s	2.8000×10^{-3}	2.0452×10^{-79}	1.9000×10^{-3}
F6	\bar{x}	8.3013×10^3	8.3010×10^3	8.3011×10^3
	S	1.2925×10^0	9.4850×10^{-1}	1.1407×10^{-7}
F7	\bar{x}	-7.4383×10^2	7.4291×10^2	7.3192×10^2
	s	2.1831×10^1	1.9891×10^1	1.8750×10^1
F8	\bar{x}	5.6132×10^{-31}	1.5920×10^{-1}	2.4200×10^{-2}
	s	1.9415×10^{-30}	7.4650×10^{-1}	1.3250×10^{-1}
F9	\bar{x}	5.8224×10^{-83}	9.8517×10^{-72}	1.8958×10^{-74}
	s	3.1608×10^{-82}	5.3960×10^{-71}	1.0383×10^{-73}
F10	\bar{x}	3.4879×10^{-19}	3.1166×10^{-16}	1.9360×10^{-16}
	s	1.1700×10^{-18}	7.2152×10^{-16}	2.5051×10^{-16}

Tabla 4.7 Evaluación de las funciones benchmark para 20 dimensiones variando el parámetro a_2

Función	Parámetro a_2 para 20 dimensiones			
		1	3	10
F1	\bar{x}	1.0091×10^{-30}	4.4000×10^{-3}	1.1916×10^0
	s	5.4293×10^{-30}	9.6000×10^{-3}	6.4280×10^{-1}
F2	\bar{x}	2.1557×10^0	1.2795×10^1	5.5739×10^1
	s	1.3598×10^0	1.0197×10^1	2.2720×10^1
F3	\bar{x}	2.9847×10^0	3.0682×10^1	3.5516×10^0
	s	1.6100×10^{-2}	9.1300×10^{-2}	1.7040×10^{-1}
F4	\bar{x}	2.7401×10^{-30}	8.7000×10^{-3}	1.3084×10^0
	S	1.4763×10^{-29}	1.7800×10^{-2}	8.1950×10^{-1}
F5	\bar{x}	1.7000×10^{-3}	1.6000×10^{-3}	9.4200×10^{-2}
	s	3.6000×10^{-3}	4.1000×10^{-3}	4.4400×10^{-2}
F6	\bar{x}	8.3011×10^3	8.3016×10^3	8.3009×10^3
	S	1.1407×10^0	1.8843×10^0	1.1520×10^{-1}
F7	\bar{x}	7.4192×10^2	7.3318×10^2	6.6449×10^2
	s	2.4846×10^1	2.3973×10^1	3.4779×10^1
F8	\bar{x}	2.53200×10^{-1}	3.1840×10^{-1}	1.7640×10^{-1}
	s	9.7680×10^{-1}	3.6610×10^{-1}	2.4900×10^{-1}
F9	\bar{x}	5.2077×10^{-32}	4.9800×10^{-2}	1.0782×10^1
	s	2.4898×10^{-31}	8.8820×10^{-2}	5.8278×10^0

F10	\bar{x}	3.5991×10^{-156}	4.0187×10^{-9}	1.5000×10^{-3}
	s	1.9711×10^{-155}	3.8543×10^{-9}	2.7000×10^{-3}

Tabla 4.8 Evaluación de las funciones benchmark para 50 dimensiones variando el parámetro beta

Función	Parámetro beta para 50 dimensiones			
		1	3	10
F1	\bar{x}	2.6622×10^{-18}	8.6067×10^{-18}	3.1524×10^{-18}
	s	4.3049×10^{-18}	2.1176×10^{-17}	7.2702×10^{-18}
F2	\bar{x}	2.0367×10^1	2.0235×10^1	2.0017×10^1
	s	7.5310×10^0	5.7713×10^0	3.9721×10^0
F3	\bar{x}	3.1119×10^0	3.1232×10^0	3.1250×10^0
	s	4.1982×10^{-2}	3.9477×10^{-2}	3.6415×10^{-2}
F4	\bar{x}	1.0703×10^{-17}	2.6249×10^{-9}	5.3448×10^{-18}
	S	2.2941×10^{-17}	1.4377×10^{-8}	1.0206×10^{-17}
F5	\bar{x}	9.8667×10^{-4}	5.7667×10^{-4}	1.6000×10^{-3}
	s	3.2129×10^{-3}	2.2190×10^{-3}	4.1000×10^{-3}
F6	\bar{x}	2.0755×10^4	2.0754×10^4	2.0754×10^4
	S	2.6869×10^0	3.0072×10^0	2.7302×10^0
F7	\bar{x}	1.5629×10^3	1.5929×10^3	1.6262×10^3
	s	6.3748×10^2	6.4187×10^1	5.5695×10^1
F8	\bar{x}	1.2424×10^{-31}	1.0058×10^{-31}	7.0257×10^{-32}
	s	2.9164×10^{-31}	1.4505×10^{-31}	1.1683×10^{-31}
F9	\bar{x}	8.4588×10^1	9.2035×10^1	5.1836×10^1
	s	3.8959×10^1	4.8202×10^1	1.7352×10^1
F10	\bar{x}	4.6825×10^{-78}	3.0523×10^{-97}	1.9030×10^{-86}
	s	8.5490×10^{-79}	1.6583×10^{-96}	1.0423×10^{-85}

Tabla 4.9 Evaluación de las funciones benchmark para 50 dimensiones variando el parámetro a_1

Función	Parámetro a_1 para 50 dimensiones			
		1	3	10
F1	\bar{x}	2.6622×10^{-18}	8.6067×10^{-18}	3.1524×10^{-18}
	s	4.3049×10^{-18}	2.1176×10^{-17}	7.2702×10^{-18}
F2	\bar{x}	2.0367×10^1	2.0235×10^1	2.0017×10^1
	s	7.5310×10^0	5.7713×10^0	3.9721×10^0
F3	\bar{x}	3.1119×10^0	3.1232×10^0	3.1250×10^0
	s	4.1982×10^{-2}	3.9477×10^{-2}	3.6415×10^{-2}
F4	\bar{x}	1.0703×10^{-17}	2.6249×10^{-9}	5.3448×10^{-18}
	S	2.2941×10^{-17}	1.4377×10^{-8}	1.0206×10^{-17}
F5	\bar{x}	9.8667×10^{-4}	5.7667×10^{-4}	1.6000×10^{-3}
	s	3.2129×10^{-3}	5.7667×10^{-4}	4.1000×10^{-3}
F6	\bar{x}	2.0755×10^4	2.0754×10^4	2.0754×10^4
	S	2.6869×10^0	3.0072×10^0	2.7302×10^0
F7	\bar{x}	1.5629×10^3	1.5929×10^3	1.6262×10^3
	s	6.3748×10^2	6.4187×10^1	5.5695×10^1
F8	\bar{x}	1.2424×10^{-31}	1.0058×10^{-31}	7.0257×10^{-32}
	s	2.9164×10^{-31}	1.4505×10^{-31}	1.1683×10^{-31}
F9	\bar{x}	8.4588×10^1	9.2035×10^1	5.1836×10^1
	s	3.8959×10^1	4.8202×10^1	1.7352×10^1

F10	\bar{x}	4.6825×10^{-78}	3.0523×10^{-97}	1.9030×10^{-86}
	s	8.5490×10^{-79}	1.6583×10^{-96}	1.0423×10^{-85}

Tabla 4.10 Evaluación de las funciones benchmark para 50 dimensiones variando el parámetro a_2

Función	Parámetro a_2 para 50 dimensiones			
		1	3	10
F1	\bar{x}	7.7702×10^{-18}	6.0070×10^{-1}	1.1431×10^1
	s	1.4913×10^{-17}	4.8350×10^{-1}	3.7181×10^0
F2	\bar{x}	2.1481×10^1	1.0418×10^2	3.2912×10^2
	s	1.1336×10^1	6.5609×10^1	5.4157×10^1
F3	\bar{x}	2.1481×10^1	3.4216×10^0	4.2677×10^0
	s	1.1356×10^1	3.0420×10^1	3.3240×10^{-1}
F4	\bar{x}	4.9914×10^{-18}	3.2050×10^{-1}	1.0555×10^1
	S	1.3840×10^{-17}	3.5010×10^{-1}	3.4654×10^0
F5	\bar{x}	9.0310×10^{-4}	1.0000×10^{-2}	3.2400×10^{-1}
	s	3.5000×10^{-3}	1.1500×10^{-2}	8.2600×10^{-2}
F6	\bar{x}	2.0754×10^4	2.0754×10^4	2.0756×10^4
	S	3.6931×10^0	3.7302×10^0	3.1330×10^0
F7	\bar{x}	1.7510×10^3	1.6892×10^3	1.4881×10^3
	s	5.0532×10^1	6.1035×10^1	7.7586×10^1
F8	\bar{x}	4.9107×10^{-30}	4.0600×10^{-2}	1.1170×10^{-1}
	s	1.1347×10^{-29}	2.2380×10^{-2}	1.0650×10^{-1}
F9	\bar{x}	3.4501×10^{-15}	7.0498×10^0	2.4866×10^2
	s	3.1062×10^{-15}	8.0035×10^0	8.5808×10^1
F10	\bar{x}	1.8524×10^{-155}	1.1834×10^{-8}	2.4866×10^2
	s	1.0416×10^{-154}	2.7001×10^{-8}	8.5808×10^1

4.1.1 Graficas del estudio de parámetros.

A continuación, mostraremos las gráficas que representan el comportamiento de los parámetros variando sus parámetros a_1 , a_2 y beta.

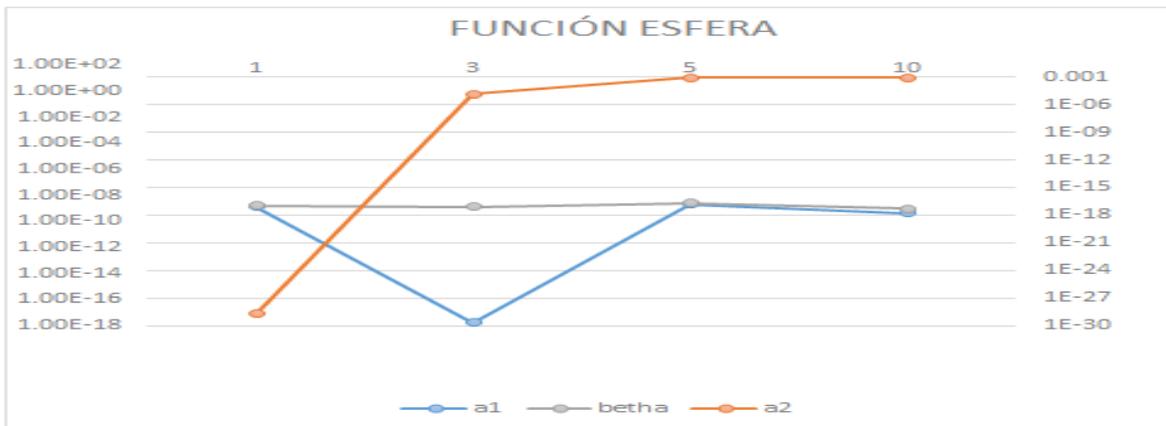


Figura 4.1 Impacto de parámetros función benchmark de la esfera

En la Figura 4.1 podemos observar cómo se comporta la función benchmark de la esfera, donde resalta el parámetro a_2 en el impacto del rendimiento del algoritmo de Mayfly

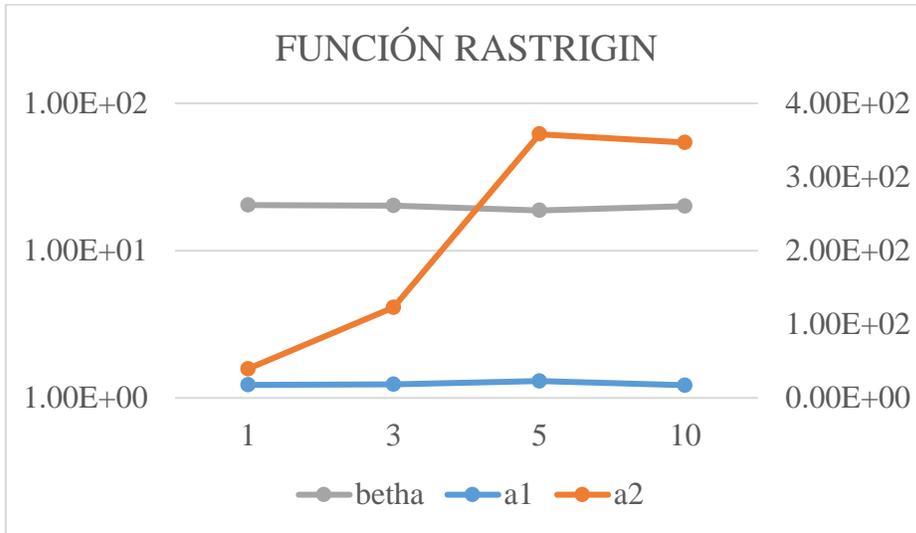


Figura 4.2 Impacto de parámetros función benchmark rastrigin

En la Figura 4.2 analizamos el impacto del algoritmo de Mayfly en la función de benchmark Rastrigin, aquí observamos que a_1 representa un mayor impacto al variarlo.

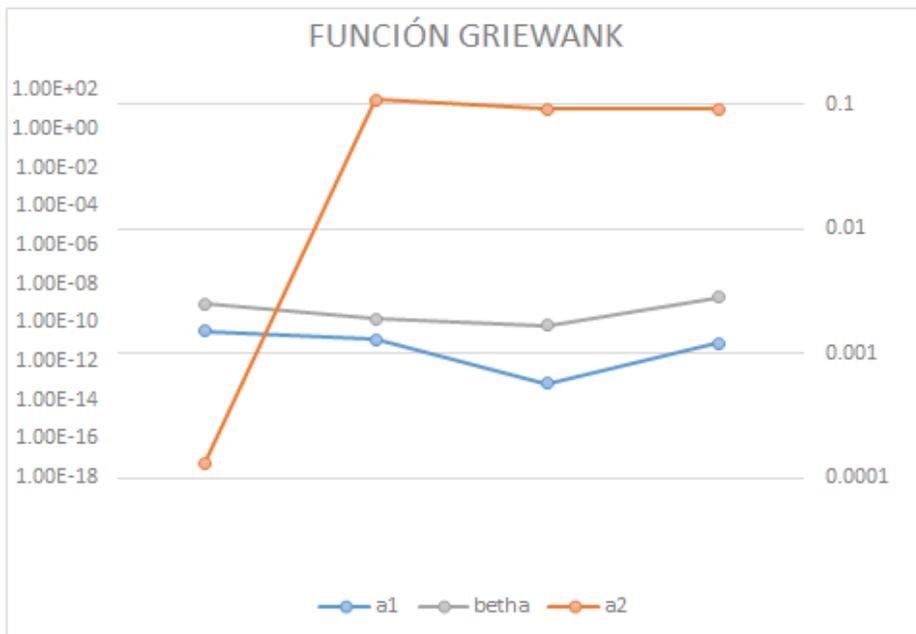


Figura 4.3 Impacto de parámetros función benchmark Griewank

En la Figura 4.3 analizamos el impacto del algoritmo de Mayfly en la función de benchmark Griewank, donde a_1 y a_2 representan un mayor impacto al variarlos.

