



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Ciencias en Inteligencia Artificial

**Optimización de modelo de partículas contaminantes utilizando el algoritmo de optimización de forrajeo bacteriano BFOA**

**Tesis**

Que como parte de los requisitos para obtener el Grado de  
Maestría en Ciencias en Inteligencia Artificial

**Presenta:**

María del Carmen Cabrera Hernández

**Dirigido por:**

Dr. Marco Antonio Aceves Fernández

Dr. Marco Antonio Aceves Fernández

Presidente

Dr. Juvenal Rodríguez Reséndiz

Secretario

Dr. Saúl Tovar Arriaga

Vocal

Dr. Jesús Carlos Pedraza Ortega

Suplente

M. en I. Artemio Sotomayor Olmedo

Suplente

Centro Universitario, Querétaro, Qro.

Agosto 2019

México

## AGRADECIMIENTOS

A la Universidad Autónoma de Querétaro y la Facultad de Ingeniería, por ser un segundo hogar para mí, por todo lo que me ha dado y enseñado.

Gracias al Dr. Marco Antonio Aceves Fernández por permitirme trabajar con él, por motivarme a terminar el proyecto, por la confianza, su paciencia, la dedicación que tiene para con sus alumnos, por los conocimientos que compartió conmigo y por inspirarme con la pasión que tiene por su trabajo.

Agradezco a mis papás, Joel Cabrera Méndez y María del Carmen Hernández Medina por su apoyo y amor incondicional, por ser mis guías, enseñarme a ser mejor cada día, y también por darme una familia maravillosa, los amo.

Gracias a José María, Joel, Jorge y Ana, mis hermanos mayores, mis mejores amigos, que han sido un ejemplo y un apoyo para mí durante toda mi vida, un agradecimiento especial a Rosita, mi hermana pequeña, cómplice, mejor amiga y mi apoyo principal durante la realización de este trabajo, y a mi sobrino Edmundo, los amo a todos.

A mis amigos Música, Abif, Manuel, Martín, Iván, Denisse, Karla, María José, Miguel, Francisco, Emiliano, David, Pedro, Guillermo, José Manuel, Aranza y Andrés, gracias por acompañarme en esta aventura y las que vienen.

A mis profesores, Dr. Saúl Tovar, Dr. Carlos Pedraza, Dr. Emilio Vargas, Dr. Efrén Gorrostieta, Dr. Eric Huerta, gracias por contribuir con sus enseñanzas para llevar a cabo este proyecto.

## TABLA DE CONTENIDOS

<b>1. INTRODUCCIÓN</b>	<b>8</b>
<b>2. MARCO TEÓRICO</b>	<b>11</b>
<b>3. HIPÓTESIS Y OBJETIVOS</b>	<b>17</b>
3.1 HIPÓTESIS	17
3.2 OBJETIVO GENERAL:	17
3.3 OBJETIVO PARTICULAR:	17
<b>4. MATERIALES Y MÉTODOS</b>	<b>18</b>
4.1 DATOS	18
4.2 CONSTRUCCIÓN DEL MODELO	20
4.2.1 <i>Sistema adaptativo de inferencia neurodifusa (ANFIS)</i>	20
4.3 OPTIMIZACIÓN DEL MODELO	23
<b>4.4 METODOLOGÍA PROPUESTA</b>	<b>31</b>
<b>5. RESULTADOS Y DISCUSIÓN</b>	<b>39</b>
5.1 RESULTADOS	39
5.1.1 <i>Variación del número de bacterias</i>	43
5.1.2 <i>Variación de los pasos de quimiotaxis</i>	46
5.1.3 <i>Variación de los pasos de reproducción</i>	48
5.1.4 <i>Variación de los eventos de eliminación/dispersión</i>	52
5.1.5 <i>Variación del tamaño del paso tomado durante el giro</i>	54
5.1.6 <i>Configuración en conjunto de los parámetros</i>	55
5.2 DISCUSIÓN DE RESULTADOS	60
<b>6. CONCLUSIONES</b>	<b>63</b>
<b>REFERENCIAS</b>	<b>66</b>
<b>APÉNDICE</b>	<b>74</b>

## ÍNDICE DE FIGURAS

<b>Figura</b>		<b>Página</b>
1.1	Comparativa del tamaño de partículas atmosféricas y otros materiales	9
4.1	Mapa de las estaciones de RAMA	19
4.2	Estructura de ANFIS	21
4.3	Quimiotaxis de una bacteria	24
4.4	Diagrama de flujo BFOA	28
4.5	Muestra de los datos	31
4.6	Herramienta Neuro-Fuzzy designer	32
4.7	Reglas del modelo ANFIS	33
4.8	ANFIS de 7 estaciones	35
4.9	Metodología propuesta	38
5.1	Concentración de PM10 enero 2016 estación FAC	39
5.2	Comparación datos reales vs ANFIS	42
5.3	Detalle del error del modelo generado con ANFIS	42
5.4	Error del modelo generado con ANFIS	43
5.5	Error RMSE de acuerdo a la población S	45
5.6	Tiempo de ejecución y el número de bacterias	46
5.7	Diferentes valores para Nc (pasos de quimiotaxis)	47
5.8	Tiempo de ejecución para diferentes valores para Nc	47
5.9	Variación de Nre	49
5.10	Variación de Nre y Nc	50
5.11	Costo de tiempo de variación de Nre y Nc	51
5.12	Costo/beneficio de Nre	51
5.13	Variación de los eventos de eliminación/dispersión	53
5.14	Complejidad computacional (tiempo) de Ned	53
5.15	Variación del tamaño de paso tomado durante el giro	55
5.16	Errores de ANFIS y Optimizado con BFOA con población diferente	57
5.17	Comparación de errores del Modelo ANFIS y Modelo optimizado BFOA/ANFIS	57
5.18	Comparación de errores de Modelo optimizado BFOA/ANFIS con variación en la población S	58
5.19	Comparación Modelo optimizado BFOA/ANFIS vs. Datos reales vs. ANFIS	59
5.20	Complejidad computacional del