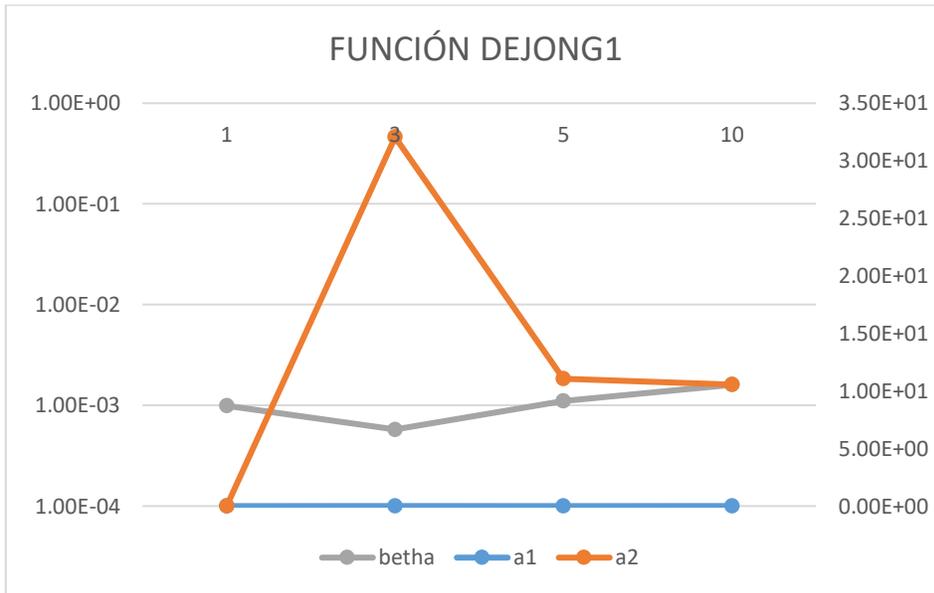


Figura 4.4 Impacto de parámetros función benchmark dejong1

En la Figura 4.4 analizamos el impacto del algoritmo de Mayfly en la función benchmark dejong1, donde a_2 representa claramente un mayor impacto al variarlo

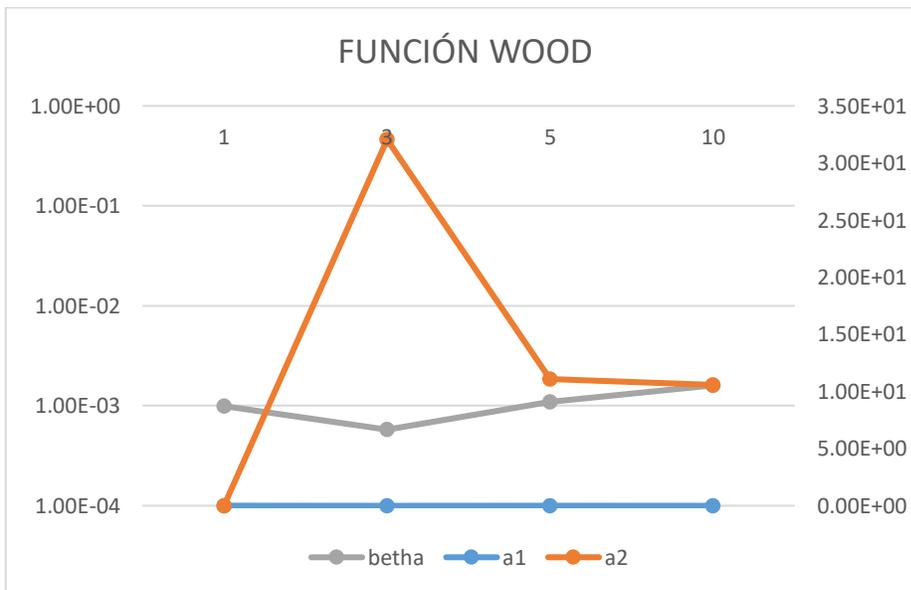


Figura 4.5 Impacto de parámetros función benchmark Wood

En la Figura 4.5 analizamos el impacto del algoritmo de Mayfly en la función de benchmark Wood, donde a_2 representa un mayor impacto al variarlo.

4.2 Propuesta de adaptación difusa.

En el estudio de impacto de parámetros se determinó que el parámetro beta no representa una variación importante en el desempeño del algoritmo de Mayfly, por lo tanto, se ha descartado de la aplicación de adaptación difusa, sin embargo, los parámetros a_1 y a_2 si presentaron un cambio brusco en el desempeño del algoritmo Mayfly, por tal motivo serán elegidos para ser parte de un adaptador difuso que será incluido

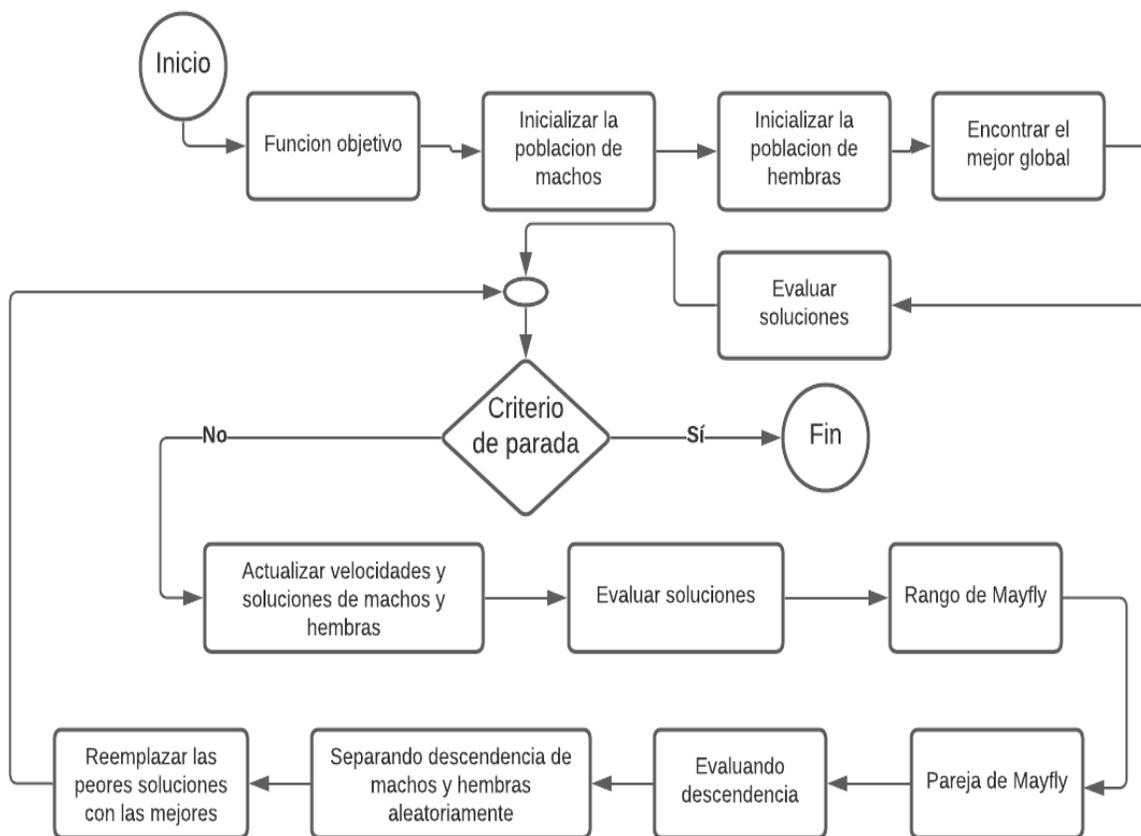


Figura 4.6 Diagrama de flujo del método original de Mayfly

En la Figura 4.6 podemos mirar el diagrama de flujo donde se explica a grandes rasgos la secuencia del algoritmo de Mayfly, donde comenzamos con la función objetivo y terminamos con el remplazo de las peores soluciones por las mejores con el fin de acercarnos a la función objetivo en cada iteración.

Una vez mostrado el algoritmo original de Mayfly, será necesario mostrarlo con el adaptador difuso para identificar en que parte del algoritmo será colocado.

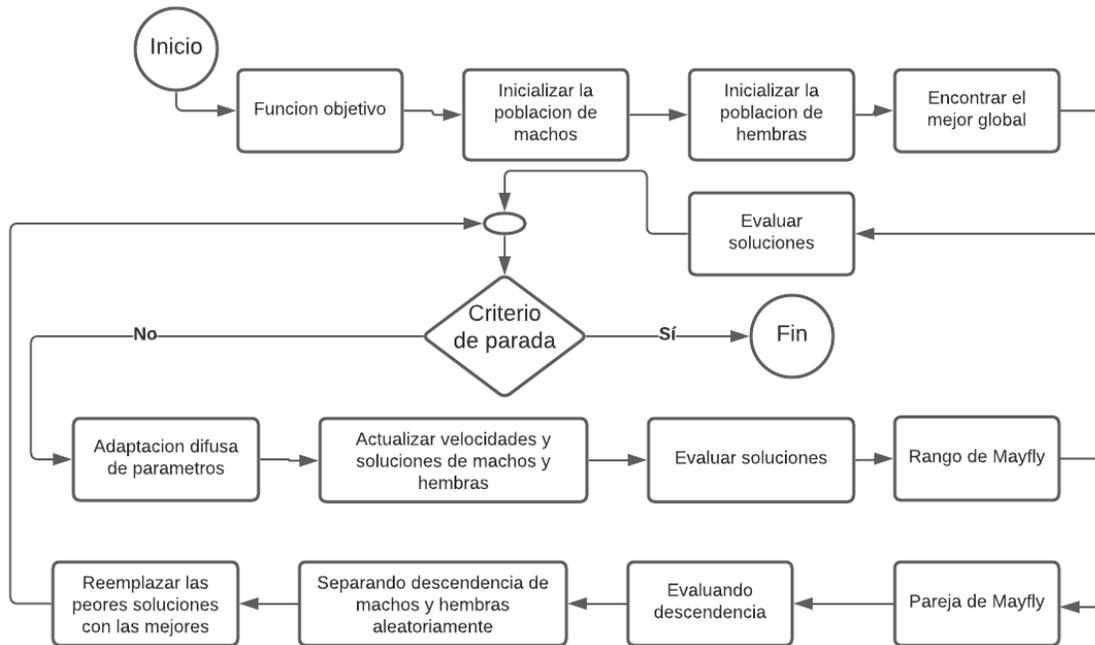


Figura 4.7 Diagrama de flujo del método modificado de Mayfly

En la Figura 4.7 podemos observar el diagrama de flujo del algoritmo de Mayfly con el adaptador difuso, donde se muestra con más exactitud en que parte será colocado.

4.3 Diseño del adaptador difuso

Primeramente, se realizó un adaptador difuso con el parámetro a_2 debido a que fue el parámetro que resulto más relevante en el desempeño del algoritmo de Mayfly, aunque estuvo muy cercano el parámetro a_1 .

4.3.1 Adaptador difuso de 1 entrada y 1 salida.

En este adaptador difuso tendrá como funciones de membresía las funciones triangulares debido a su simplicidad, por lo que inicialmente probaremos estas con un total de 5 reglas difusas.

- 1.- SI iteración es muy pequeña entonces a_2 es muy grande
- 2.- SI iteración es pequeña entonces a_2 es grande
- 3.- SI iteración es mediana entonces a_2 es mediana
- 4.- SI iteración es grande entonces a_2 es pequeña
- 5.- SI iteración es muy grande entonces a_2 es muy pequeña

Figura 4.8 Reglas difusas

Como podemos apreciar en la Figura 4.8 tenemos las reglas difusas iniciales en las cuales empezamos con a_2 con valores altos en las primeras iteraciones y conforme están van avanzando a_2 ira disminuyendo, en la primera regla difusa el algoritmo apenas va empezando, mientras que en la última regla este casi ha finalizado, eso hace que las reglas difusas ajusten el valor del parámetro a_2 al transcurrir de las iteraciones, y se acerque así a sus valores óptimos para dicho parámetro.

$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a \leq x \leq b \\ \frac{c - x}{c - b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (4.1)$$

En la ecuación 4.1 tenemos las funciones de membresía triangulares en la que los valores a , b y c serán dados por las tablas a continuación.

Tabla 4.11 Funciones de membresía de entrada en el adaptador difuso inicial

Función de membresía	a	b	C
IMP(Iteración muy pequeña)	0	0	0.25
IP(Iteración pequeña)	0	0.25	0.5
IM(Iteración media)	0.25	0.5	0.75
IG(Iteración grande)	0.5	0.75	1
IMG(Iteración muy grande)	0.75	1	1

Tabla 4.12 Funciones de membresía de salida en el adaptador difuso inicial

Función de membresía	A	b	C
A2MP(a_2 muy pequeña)	0	0	0.25
A2P(a_2 pequeña)	0	0.25	0.5
A2M(a_2 media)	0.25	0.5	0.75
A2G(a_2 grande)	0.5	0.75	1
A2MG(a_2 Muy grande)	0.75	1	1

Una vez realizadas las pruebas con este adaptador difuso, continuamos aplicándolo al parámetro a_1 el cual fue identificado como parte de los parámetros relevantes en el desempeño del algoritmo de Mayfly y ver cuál de los 2 representa una mayor mejoría en los resultados con las funciones benchmark, para lo cual se hicieron experimentos con 10 de las funciones principales y a si mismo se compararon con los resultados previos en la etapa de estudio del impacto de los parámetros.

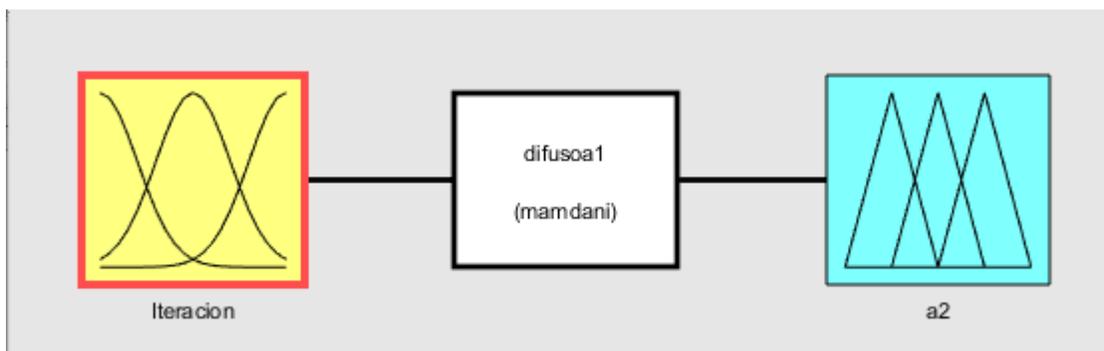
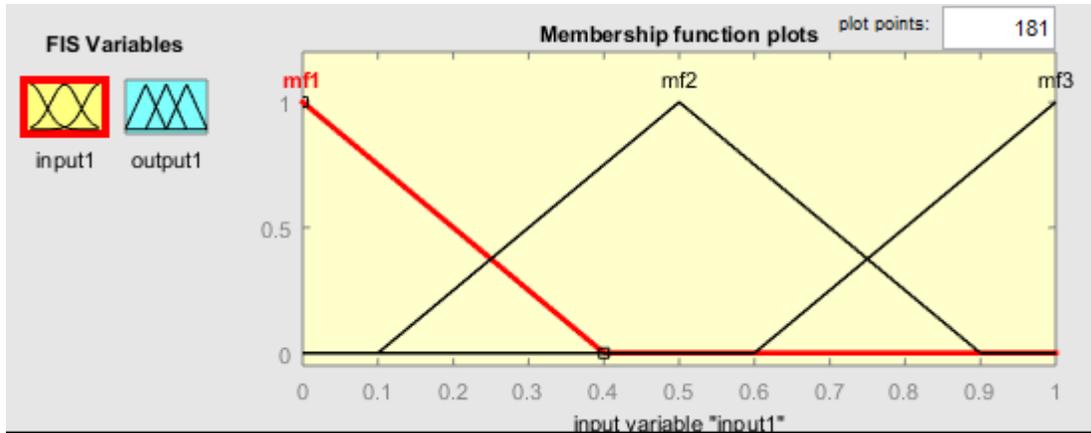
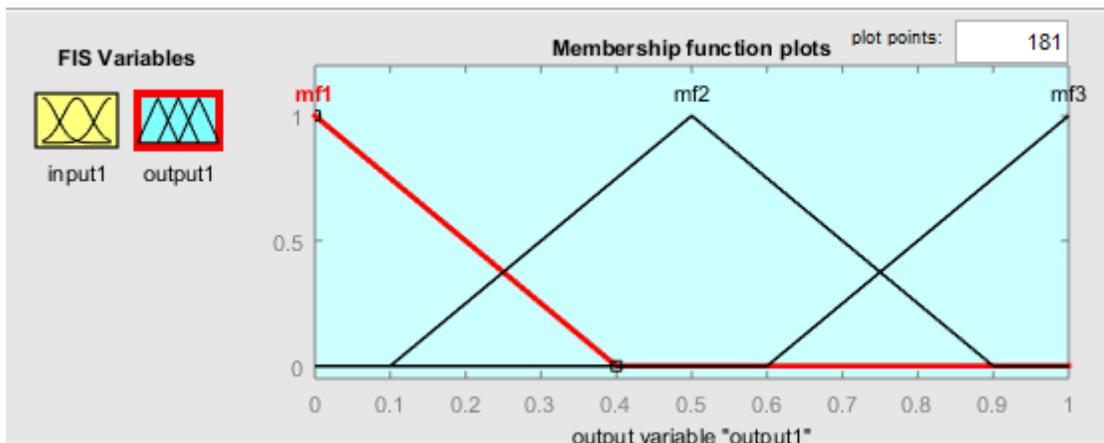


Figura 4.9 Adaptador difuso inicial

En la Figura 4.9 tenemos el adaptador difuso inicial el cual consistió de una variable de entrada y una de salida, en la variable de entrada solo tomamos en cuenta las iteraciones.

**Figura 4.10** Funciones de membresía triangulares iniciales de entrada

En la Figura 4.10 tenemos las funciones de membresía utilizadas inicialmente para la entrada de nuestro adaptador difuso, en este caso usamos 3 funciones de membresía triangulares ya que al ser el primer adaptador comenzamos con lo más simple.

**Figura 4.11** Funciones de membresía triangulares iniciales de salida

En la Figura 4.11 podemos apreciar las funciones de membresía que fueron utilizadas en las pruebas iniciales para la salida de este primer adaptador difuso, al igual que en la entrada aquí se usaron 3 funciones de membresía.

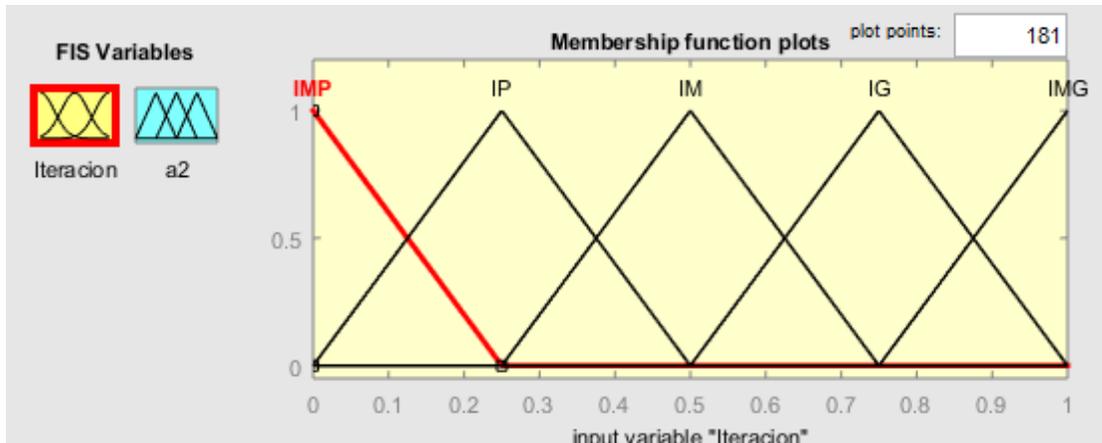


Figura 4.12 Funciones de membresía triangular de entrada para el adaptador difuso 1

En la Figura 4.12 tenemos las funciones de membresía de entrada que usamos finalmente para el adaptador difuso 1 el cual será comparado más adelante con el adaptador difuso final el cual empleamos para hacer la mejora del algoritmo de Mayfly, en este caso son 5 funciones de membresía haciendo más robusto el adaptador difuso.

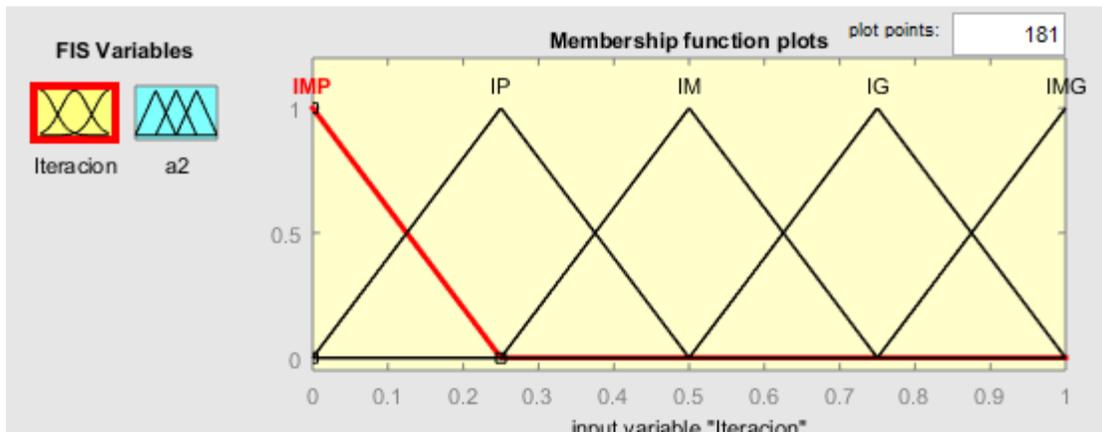


Figura 4.13 Funciones de membresía triangular de entrada para el adaptador difuso 1

En la Figura 4.13 mostramos las funciones de membresía que fueron utilizadas para la salida del adaptador difuso, mismas que contarán con un total de 5 para de esta manera darle más diversidad a la salida del adaptador.

Para comprobar si el parámetro a_1 o a_2 afecta más en particular a nuestro algoritmo modificado realizamos una comparación con los resultados obtenidos del adaptador difuso inicial aplicado para ambos parámetros por separado. A continuación, presentaremos una tabla con los resultados resumidos de los experimentos realizados con ambos adaptadores difusos iniciales.

Tabla 4.13 Comparación de resultados con adaptador difuso para a_2 y a_1

Función benchmark	Parámetro Beta /50 dimensiones			
	Difuso para a_2		Difuso para a_1	
	\bar{x}	S	\bar{x}	S
Esfera	2.1415×10^{-16}	7.7240×10^{-17}	8.3797×10^{-17}	5.2048×10^{-17}
Rastrigin	2.2920×10^1	8.9131×10^0	2.2685×10^1	8.5655×10^0
Ackley	3.0944×10^0	5.3400×10^{-2}	3.1116×10^0	3.4500×10^{-2}
Dejong1	1.9221×10^{-18}	2.9649×10^{-18}	8.2611×10^{-17}	4.5886×10^{-17}
Griewank	1.600×10^{-3}	3.6000×10^{-3}	1.700×10^{-3}	3.600×10^{-3}
Schwefel	2.0755×10^4	4.0579×10^0	2.0754×10^4	2.8595×10^0
Styblinski tank	-1.7482×10^3	4.1381×10^1	-1.7465×10^3	4.5256×10^1
Wood	1.074×10^{-1}	5.8840×10^{-1}	4.6900×10^{-2}	1.7930×10^{-1}
HYPHER-ELLIPSOID	6.5445×10^{-17}	1.0540×10^{-16}	2.5736×10^{-15}	2.1093×10^{-15}
Powell	5.1923×10^{-16}	1.2900×10^{-15}	9.2123×10^{-17}	5.1465×10^{-17}

Como podemos apreciar en la tabla 4.13 se compararon los resultados de ambos adaptadores difusos iniciales, donde se puede apreciar que no hubo una diferencia considerable, sin embargo, se puede notar que usando el parámetro a_2 se logran mejores resultados.

4.3.2 Adaptador difuso de 2 entradas y 2 salidas.

Al notar que los resultados del difuso inicial no mejoraron significativamente el desempeño del algoritmo de Mayfly ni siquiera al aplicarle reglas difusas trapezoidales se optó por hacer un adaptador difuso de 2 entradas y 2 salidas, y a su vez se utilizaron los parámetros a_1 y a_2 en conjunto para mejorar la efectividad del mismo, ya que individualmente no se lograba diferencia entre un parámetro y otro.

A continuación, se mostrará el diseño final de nuestro adaptador difuso el cual tendrá como variables de entrada la iteración y la diversidad.

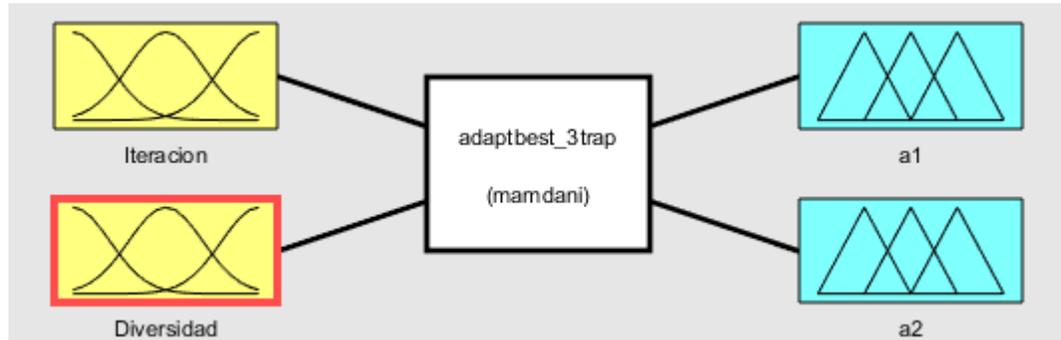


Figura 4.14 Adaptador difuso final

En la Figura 4.14 se puede apreciar de manera general el diseño del adaptador difuso final donde contamos con 2 entradas y 2 salidas respectivamente para controlar los parámetros a_1 y a_2

- 1.- SI iteración es baja y diversidad es baja entonces a_1 es baja y a_2 es alta.
- 2.- SI iteración es baja y diversidad es media entonces a_1 es media baja y a_2 es media alta.
- 3.- SI iteración es baja y diversidad es alta entonces a_1 es media y a_2 es media.
- 4.- SI iteración es media y diversidad es baja entonces a_1 es media baja y a_2 es media alta.
- 5.- SI iteración es media y diversidad es media entonces a_1 es media y a_2 es media.
- 6.- SI iteración es media y diversidad es alta entonces a_1 es media alta y a_2 es media baja.
- 7.- SI iteración es alta y diversidad es baja entonces a_1 es media y a_2 es media.
- 8.- SI iteración es alta y diversidad es media entonces a_1 es media alta y a_2 es media baja.
- 9.- SI iteración es alta y diversidad es alta entonces a_1 es alta y a_2 es baja.

Figura 4.15 Adaptador difuso final

En la Figura 4.15 tenemos las reglas difusas las cuales fueron basadas en el trabajo [12] en estas reglas difusas, favorecemos la exploración al principio del algoritmo, sin embargo conforme vayan avanzando las iteraciones nos enfocaremos más en la explotación, cabe resaltar que se utilizaron tanto las iteraciones como la diversidad de tal manera, que cuando más avanzaran las iteraciones más se acercara al valor óptimo para cada parámetro sin embargo al combinarlo con la diversidad podemos salir de mínimos locales de una mejor manera y estas reglas fueron aplicadas a los parámetros a_1 y a_2 con el objetivo de mejorar el desempeño ya que con la ayuda del cálculo de la diversidad podremos salir de mínimos locales.

En este adaptador difuso probamos inicialmente con las funciones triangulares, posteriormente las trapezoidales y al notar una mejoría, optamos por utilizar estas funciones de membresía

$$f(x; a, b, c, d) = \begin{cases} 0, & \text{si } (x < a) \text{ o } (x > d) \\ \frac{x - a}{b - a}, & \text{si } a \leq x \leq b \\ 1, & \text{si } b \leq x \leq c \\ \frac{d - x}{d - c}, & \text{si } c \leq x \leq d \end{cases} \quad (4.2)$$

En la ecuación 4.2 se muestran las funciones de membresía trapezoidales, en las que los valores a, b y c d serán dados por las tablas a continuación.

Tabla 4.14 Funciones de membresía de entrada en el adaptador difuso de 2 entradas y 2 salidas

Función de membresía	a	B	C	d
IB (Iteración baja)	0	0	0.1	0.33
IM (Iteración media)	0.25	0.42	0.58	0.75
IA (Iteración alta)	0.67	0.9	1	1
DB (Diversidad baja)	0	0	0.1	0.33
DM (Diversidad media)	0.25	0.42	0.58	0.75
DA (Diversidad alta)	0.67	0.9	1	1

Tabla 4.15 Funciones de membresía de salida en el adaptador difuso de 2 entradas y 2 salidas.

Función de membresía	a	b	c	d
A1B(a ₁ baja)	1.094	1.241	1.78	1.427
A1MB(a ₁ media baja)	1.278	1.399	1.485	1.611
A1M(a ₁ media)	1.475	1.578	1.659	1.779
A1MA(a ₁ media alta)	1.648	1.763	1.844	1.982
A1A(a ₁ alta)	1.833	1.982	2.018	2.167
A2B(a ₂ baja)	0.42	0.58	0.62	0.7803
A2MB(a ₂ media baja)	0.62	0.7803	0.8203	0.9797
A2M(a ₂ media)	0.85	0.9797	1.02	1.15
A2MA(a ₂ media alta)	1.02	1.18	1.22	1.38
A2A(a ₂ alta)	1.22	1.38	1.42	1.58

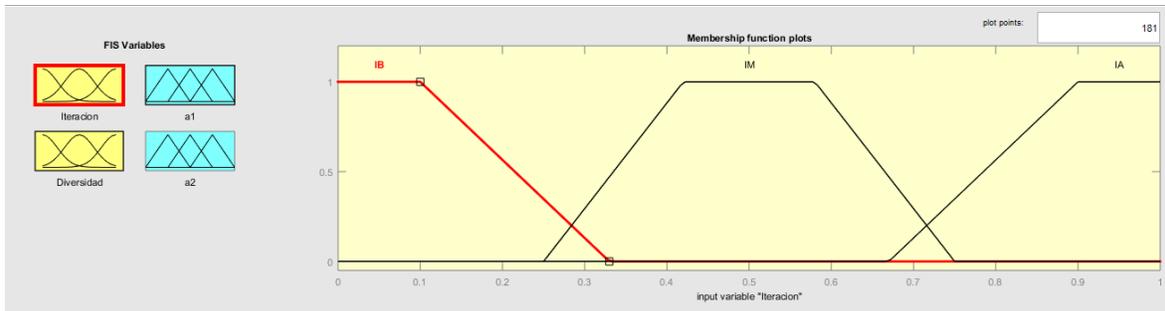


Figura 4.16 Adaptador difuso final 2 variables de entrada y 2 de salida

En la Figura 4.16 tenemos el modelo general del adaptador difuso final, este se compone por 2 variables de entrada y 2 de salida, al principio se utilizaron funciones de membrecía triangulares sin embargo resultaron dar mejores resultados las trapezoidales, podemos apreciar que contamos con 3 funciones de membrecía como entrada en ambos parámetros, la diversidad fue calculada a través de la siguiente formula:

$$d = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_s} (x_{ij}(t) - \bar{X}_j(t))^2} \tag{4.3}$$

Como podemos apreciar en la ecuación 4.3 haremos el cálculo de la diversidad la cual nos servirá para salir de mínimos locales en caso de que estos se presenten, para conocer a detalle cómo se desarrolla la diversidad (véase anexo 3).

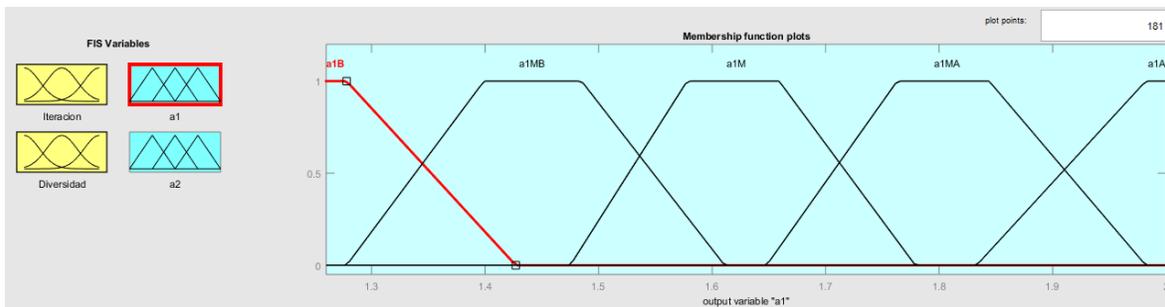


Figura 4.17 Funciones de membrecía de salida para el adaptador difuso final

En la Figura 4.17 tenemos las salidas del adaptador difuso final, en este caso usamos 5 funciones de membrecía al igual que en la entrada y con el fin de conseguir un mejor desempeño establecimos los valores de salida para los parámetros a₁ y a₂ en el rango encontrado en el estudio del impacto de los parámetros.

Tabla 4.16 Resultados de los adaptadores difusos

Función benchmark	50 dimensiones				Resultados autor	
	Adaptador difuso Inicial		Adaptador difuso final			
	\bar{x}	S	\bar{x}	s	\bar{x}	S
Esfera	2.1415×10^{-16}	$7.7240 \times 10^{-}$	8.3455×10^{-17}	5.1033×10^{-17}	1.1777×10^{-7}	4.5203×10^{-7}
Rastrigin	1.7254×10^1	5.5900×10^0	1.1270×10^1	3.7928×10^0	1.1900×10^1	3.8200×10^0
Ackley	3.1267×10^0	$3.0100 \times 10^{-}$	3.1224×10^0	2.9100×10^{-2}	0.0000×10^0	0.0000×10^0
Griewank	1.1000×10^{-3}	$4.0000 \times 10^{-}$	9.0247×10^{-4}	3.8000×10^{-3}	4.1400×10^{-3}	1.2900×10^{-2}
Schwefel	2.0754×10^4	3.3438×10^0	2.0755×10^4	2.9589×10^0	3.8800×10^0	8.8300×10^{-1}
Powell	1.8052×10^{-15}	$5.7657 \times 10^{-}$	3.9987×10^{-127}	2.1902×10^{-122}	5.2800×10^{-49}	1.6500×10^{-48}
Rosenbrock	5.6907×10^1	2.5495×10^1	5.52105×10^1	2.3000×10^1	6.7703×10^1	3.9877×10^1
Alpine	3.6000×10^{-3}	$3.0000 \times 10^{-}$	7.9186×10^{-4}	2.1000×10^{-3}	3.3465×10^{-8}	1.6242×10^{-7}
zakharov	5.9987×10^1	1.0816×10^2	6.8769×10^1	1.3059×10^2	1.7130×10^{-1}	8.3733×10^{-2}
Sum Squares	2.7503×10^{-10}	$3.8725 \times 10^{-}$	3.6940×10^{-15}	5.6091×10^{-15}	7.3923×10^{-6}	2.5797×10^{-5}

4.4 Prueba estadística

En esta sección se realizaron pruebas estadísticas a los resultados obtenidos por el difuso final, con el fin de saber si tenemos evidencia significativa para poder afirmar que la modificación hecha al algoritmo de Mayfly hace que su rendimiento sea superior, la prueba estadística usada fue la prueba de hipótesis con 2 muestras para las cuales se utilizó un 90% de confianza y la siguiente ecuación.

$$z = \frac{(\bar{X}_1 - \bar{X}_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \tag{4.4}$$

Como podremos apreciar en la ecuación 10 tenemos que conocer las medias y desviaciones estándar, así como el tamaño de la muestra para poder realizar el cálculo, también es importante recordar la condición de $n > 30$.

4.4.1 Prueba de hipótesis función de la esfera.

Se realiza la prueba estadística para la función esfera con el fin de descubrir si hay evidencia significativa con los resultados obtenidos del Mayfly con adaptación difusa comparado con el Mayfly original.

$h_0: m_1 \geq m_2$ y $h_a: m_1 < m_2$ (afirmación).

$$\bar{X}_1 = 8.3455 \times 10^{-17}$$

$$\bar{X}_2 = 1.1777 \times 10^{-7}$$

$$S_1 = 5.1033 \times 10^{-17}$$

$$S_2 = 4.5203 \times 10^{-17}$$

$$z = \frac{(7.7240 \times 10^{-7} - 1.1777 \times 10^{-7})}{\sqrt{\frac{(5.1033 \times 10^{-17})^2}{30} + \frac{(4.5203 \times 10^{-7})^2}{30}}}$$

$$z = \frac{-1.1777 \times 10^{-7}}{\sqrt{\frac{(5.1033 \times 10^{-17})^2}{30} + \frac{(4.5203 \times 10^{-7})^2}{30}}}$$

$$z = -1.4270$$

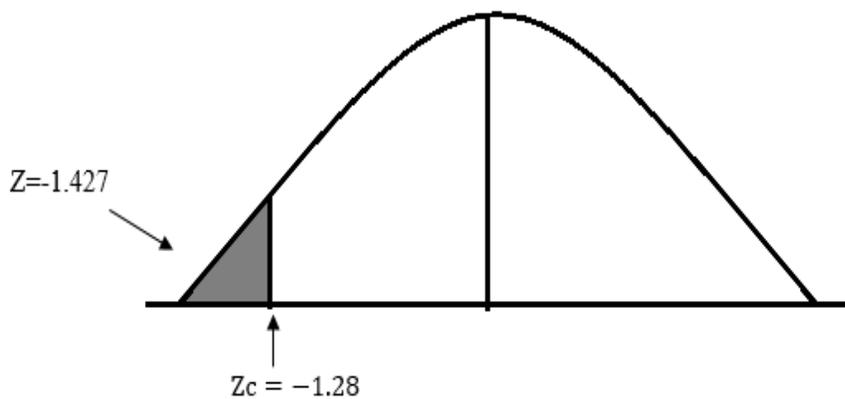


Figura 4.18 Prueba cola izquierda para función benchmark de la esfera

Como se puede apreciar en la Figura 4.18 hemos realizado una prueba estadística de cola izquierda en la cual la región de rechazo se encuentra en $Z < -1.28$, en este caso la z calculada resulto -1.4270 por lo que se rechaza la h_0 (hipótesis nula) y entonces concluimos que tenemos suficiente evidencia para apoyar la afirmación(h_a) la cual nos dice que $m_1 < m_2$.

4.4.2 Prueba de hipótesis función de rastrigin.

Se realiza la prueba estadística para la función rastrigin con el fin de descubrir si hay evidencia significativa con los resultados obtenidos del Mayfly con adaptación difusa comparado con el Mayfly original.

$h_0: m_1 \geq m_2$ y $h_a: m_1 < m_2$ (afirmación).

$$\bar{X}_1 = 1.1270 \times 10^1$$

$$\bar{X}_2 = 1.1900 \times 10^1$$

$$S_1 = 3.7928 \times 10^0$$

$$S_2 = 3.8200 \times 10^0$$

$$z = \frac{(1.1270 \times 10^1 - 1.1900 \times 10^1)}{\sqrt{\frac{(3.7928 \times 10^0)^2}{30} + \frac{(3.8200 \times 10^0)^2}{30}}}$$

$$z = \frac{-0.6300}{\sqrt{0.4795} + \sqrt{0.4864}}$$

$$z = \frac{-0.6300}{0.9828}$$

$$z = -0.6410$$

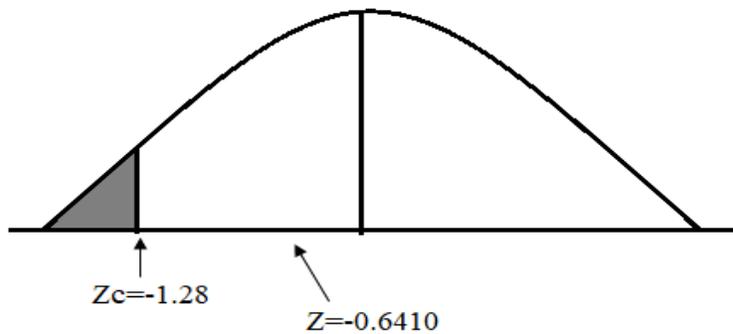


Figura 4.19 Prueba cola izquierda para función benchmark de rastrigin

Como se puede apreciar en la Figura 4.19 hemos realizado una prueba estadística de cola izquierda en la cual la región de rechazo se encuentra en $Z < -1.28$, en este caso la z calculada resultó -0.6410 por lo que fallamos rechazar la h_0 (hipótesis nula) y entonces concluimos que no tenemos suficiente evidencia para apoyar la afirmación (h_a) la cual nos dice que $m_1 < m_2$.

4.4.3 Prueba de hipótesis función de Griewank.

Se realiza la prueba estadística para la función Griewank con el fin de descubrir si hay evidencia significativa con los resultados obtenidos del Mayfly con adaptación difusa comparado con el Mayfly original.

$h_0: m_1 \geq m_2$ y $h_a: m_1 < m_2$ (afirmación).

$$\bar{x}_1 = 9.0247 \times 10^{-4}$$

$$\bar{x}_2 = 4.1400 \times 10^{-3}$$

$$s_1 = 3.8000 \times 10^{-3}$$

$$s_2 = 1.2900 \times 10^{-2}$$

$$z = \frac{(9.0247 \times 10^{-4} - 4.1400 \times 10^{-3})}{\sqrt{\frac{(3.8000 \times 10^{-3})^2}{30} + \frac{(1.2900 \times 10^{-2})^2}{30}}}$$

$$z = \frac{(-0.0032)}{\sqrt{4.8133 \times 10^{-7} + 5.5470 \times 10^{-6}}}$$

$$z = \frac{-0.0032}{0.0025}$$

$$z = -1.3186$$

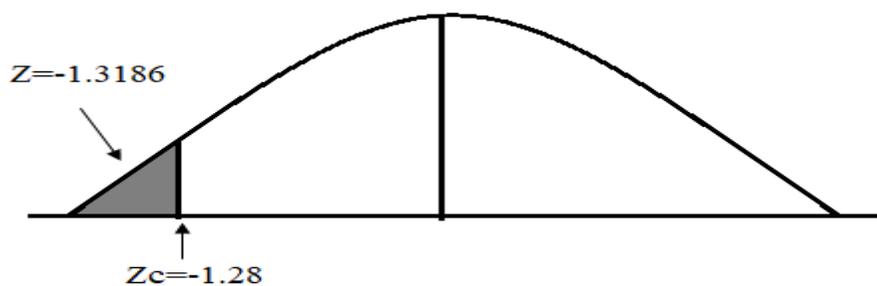


Figura 4.20 Prueba cola izquierda para función benchmark de Griewank

Como se puede apreciar en la Figura 4.20 hemos realizado una prueba estadística de cola izquierda en la cual la región de rechazo se encuentra en $Z < -1.28$, en este caso la z calculada resulto -1.3186 por lo que rechazamos la h_0 (hipótesis nula) y entonces concluimos que tenemos suficiente evidencia para apoyar la afirmación (h_a) la cual nos dice que $m_1 < m_2$.

4.4.4 Prueba de hipótesis función de Powell.

Se realiza la prueba estadística para la función Powell con el fin de descubrir si hay evidencia significativa con los resultados obtenidos del Mayfly con adaptación difusa comparado con el Mayfly original.

$h_0: m_1 \geq m_2$ y $h_a: m_1 < m_2$ (afirmación).

$$\bar{X}_1 = 3.9987 \times 10^{-127}$$

$$\bar{X}_2 = 5.2800 \times 10^{-49}$$

$$S_1 = 2.1902 \times 10^{-122}$$

$$S_2 = 1.6500 \times 10^{-48}$$

$$z = \frac{(3.9987 \times 10^{-127} - 5.2800 \times 10^{-49})}{\sqrt{\frac{(2.1902 \times 10^{-122})^2}{30} + \frac{(1.6500 \times 10^{-48})^2}{30}}}$$

$$z = \frac{(5.2800 \times 10^{-49})}{\sqrt{1.5990 \times 10^{-245} + 9.0750 \times 10^{-98}}}$$

$$z = \frac{5.2800 \times 10^{-49}}{3.0125 \times 10^{-49}}$$

$$z = -1.7527$$

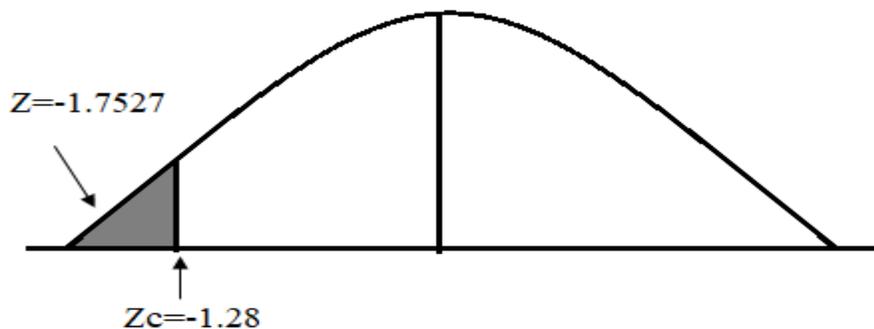


Figura 4.21 Prueba cola izquierda para función benchmark de Powell

Como se puede apreciar en la Figura 4.21 hemos realizado una prueba estadística de cola izquierda en la cual la región de rechazo se encuentra en $Z < -1.28$, en este caso la z calculada resulto -1.7527 por lo que rechazamos la h_0 (hipótesis nula) y entonces concluimos que tenemos suficiente evidencia para apoyar la afirmación(h_a) la cual nos dice que $m_1 < m_2$

4.4.5 Prueba de hipótesis función de Sum Squares.

Se realiza la prueba estadística para la función sum squares con el fin de descubrir si hay evidencia significativa con los resultados obtenidos del Mayfly con adaptación difusa comparado con el Mayfly original.

$h_0: m_1 \geq m_2$ y $h_a: m_1 < m_2$ (afirmación).

$$\bar{X}_1 = 3.6940 \times 10^{-15}$$

$$\bar{X}_2 = 7.3923 \times 10^{-6}$$

$$S_1 = 5.6091 \times 10^{-15}$$

$$S_2 = 2.5797 \times 10^{-5}$$

$$z = \frac{(3.6940 \times 10^{-15} - 7.3923 \times 10^{-6})}{\sqrt{\frac{(5.6091 \times 10^{-15})^2}{30} + \frac{(2.5797 \times 10^{-5})^2}{30}}}$$

$$z = \frac{(-7.3923 \times 10^{-6})}{\sqrt{1.0487 \times 10^{-30} + 2.2183 \times 10^{-11}}}$$

$$z = \frac{-7.3923 \times 10^{-6}}{4.7099 \times 10^{-6}}$$

$$z = -1.5695$$

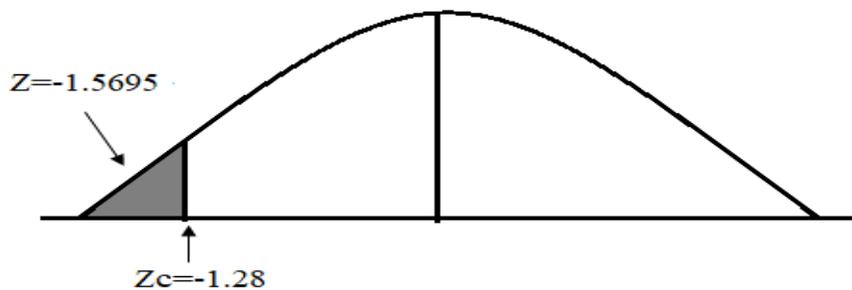


Figura 4.22 Prueba cola izquierda para función benchmark de Sum Squares

Como se puede apreciar en la Figura 4.22 hemos realizado una prueba estadística de cola izquierda en la cual la región de rechazo se encuentra en $Z < -1.28$, en este caso la z calculada resulto -1.5695 por lo que rechazamos la h_0 (hipótesis nula) y entonces concluimos que tenemos suficiente evidencia para apoyar la afirmación(h_a) la cual nos dice que $m_1 < m_2$.

4.4.6 Prueba de hipótesis resultados.

A continuación, se mostrará una tabla que contendrá un concentrado de los resultados de las pruebas de hipótesis aplicadas a las diferentes funciones benchmark a un nivel de confianza del 90%. Con una $Z_c = -1.28$

Tabla 4.17 Resultados prueba de hipótesis

Función benchmark	50 dimensiones				Resultados prueba estadística	
	Adaptador difuso final		Método original Mayfly			
	\bar{x}	s	\bar{x}	S	Z	S
Esfera	8.3455×10^{-7}	5.1033×10^{-17}	1.1777×10^{-7}	4.5203×10^{-7}	-1.4270	Pasa la prueba
Rastrigin	1.1270×10^1	3.7928×10^0	1.1900×10^1	3.8200×10^0	-0.6410	No pasa la prueba
Ackley	3.1224×10^0	2.9100×10^{-2}	0.0000×10^0	0.0000×10^0	0	No pasa la prueba
Griewank	9.0247×10^{-4}	3.8000×10^{-3}	4.1400×10^{-3}	1.2900×10^{-2}	-1.3186	Pasa la prueba
Schwefel	2.0755×10^4	2.9589×10^0	3.8800×10^0	8.8300×10^{-1}	0	No pasa la prueba
Powell	3.9987×10^{-7}	2.1902×10^{-122}	5.2800×10^{-49}	1.6500×10^{-48}	-1.7527	Pasa la prueba
Rosenbrock	5.52105×10^1	2.3000×10^1	6.7703×10^1	3.9877×10^1	-1.4764	Pasa la prueba
Alpine	7.9186×10^{-4}	2.1000×10^{-3}	3.3465×10^{-8}	1.6242×10^{-7}	2.0652	No pasa la prueba
Zakharov	6.8769×10^1	1.3059×10^2	1.7130×10^{-1}	8.3733×10^{-2}	2.8771	No pasa la prueba
Sum	3.6940×10^{-7}	5.6091×10^{-15}	7.3923×10^{-6}	2.5797×10^{-5}	-1.5696	Pasa la prueba

Como podemos observar en la tabla 4.4.6.1 tenemos los resultados de las pruebas estadísticas aplicadas en comparación con el autor de Mayfly para 50 dimensiones e iteraciones máximas de 2000, como resultados obtuvimos 5 pruebas con evidencia significativa de que la inclusión del adaptador difuso mejora el rendimiento del algoritmo de Mayfly en las funciones benchmark comúnmente usadas para este tipo de pruebas, de estas pruebas la más destacable fue la que se realizó con la función Powell ya que es la única que pasa la prueba estadística con un 95% de confianza.

4.5 Diseño de la red neuronal de Mackey Glass

Para diseñar la red neuronal pasamos por diversos pasos, primeramente, obtuvimos los datos de entrenamiento para la serie caótica de Mackey Glass, esto con ayuda de un generador de datos previamente programado en Matlab, como parámetros de entrada se utilizaron $\tau=17$, $x_0=1.2$ y generamos una muestra de 0 a 1200.

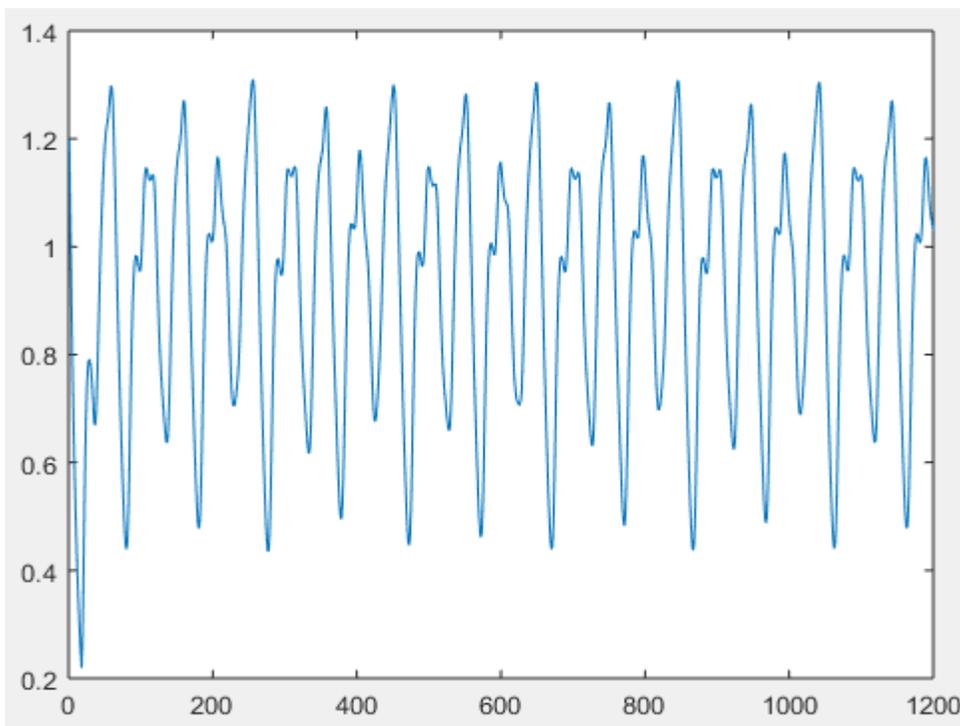


Figura 4.23 Serie de Mackey Glass

Como podemos ver en la Figura 4.23 tenemos la serie caótica de Mackey Glass con una muestra de 1200 datos de los cuales tomamos 800 para entrenar nuestra red neuronal, sin ningún retardo, por lo tanto, nuestra red solo será capaz de predecir algunos cuantos pasos delante de los datos no conocidos, sin embargo lo fundamental de esta red neuronal es extraer el error cuadrático medio para de esta manera determinar qué tan efectiva es y determinar cuántas neuronas requiere la capa oculta, dicha información será mandada al algoritmo de Mayfly original y al que hemos modificado con el adaptador difuso para verificar que tan

eficiente es el método y a su vez maximizar el rendimiento de la red neuronal previamente diseñada.

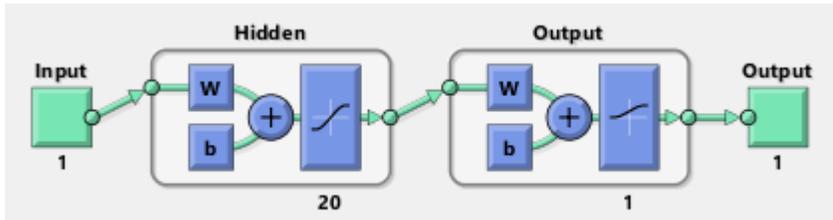


Figura 4.24 Arquitectura inicial de la red neuronal

En la Figura 4.24 podemos ver la arquitectura inicial para nuestra red neuronal la cual consistió en 3 capas de las cuales solo tendremos una entrada y una salida, en la capa oculta diseñamos inicialmente la red neuronal con 20 neuronas para ver su comportamiento y error cuadrático medio generado.

```

epocas = 400; %epocas maximas
% minimos = 100; %minimos locales posibles
factor = 0.1; %factor de aprendizaje
%%
salida=[datos entrenamiento'];
red=patternnet(neuronas,'trainlm');
red.trainParam.epochs=(epocas); %numero de epocas maximas
% red.trainParam.max fail =minimos; %minimos locales posibles
red.trainParam.min_grad=1e-29; %Error maximo
red.trainParam.mu=factor; %Factor de aprendizaje
red.trainParam.mu_dec=0.1; %Factor de aprendizaje decreciente
red.trainParam.mu_inc=10; %Factor de aprendizaje creciente
configure(red,t_entrada',salida');
[red,tr]=train(red,t_entrada',salida);

```

Figura 4.25 Configuración de la red neuronal

En la Figura 4.25 podemos apreciar la configuración utilizada para crear nuestra red neuronal, primeramente, configuramos las épocas con un máximo de 400, un factor aprendizaje de 0,1 el cual es usado normalmente por defecto al crear este tipo de redes neuronales, utilizamos el método de entrenamiento de Levenberg-Marquardt y un error máximo gradiente de 1e-29.

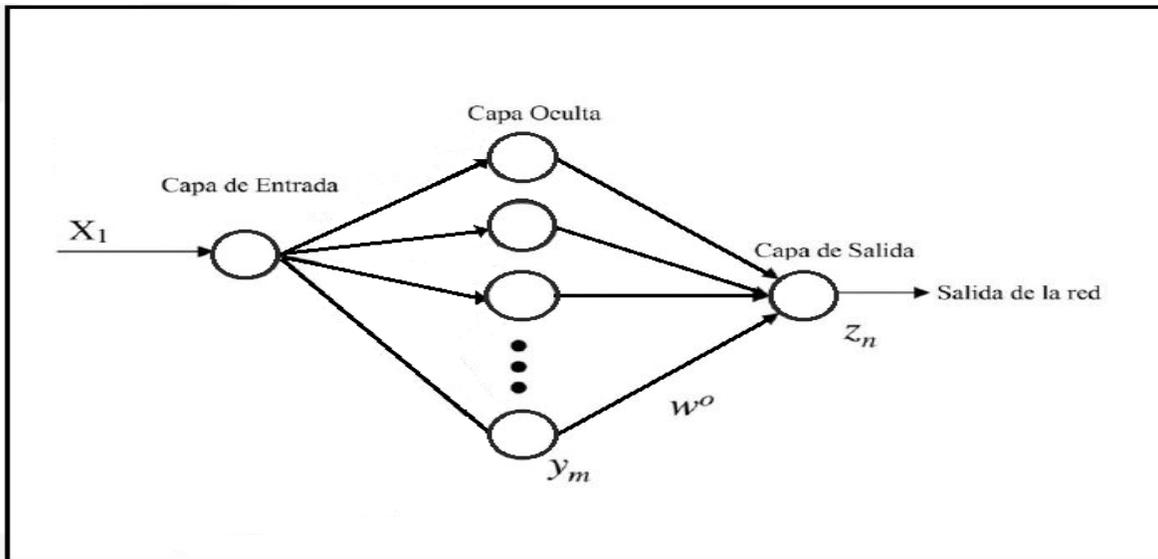


Figura 4.26 Arquitectura de red neuronal

Como se puede observar en la Figura 4.26 tenemos la arquitectura de la red neuronal, la cual consta de 3 capas, teniendo una capa de entrada, es decir, no generaremos retrasos en las entradas para nuestra serie de tiempo de Mackey Glass, posteriormente tendremos la capa oculta, la cual será la que optimizaremos para determinar cuál será el número ideal de neuronas y por último contamos con la capa de salida la cual nos dará el resultado de la red neuronal.

4.6 Optimización de la red neuronal.

Una vez diseñada nuestra red neuronal utilizamos el algoritmo de Mayfly el cual fue optimizado con adaptación difusa de parámetros para ser aplicado a la red neuronal que predecirá el comportamiento de la serie caótica de Mackey Glass, en este caso mejoraremos la arquitectura optimizando el número de neuronas de la capa oculta en nuestra red neuronal para así hacer una predicción más exacta.

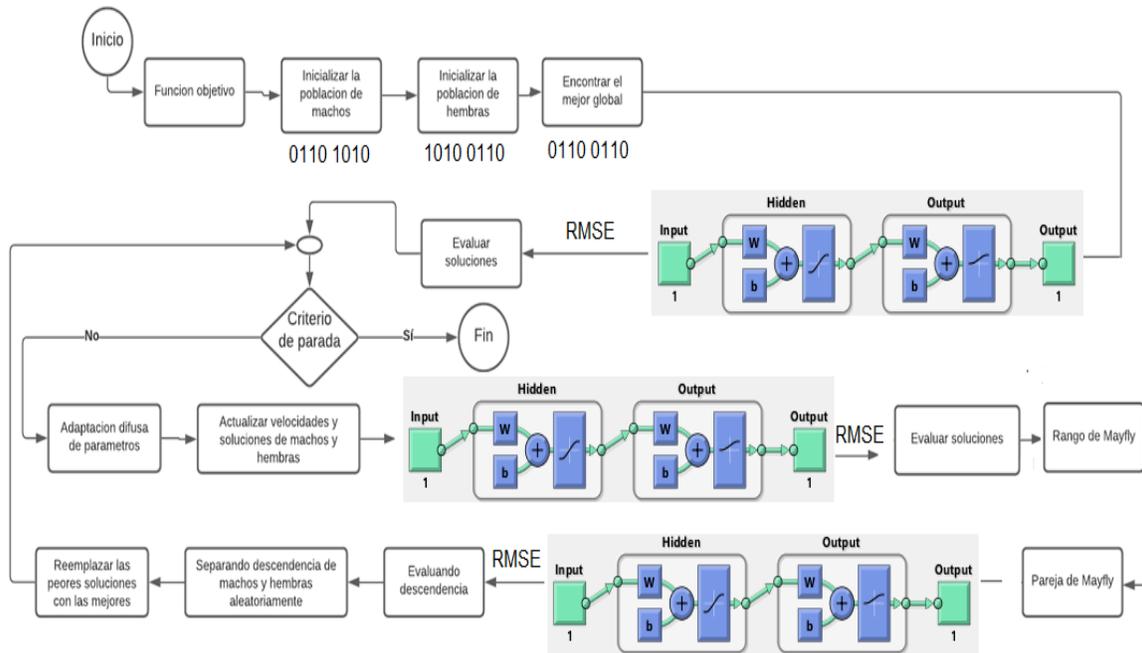


Figura 4.27 Optimización de la arquitectura de red neuronal

En la Figura 4.27 podemos ver como realizamos la optimización de la arquitectura de red neuronal para la serie de tiempo de Mackey Glass, de tal manera que primero determinamos el mejor global mandándole los parámetros iniciales a la red neuronal y posteriormente entrando al ciclo del algoritmo de Mayfly en donde se ira optimizando el número de neuronas de la capa oculta.

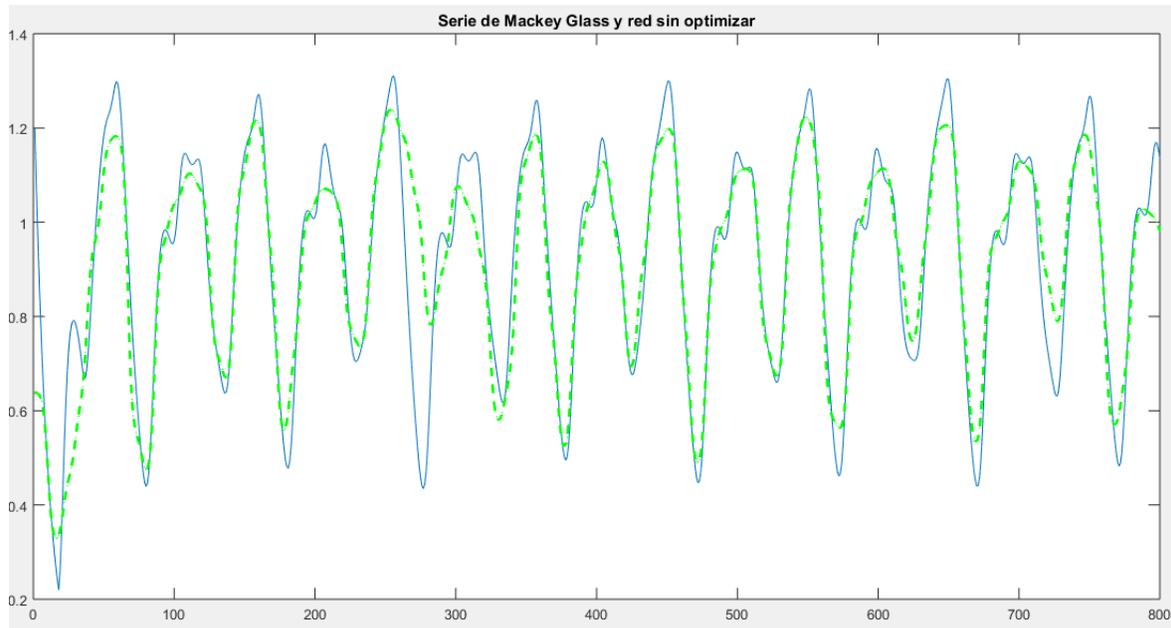


Figura 4.28 Pronostico de la red sin optimizar

En la Figura 4.28 tenemos 2 graficas, la de color azul es la que contiene la serie caótica de Mackey Glass con muestras de 0 a 800 datos, este es el rango más común para este tipo de pruebas y la de color verde es nuestra red neuronal sin optimizar, para la cual elegimos como parámetro inicial 80 neuronas en la capa oculta y cómo podemos apreciar no hace un seguimiento muy preciso ya que esta contiene un error cuadrático medio de 0.0936, por lo tanto en la búsqueda de reducir el error cuadrático medio, se le aplico el algoritmo de Mayfly, ya que para optimizar las funciones matemáticas benchmark resulto muy eficiente en la convergencia al mínimo global.

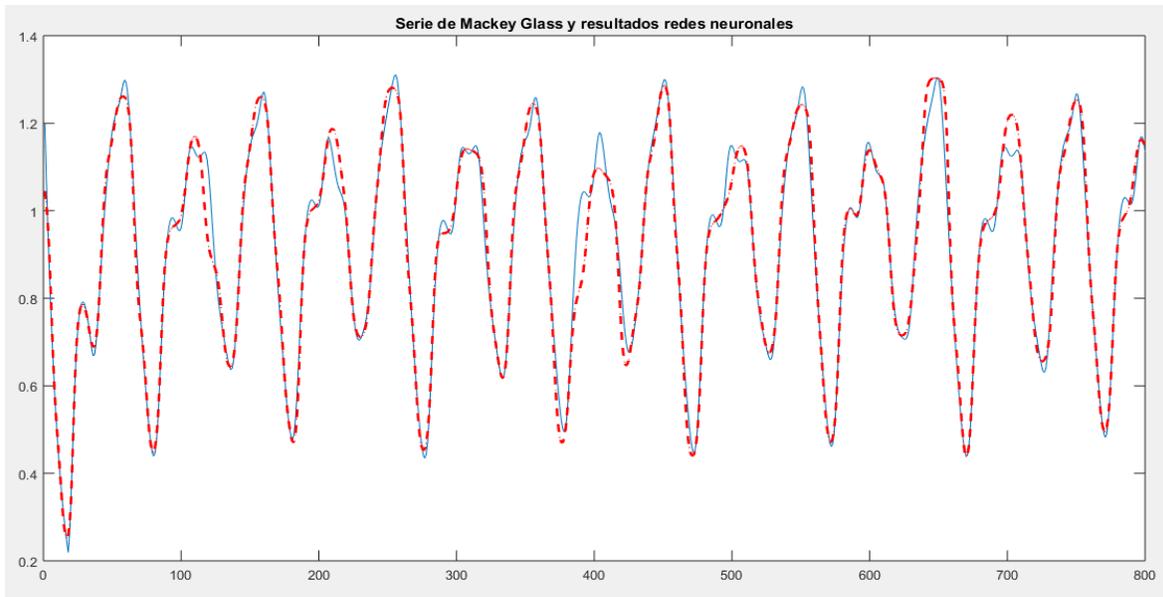


Figura 4.29 Pronostico de la red optimizada

En la Figura 4.29 podemos apreciar una comparación entre la serie caótica de Mackey Glass y nuestra red neuronal ya optimizada anteriormente, en dicho resultado de optimización obtuvimos que el número adecuado de neuronas en la capa oculta es de 73, ya que con ello el error cuadrático medio es de 0.0365.

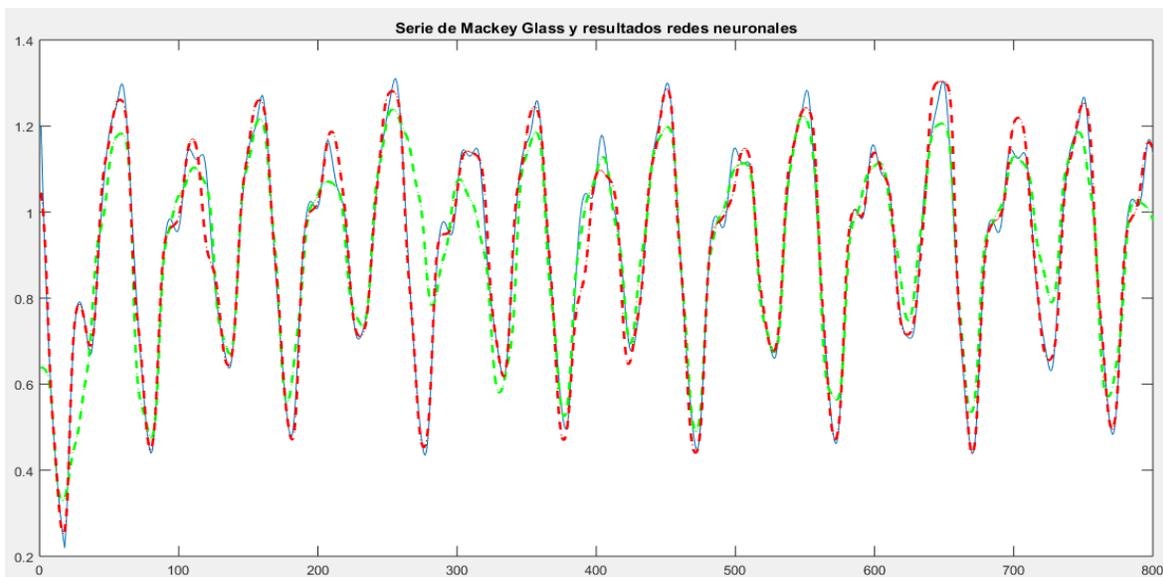


Figura 4.30 Comparación resultados de redes neuronales

En la Figura 4.30 se puede apreciar cómo se comportan las 2 redes neuronales y apreciamos que la red optimizada se acerca más a la serie caótica de Mackey Glass.

5. Conclusiones

El parámetro a_2 afecta principalmente al rendimiento del algoritmo mayfly, se pudo observar que su rendimiento aumenta en valores cercanos a 1, por lo que será uno de los parámetros a los que se aplicará la lógica difusa. El parámetro a_1 aumenta el rendimiento del algoritmo en valores entre 3 y 5, por lo que será el otro parámetro a tener en cuenta a pie de la aplicación de la lógica difusa.

En cuanto al parámetro β obtuvo un mejor resultado a un valor de 1, sin embargo su variación afecta muy poco al rendimiento del algoritmo, por lo que no se tiene en cuenta para la aplicación de lógica difusa. Los resultados mejoraron al aplicar un adaptador difuso con 2 parámetros de entrada y 2 de salida.

Al aplicar la adaptación difusa a los parámetros a_1 y a_2 en conjunto mejoro significativamente el rendimiento en las funciones matemáticas comparándonos con el autor del método original de Mayfly.

Se obtuvieron mejores resultados con las funciones trapezoidales en comparación con las triangulares.

Utilizar el algoritmo de Mayfly para optimizar la arquitectura de redes neuronales funciona bien, sin embargo, los resultados no mejoran mucho, por lo que podemos concluir que esta meta heurística no es recomendable para este tipo de optimización, debido a que el error cuadrático medio no pudo bajar de $e-3$ incluso usando el algoritmo de Mayfly modificado con el adaptador difuso.

Optimizar la arquitectura de una red neuronal es de un costo computacional elevado sin embargo el costo es menor que si lo hiciéramos con un método clásico de búsqueda, nos resulta complicado obtener un número adecuado de neuronas en la capa oculta ya que su complejidad es diferente a la optimización de funciones matemáticas que es en lo que se ha destacado principalmente este método.

Como trabajo futuro se puede optimizar el adaptador difuso por medio de un GA para maximizar el rendimiento del algoritmo de Mayfly al encontrar los valores óptimos de las funciones de membresía.

Referencias

- [1] J. Zhao and Z. Gao, "The negative mayfly optimization algorithm," *J. Phys. Conf. Ser.*, vol. 1693, no. 1, 2020, doi: 10.1088/1742-6596/1693/1/012098.
- [2] X. Guo, X. Yan, and K. Jermsittiparsert, "Using the modified mayfly algorithm for optimizing the component size and operation strategy of a high temperature PEMFC-powered CCHP," *Energy Reports*, vol. 7, pp. 1234–1245, 2021, doi: 10.1016/j.egy.2021.02.042.
- [3] Z. Gao, J. Zhao, S. R. Li, and Y. Hu, "The improved mayfly optimization algorithm," *J. Phys. Conf. Ser.*, vol. 1684, no. 1, 2020, doi: 10.1088/1742-6596/1684/1/012077.
- [4] Z. Gao, J. Zhao, S. R. Li, and Y. Hu, "The improved mayfly optimization algorithm with opposition based learning rules," *J. Phys. Conf. Ser.*, vol. 1693, no. 1, pp. 0–6, 2020, doi: 10.1088/1742-6596/1693/1/012117.
- [5] L. Chen, C. Xu, H. Song, and K. Jermsittiparsert, "Optimal sizing and siting of EVCS in the distribution system using metaheuristics: A case study," *Energy Reports*, vol. 7, pp. 208–217, 2021, doi: 10.1016/j.egy.2020.12.032.
- [6] A. Dey, S. Chattopadhyay, P. Singh, A. Ahmadian, M. Ferrara, and R. Sarkar, "A Hybrid Meta-Heuristic Feature Selection Method Using Golden Ratio and Equilibrium Optimization Algorithms for Speech Emotion Recognition," *IEEE Access*, vol. 8, pp. 200953–200970, 2020, doi: 10.1109/ACCESS.2020.3035531.
- [7] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, "Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3196–3206, Jun. 2013, doi: 10.1016/j.eswa.2012.12.033.
- [8] F. Valdez, O. Castillo, and P. Melin, "Bio-inspired algorithms and its applications for optimization in fuzzy clustering," *Algorithms*, vol. 14, no. 4, 2021, doi: 10.3390/a14040122.
- [9] M. Kaminski, "Neural Network Training Using Particle Swarm Optimization - a Case Study," 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR), 2019, pp. 115-120, doi: 10.1109/MMAR.2019.8864679.

- [10] F. Olivas, F. Valdez, O. Castillo, and P. Melin, "Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic," *Soft Computing*, vol. 20, no. 3, pp. 1057–1070, Mar. 2016, doi: 10.1007/s00500-014-1567-3.
- [11] T. Bhattacharyya, B. Chatterjee, P. Singh, J. Yoon, Z. Geem, and R. Sarkar, "Mayfly in Harmony: A new hybrid meta-heuristic feature selection algorithm," *IEEE Access*, vol. 8, pp. 195929–195945, 2020, doi: 10.1109/ACCESS.2020.3031718.
- [12] F. Olivas, F. Valdez, O. Castillo, I. Gonzalez, G. Martinez, and P. Melin, "Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems," *Applied Soft Computing Journal*, vol. 53, pp. 74–87, Apr. 2017, doi: 10.1016/j.asoc.2016.12.015.
- [13] J. Zhou, Z. Duan, Y. Li, J. Deng, and D. Yu, "PSO-based neural network optimization and its utilization in a boring machine," *Journal of Materials Processing Technology*, vol. 178, no. 1–3, pp. 19–23, Sep. 2006, doi: 10.1016/j.jmatprotec.2005.07.002.
- [14] F. Valdez, P. Melin, and O. Castillo, "A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation," *Expert Systems with Applications*, vol. 41, no. 14, Elsevier Ltd, pp. 6459–6466, Oct. 15, 2014. doi: 10.1016/j.eswa.2014.04.015.
- [15] C. Maa, M. Shanblatt, and S. Member, "A Two-Phase Optimization Neural Network," VOL. 3, NO. 6. *IEEE Transactions on Neural Networks*, NOV. 12, 1992. DOI: 10.1109/72.165602
- [16] F. Valdez, P. Melin, and O. Castillo, "An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms," in *Applied Soft Computing Journal*, Mar. 2011, vol. 11, no. 2, pp. 2625–2632. doi: 10.1016/j.asoc.2010.10.010.
- [17] K. Rajeswari and P. Lakshmi, "PSO optimized fuzzy logic controller for active suspension system," in *Proceedings - 2nd International Conference on Advances in Recent Technologies in Communication and Computing, ARTCom 2010, 2010*, pp. 278–283. doi: 10.1109/ARTCom.2010.22.
- [18] R. Martinez, O. Castillo, L. Aguilar, P. Melin (2010) Fuzzy Logic Controllers Optimization Using Genetic Algorithms and Particle Swarm Optimization. In: Sidorov G., Hernández Aguirre A., Reyes García C.A. (eds) *Advances in Soft Computing. MICAI 2010*.

Lecture Notes in Computer Science, vol 6438. Springer, Berlin, Heidelberg.
https://doi.org/10.1007/978-3-642-16773-7_41

[19] D. Cook, C. Ragsdale, and R. Major, “Combining a neural network with a genetic algorithm for process parameter optimization.”, *Engineering Applications of Artificial Intelligence*, Volume 13, Issue 4, Aug. 2000, pp. 391-396 doi:10.1016/s0952-1976(00)00021-x

[20] Z. Konstantinos, T. Stelios, “A mayfly optimization algorithm”, *Computers & Industrial Engineering*, Volume 145, 2020, 106559, ISSN 0360 8352, <https://doi.org/10.1016/j.cie.2020.106559>.

[21] J. Pacheco, “Análisis de algoritmos evolutivos para redes neuronales.” [Online]. Available: <https://www.researchgate.net/publication/26428307>.

[22] A. Uriarte, P. Melin, and F. Valdez, “Optimization of modular neural network architectures with an improved particle swarm optimization algorithm,” in *Studies in Fuzziness and Soft Computing*, vol. 361, Springer Verlag, 2018, pp. 165–174.

[23] S. Mirjalili, “Genetic algorithm,” in *Studies in Computational Intelligence*, vol. 780, Springer Verlag, 2019, pp. 43–55.

[24] J. Velásquez, C. Franco and A. Camacho, “Nonlinear time series forecasting using MARS Predicción de series de tiempo no lineales usando MARS,” *DYNA*, vol. 81, no. 184, pp. 11–19, 2014, [Online]. Available: <http://dyna.medellin.unal.edu.co/>.

[25] L. Glass and M. Mackey, “Mackey-Glass equation,” *Scholarpedia*, vol. 5, no. 3, p. 6908, 2010, doi: 10.4249/scholarpedia.6908.

[26] R. Mahajan and G. Kaur, “Neural Networks using Genetic Algorithms,” 2013.

[27] J. Soto and P. Melin, “Optimization of the Fuzzy Integrators in Ensembles of ANFIS Model for Time Series Prediction: The case of Mackey-Glass,” in *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*, Jun. 2015, vol. 89, doi: 10.2991/ifsa-eusflat-15.2015.140.

[28] Y. Shang and B. Wah, “Global Omtimization.”

[29] F. Valdez, O. Castillo, and C. Peraza, “Fuzzy Logic in Dynamic Parameter Adaptation of Harmony Search Optimization for Benchmark Functions and Fuzzy Controllers,” *Int. J. Fuzzy Syst.*, vol. 22, no. 4, pp. 1198–1211, Jun. 2020, doi: 10.1007/s40815-020-00860-7.

- [30] W. San-Um, S. Maneechot, and M. Jirayutcharonsuk, "THE PREDICTION OF HIGH-DIMENSIONAL MACKAY-GLASS CHAOTIC TIME SERIES USING NONLINEAR AUTOREGRESSIVE NEURAL NETWORK," 2014. [Online]. Available: <https://www.researchgate.net/publication/305108695>.
- [31] S. Katoch, S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.
- [32] A. Nassef, "Maximizing SOFC performance through optimal parameters identification by modern optimization algorithms," *Renew. Energy*, vol. 138, pp. 458–464, Aug. 2019, doi: 10.1016/j.renene.2019.01.072.
- [33] P. Melin, J. Soto, O. Castillo, and J. Soria, "A new approach for time series prediction using ensembles of ANFIS models," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3494–3506, Feb. 2012, doi: 10.1016/j.eswa.2011.09.040.
- [34] M. Bahiraei, S. Nazari, H. Moayedi, and H. Safarzadeh, "Using neural network optimized by imperialist competition method and genetic algorithm to predict water productivity of a nanofluid-based solar still equipped with thermoelectric modules," *Powder Technol.*, vol. 366, pp. 571–586, Apr. 2020, doi: 10.1016/j.powtec.2020.02.055.
- [35] J. Soto, P. Melin, and O. Castillo, "A New Approach for Time Series Prediction Using Ensembles of IT2FNN Models with Optimization of Fuzzy Integrators," *Int. J. Fuzzy Syst.*, vol. 20, no. 3, pp. 701–728, Mar. 2018, doi: 10.1007/s40815-017-0443-6.
- [36] F. Gaxiola, P. Melin, F. Valdez, J. Castro, and A. Manzo-Martínez, "PSO with dynamic adaptation of parameters for optimization in neural networks with interval type-2 fuzzy numbers weights," *Axioms*, vol. 8, no. 1, Jan. 2019, doi: 10.3390/axioms8010014.
- [37] P. Melin, O. Castillo, and J. Kacprzyk, "Studies in Computational Intelligence 915 Recent Advances of Hybrid Intelligent Systems Based on Soft Computing." [Online]. Available: <http://www.springer.com/series/7092>.
- [38] V. Gromov and A. Shulga, "Chaotic time series prediction with employment of ant colony optimization," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8474–8478, 2012, doi: 10.1016/j.eswa.2012.01.171.
- [39] S. Kumayl, "A Novel Artificial Neural Network (ANN) Using The Mayfly Algorithm for Classification," pp. 8–13, 2021.

[40] G. Ahmadi and M. Dehghandar, "Mackey-Glass Time Series Prediction Using Rough-Neural Networks," pp. 2–5.

Anexo 1.- Pseudocódigo de Mayfly

Función objetivo $f(x)$, $x = (x_1, \dots, x_d)^T$

Inicializar la población de Mayfly machos y velocidades x_i ($i = 1, 2, \dots, N$) v_{mi}

Inicializar la población de Mayfly hembras y velocidades y_i ($i = 1, 2, \dots, M$) v_{fi}

Evaluar soluciones

Encontrar el mejor global

Do While Criterio de parada

 Actualizar velocidades y soluciones de machos y hembras

 Evaluar soluciones

 Clasificación de Mayflys

 Pareja de Mayfly

 Evaluar descendencia

 Separar descendencia de machos y hembras aleatoriamente

 Remplaza las peores soluciones con las mejores

 Actualización y obtención del mejor local y el mejor global

end while

Anexo 2.- Pseudocódigo de Mayfly con adaptador difuso de parámetros.

Cargar el adaptador difuso

Función objetivo $f(x)$, $x = (x_1, \dots, x_d) T$

Inicializar la población de Mayfly machos y velocidades x_i ($i = 1, 2, \dots, N$) v_{mi}

Inicializar la población de Mayfly hembras y velocidades y_i ($i = 1, 2, \dots, M$) v_{fi}

Evaluar soluciones

Encontrar el mejor global

Do While Criterio de parada

 Actualizar velocidades y soluciones de machos y hembras

 Evaluar soluciones

 Clasificación de Mayflys

 Pareja de Mayfly

 Evaluar descendencia

 Calcular la diversidad

 Separar descendencia de machos y hembras aleatoriamente

 Remplaza las peores soluciones con las mejores

 Actualización y obtención del mejor local y el mejor global

 Adaptar los parámetros a_1 y a_2

end while

Anexo 3.- Pseudocódigo del cálculo de la diversidad.

Inicializar la posición i y j

Inicializar el número de dimensiones

Inicializar el número de individuos

Do While número de individuos

 Do While número de dimensiones

 Restamos la posición actual del mejor global

 Elevamos el resultado al cuadrado y lo acumulamos

 end while

end while

Sacamos la raíz cuadrada de la diversidad

Validamos diversidad ≥ 0 y ≤ 1

Anexo 4.- Generación de la red neuronal para Mackey Glass.

Tomamos los valores de entrenamiento

Inicializamos datos de entrada;

Inicializamos el número de neurona

Seleccionamos el número de épocas, en este caso 100

Seleccionamos el factor de aprendizaje, por defecto en 0.1

Elegimos el algoritmo de entrenamiento Levenberg-Marquardt

Seleccionamos el porcentaje de entrenamiento

Seleccionamos el porcentaje de validación

Seleccionamos el porcentaje de testeo.

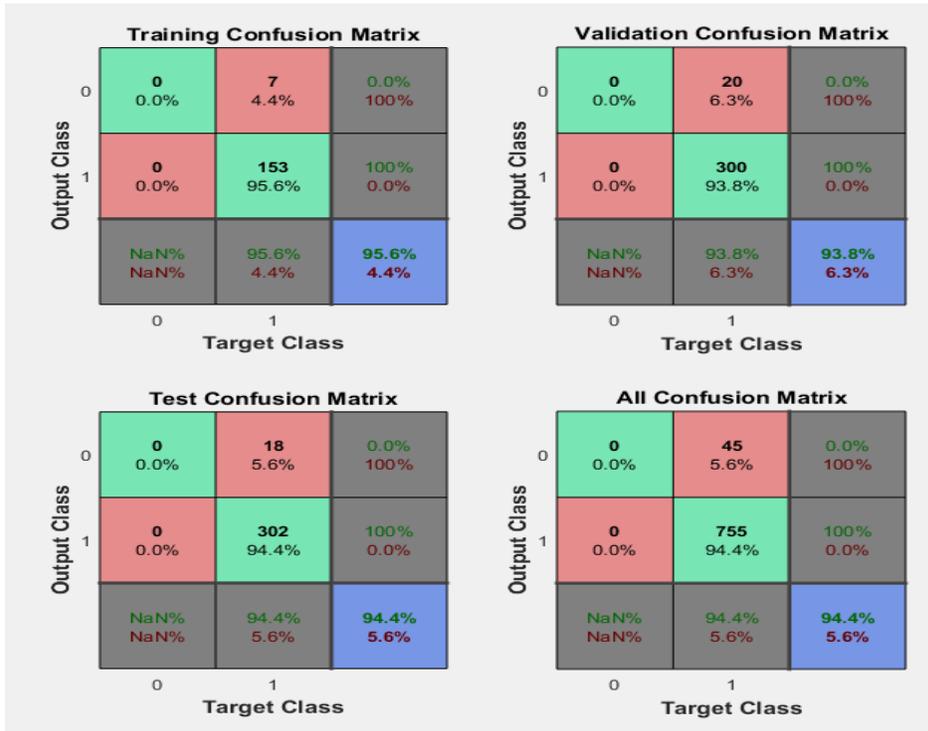
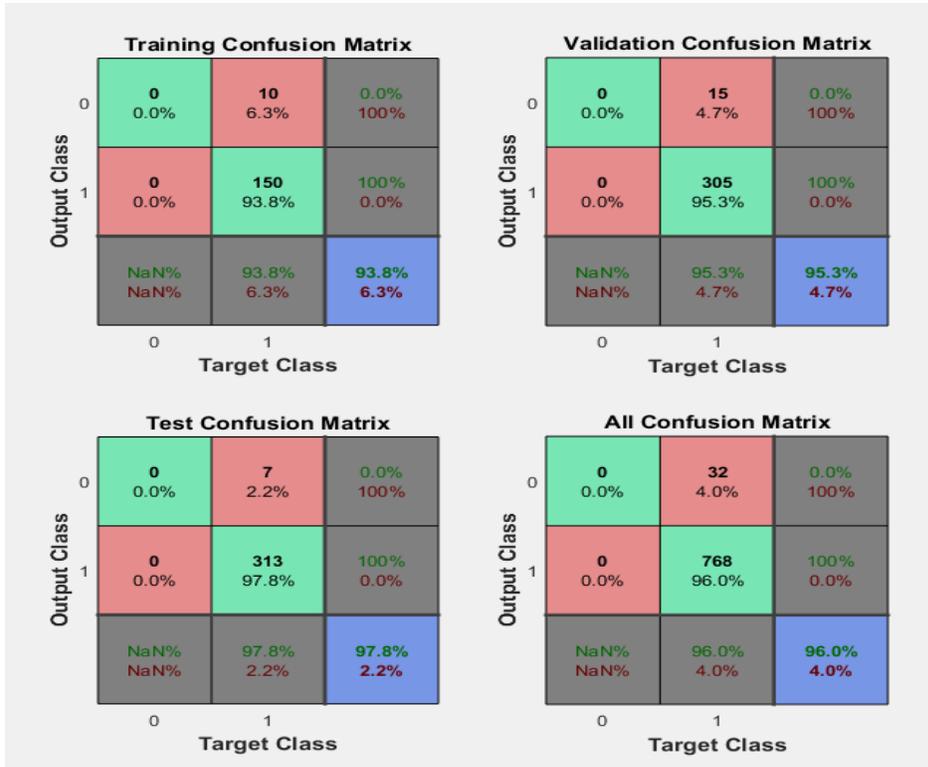
Do While error mínimo o épocas

Actualización de pesos 'w' en la red

End

Calculamos el error cuadrático medio





Anexo 5 Datos utilizados para el entrenamiento de la red neuronal

x	t	x	t	x	t
1	1.2	268	0.73067874	535	0.92803078
2	1.085805	269	0.68205143	536	0.97707299
3	0.98247708	270	0.63591552	537	1.02037963
4	0.88898211	271	0.59241041	538	1.05803884
5	0.80438435	272	0.55187641	539	1.09045138
6	0.72783712	273	0.51500975	540	1.11804949
7	0.65857432	274	0.48293372	541	1.14120505
8	0.59590274	275	0.45724287	542	1.16026321
9	0.53919515	276	0.44014243	543	1.17565406
10	0.48788399	277	0.43462063	544	1.18805232
11	0.44145573	278	0.4443293	545	1.1985572
12	0.3994457	279	0.47264395	546	1.20881167
13	0.36143345	280	0.52063485	547	1.22083403
14	0.32703854	281	0.58502358	548	1.23622105
15	0.29591673	282	0.65840842	549	1.25466073
16	0.26775656	283	0.73219347	550	1.27255718
17	0.24227617	284	0.79961371	551	1.28346665
18	0.21922057	285	0.85680533	552	1.28117114
19	0.26140582	286	0.9023133	553	1.26296725
20	0.33826746	287	0.93618269	554	1.23012007
21	0.43540483	288	0.95924662	555	1.18607257
22	0.53147163	289	0.97272364	556	1.13465652
23	0.61387325	290	0.97804756	557	1.07919292
24	0.67890403	291	0.97682509	558	1.0222533
25	0.72707625	292	0.97085797	559	0.96570519
26	0.76029545	293	0.96221596	560	0.91082606
27	0.78072946	294	0.95334837	561	0.85839836
28	0.79042974	295	0.94716697	562	0.80876009
29	0.79122711	296	0.94693813	563	0.76182458
30	0.78471917	297	0.95577017	564	0.7171387
31	0.77228607	298	0.97564087	565	0.67409557
32	0.75511346	299	1.00621853	566	0.63233446
33	0.73421564	300	1.04390934	567	0.59211519
34	0.71045703	301	1.08206259	568	0.55437302
35	0.6845713	302	1.11350328	569	0.52049869
36	0.66917778	303	1.13402334	570	0.49219046
37	0.66987664	304	1.14355103	571	0.4715591

38	0.68897764	305	1.14473322	572	0.46136277
39	0.72438359	306	1.1410832	573	0.46505586
40	0.77140367	307	1.13585586	574	0.48615394
41	0.8245744	308	1.13157058	575	0.52650815
42	0.879	309	1.12984015	576	0.58417731
43	0.93128254	310	1.13130791	577	0.65303439
44	0.97971939	311	1.13564124	578	0.72511012
45	1.02384867	312	1.14158466	579	0.79357523
46	1.06379604	313	1.14703613	580	0.85409301
47	1.09976614	314	1.14906365	581	0.90456479
48	1.13178027	315	1.14395069	582	0.94434302
49	1.15962438	316	1.12787667	583	0.97360846
50	1.18292676	317	1.09886391	584	0.99305793
51	1.20128468	318	1.05864033	585	1.00377861
52	1.21437845	319	1.01181866	586	1.00717607
53	1.22392373	320	0.96330977	587	1.00492874
54	1.23266807	321	0.91675311	588	0.99899792
55	1.24340787	322	0.87430475	589	0.9917013
56	1.25767555	323	0.83689649	590	0.98579842
57	1.27461546	324	0.8045119	591	0.98444644
58	1.29034564	325	0.7764361	592	0.99080795
59	1.29872917	326	0.75153442	593	1.00719148
60	1.2940404	327	0.72857609	594	1.03390473
61	1.2736468	328	0.70656041	595	1.0681911
62	1.2384465	329	0.68497661	596	1.10396384
63	1.19148403	330	0.66396247	597	1.1336338
64	1.13643259	331	0.6444061	598	1.15171629
65	1.07670979	332	0.62808799	599	1.15703253
66	1.0151402	333	0.61787826	600	1.15189434
67	0.95390797	334	0.6177104	601	1.14006937
68	0.89463124	335	0.63171839	602	1.12529852
69	0.8384775	336	0.66223961	603	1.11063246
70	0.78595307	337	0.70782687	604	1.09822298
71	0.7369421	338	0.76340294	605	1.08925703
72	0.69087854	339	0.82271676	606	1.08390331
73	0.64709076	340	0.88073316	607	1.08126507
74	0.6052051	341	0.93442098	608	1.07937929
75	0.56541397	342	0.98237538	609	1.07530106
76	0.52847823	343	1.02415154	610	1.06538698
77	0.49557188	344	1.05974918	611	1.04613861
78	0.46822807	345	1.08933231	612	1.01587422

79	0.44851467	346	1.11313845	613	0.97616147
80	0.43934839	347	1.13150839	614	0.93115728
81	0.44463219	348	1.14498185	615	0.88547312
82	0.46855182	349	1.15443202	616	0.84275975
83	0.513322	350	1.161225	617	0.805441
84	0.57627828	351	1.1673424	618	0.77486706
85	0.64961274	352	1.17528228	619	0.75145186
86	0.72398539	353	1.18746802	620	0.73476568
87	0.79233167	354	1.20505992	621	0.72368667
88	0.85086266	355	1.22652575	622	0.71667969
89	0.89820493	356	1.2469115	623	0.71218435
90	0.93432189	357	1.25912488	624	0.70903715
91	0.9598408	358	1.25727693	625	0.70685171
92	0.9757542	359	1.23940645	626	0.70631867
93	0.98331098	360	1.20734787	627	0.70941288
94	0.98397217	361	1.1648342	628	0.71937746
95	0.97940532	362	1.11582081	629	0.73999575
96	0.9715381	363	1.06367183	630	0.77362221
97	0.96268069	364	1.01095171	631	0.81894057
98	0.95566866	365	0.95947167	632	0.87101331
99	0.95385651	366	0.91038514	633	0.92397182
100	0.9606592	367	0.86424169	634	0.97354093
101	0.97847937	368	0.82098479	635	1.01769802
102	1.00736961	369	0.7799637	636	1.05610601
103	1.0440542	370	0.74011548	637	1.08931582
104	1.08210122	371	0.70043757	638	1.11816828
105	1.11420201	372	0.66059911	639	1.14345486
106	1.13557721	373	0.62126514	640	1.16578508
107	1.14551596	374	0.58389836	641	1.18559473
108	1.14630823	375	0.5503474	642	1.20323965
109	1.14141647	376	0.52266677	643	1.21913664
110	1.13422643	377	0.50323764	644	1.23391689
111	1.12747624	378	0.49494579	645	1.24850982
112	1.1230482	379	0.50104562	646	1.26392908
113	1.12188609	380	0.52430158	647	1.2803902
114	1.12395015	381	0.5653782	648	1.29578841
115	1.12820869	382	0.62150397	649	1.30495725
116	1.13266795	383	0.68691844	650	1.30176062
117	1.13440251	384	0.75501152	651	1.28287403
118	1.12965352	385	0.82033496	652	1.24916591
119	1.11449881	386	0.87929927	653	1.20402412

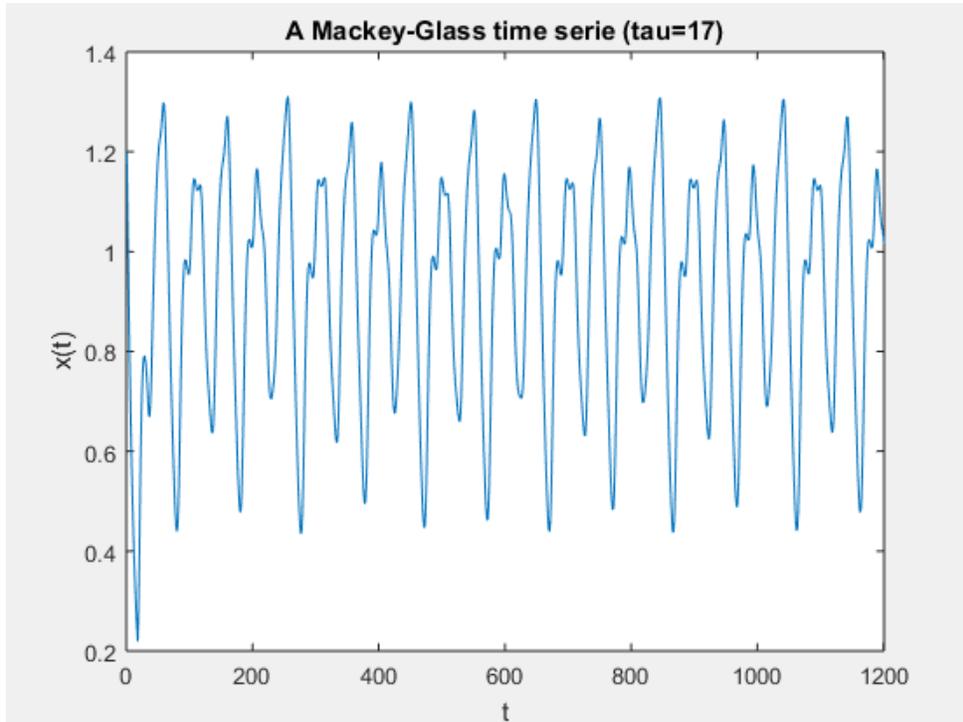
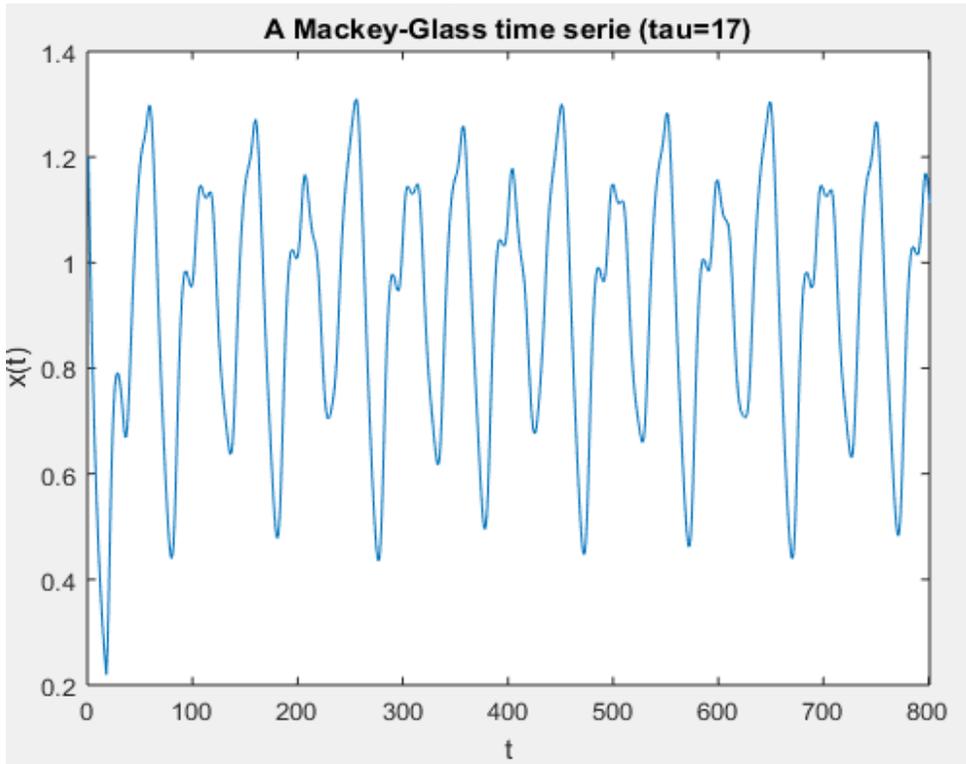
120	1.08670416	387	0.92986504	654	1.15128695
121	1.04762424	388	0.97100297	655	1.09420431
122	1.00163276	389	1.00236961	656	1.03521769
123	0.9536505	390	1.02421965	657	0.97606869
124	0.90747862	391	1.03737323	658	0.91796638
125	0.86548654	392	1.04313275	659	0.86172145
126	0.82882535	393	1.04319107	660	0.80783242
127	0.79767252	394	1.03959744	661	0.75653378
128	0.77144254	395	1.03478427	662	0.7078301
129	0.74902578	396	1.03157812	663	0.66156415
130	0.72909216	397	1.03304188	664	0.61758415
131	0.7104352	398	1.04198272	665	0.5760046
132	0.69229156	399	1.06010432	666	0.53740175
133	0.67457841	400	1.08697168	667	0.50280744
134	0.65805383	401	1.11907199	668	0.47363493
135	0.64447211	402	1.14977409	669	0.45175465
136	0.63675313	403	1.17148288	670	0.43972097
137	0.63889478	404	1.17931594	671	0.44091592
138	0.65500854	405	1.17299025	672	0.45914156
139	0.68721666	406	1.1556248	673	0.4970819
140	0.73371039	407	1.13151284	674	0.55385827
141	0.78915767	408	1.10464389	675	0.62376681
142	0.84734917	409	1.07813686	676	0.69820672
143	0.90352955	410	1.05413016	677	0.76924208
144	0.95502662	411	1.03377585	678	0.8316949
145	1.00076946	412	1.01719858	679	0.88314348
146	1.04058506	413	1.00342868	680	0.92302346
147	1.07467484	414	0.99040732	681	0.95178679
148	1.10333154	415	0.97520848	682	0.97040148
149	1.12684654	416	0.95465457	683	0.98013156
150	1.14554483	417	0.9264518	684	0.98245788
151	1.15989753	418	0.89048844	685	0.97905812
152	1.1706803	419	0.84918667	686	0.97183828
153	1.17915193	420	0.80637163	687	0.963023
154	1.18717607	421	0.76581658	688	0.95527335
155	1.19707967	422	0.73058001	689	0.9517169
156	1.21095487	423	0.70296756	690	0.95566919
157	1.22932044	424	0.68457168	691	0.96985284
158	1.24966673	425	0.67616612	692	0.99522837
159	1.26617445	426	0.67754706	693	1.02984553
160	1.27188549	427	0.68753878	694	1.06834573

161	1.2622814	428	0.70429679	695	1.1033316
162	1.23702902	429	0.72583998	696	1.12877417
163	1.19889312	430	0.75062197	697	1.14260588
164	1.15175942	431	0.77797939	698	1.14638088
165	1.09932091	432	0.80833608	699	1.14340365
166	1.04457106	433	0.84296897	700	1.13719409
167	0.98973913	434	0.8830414	701	1.1307206
168	0.9363771	435	0.92803675	702	1.12611072
169	0.88545513	436	0.97497068	703	1.12454989
170	0.83741149	437	1.01968089	704	1.12624195
171	0.79215967	438	1.05894295	705	1.13041501
172	0.74912621	439	1.09144562	706	1.13537317
173	0.70745855	440	1.11754348	707	1.13854448
174	0.66647633	441	1.13860632	708	1.13650805
175	0.62617029	442	1.15642448	709	1.12532807
176	0.5873731	443	1.17275368	710	1.10193397
177	0.55151877	444	1.18898992	711	1.06630387
178	0.52035235	445	1.20597992	712	1.02199751
179	0.49590563	446	1.22397287	713	0.97403901
180	0.48069493	447	1.24268251	714	0.92672021
181	0.47786657	448	1.26137147	715	0.88287746
182	0.49087159	449	1.27880942	716	0.84401523
183	0.52222683	450	1.29294308	717	0.81057501
184	0.57156928	451	1.30036042	718	0.78216849
185	0.63456272	452	1.29645554	719	0.75781735
186	0.70418471	453	1.27755864	720	0.73624765
187	0.77344351	454	1.24359221	721	0.71622757
188	0.83723916	455	1.19792766	722	0.69688816
189	0.89262529	456	1.14504564	723	0.67796411
190	0.93822794	457	1.08874879	724	0.6599465
191	0.97363477	458	1.03162258	725	0.6442163
192	0.99906288	459	0.97515683	726	0.63322537
193	1.01523755	460	0.92005448	727	0.63058762
194	1.02332022	461	0.8665468	728	0.64055417
195	1.02483064	462	0.81465804	729	0.66631385
196	1.02160362	463	0.76439176	730	0.70771875
197	1.0158133	464	0.71582835	731	0.76058595
198	1.01003948	465	0.66915085	732	0.81869367
199	1.00727091	466	0.62464249	733	0.87652182
200	1.01065959	467	0.58270762	734	0.93055355
201	1.02286331	468	0.54395087	735	0.97911461

202	1.04503308	469	0.50930512	736	1.02169131
203	1.07570769	470	0.48016268	737	1.05832631
204	1.11007134	471	0.45850643	738	1.0892531
205	1.14071754	472	0.44705096	739	1.1147432
206	1.16075685	473	0.44921581	740	1.13510109
207	1.16711853	474	0.46850196	741	1.15074841
208	1.16101107	475	0.50693032	742	1.1623596
209	1.14604536	476	0.56309938	743	1.17102878
210	1.12630607	477	0.631571	744	1.17842593
211	1.10533069	478	0.70460916	745	1.18679596
212	1.08576949	479	0.77492072	746	1.19852319
213	1.06931316	480	0.83738423	747	1.21504921
214	1.05665415	481	0.88928814	748	1.23537286
215	1.04742136	482	0.92976568	749	1.25506303
216	1.04012992	483	0.95910648	750	1.26722526
217	1.03223132	484	0.97824103	751	1.26574726
218	1.02038438	485	0.98844238	752	1.24826726
219	1.00117473	486	0.99119953	753	1.21631344
220	0.97248842	487	0.98820197	754	1.17347323
221	0.93494549	488	0.98139162	755	1.12366479
222	0.89186773	489	0.97305779	756	1.07026859
223	0.84760359	490	0.96592758	757	1.01589007
224	0.80598422	491	0.96312973	758	0.96239717
225	0.76982932	492	0.96784941	759	0.91102139
226	0.74100722	493	0.98257592	760	0.86242959
227	0.72051522	494	1.00807365	761	0.81674216
228	0.70846718	495	1.04232094	762	0.7735398
229	0.70410678	496	1.07993537	763	0.73198379
230	0.70598455	497	1.11343879	764	0.69118424
231	0.71232932	498	1.13674842	765	0.65074847
232	0.72153147	499	1.14791568	766	0.61115133
233	0.73262505	500	1.14873031	767	0.57364345
234	0.74568509	501	1.14270475	768	0.539902
235	0.76206194	502	1.13348597	769	0.51184985
236	0.78425723	503	1.12411182	770	0.49176692
237	0.81499532	504	1.11675216	771	0.48249639
238	0.85525488	505	1.11261642	772	0.48739153
239	0.90261499	506	1.11190282	773	0.50954727
240	0.95212445	507	1.11378649	774	0.55012741
241	0.99901135	508	1.11646583	775	0.60669949
242	1.04047927	509	1.11723967	776	0.67336466

243	1.07579201	510	1.11262158	777	0.74299153
244	1.10556596	511	1.09881761	778	0.80959662
245	1.13103368	512	1.07317355	779	0.86928897
246	1.15351405	513	1.03614031	780	0.9199883
247	1.17409981	514	0.99147996	781	0.96079543
248	1.1935318	515	0.94413232	782	0.99155339
249	1.2122195	516	0.89822048	783	1.01267263
250	1.23035195	517	0.85646026	784	1.02506321
251	1.2480278	518	0.82030342	785	1.03005668
252	1.2653149	519	0.79017343	786	1.02933164
253	1.2820927	520	0.76565548	787	1.02489605
254	1.29744137	521	0.74570605	788	1.01913275
255	1.30856741	522	0.72894173	789	1.01484221
256	1.31045545	523	0.71399695	790	1.01513258
257	1.29814604	524	0.69988392	791	1.02296788
258	1.27014981	525	0.68628185	792	1.04031481
259	1.22890236	526	0.67373558	793	1.06706405
260	1.17851493	527	0.66381853	794	1.1000333
261	1.12280657	528	0.65929419	795	1.13280923
262	1.0645965	529	0.66404677	796	1.15774845
263	1.00574423	530	0.68213461	797	1.16964959
264	0.94741592	531	0.71560685	798	1.16781813
265	0.8903422	532	0.76249113	799	1.15504392
266	0.83499833	533	0.8173326	800	1.13543568
267	0.78170562	534	0.87400652		

Anexo 6.- Graficas de Mackey Glass



Anexo 7.- Adaptadores difusos

FIS Name: difusoa1 FIS Type: mamdani

And method: Current Variable Name:

Or method: Current Variable Type:

Implication: Current Variable Range:

Aggregation:

Defuzzification:

Help Close

Updating Membership Function Editor

FIS Name: adaptbest_3trap FIS Type: mamdani

And method: Current Variable Name:

Or method: Current Variable Type:

Implication: Current Variable Range:

Aggregation:

Defuzzification:

Help Close

Opening Membership Function Editor

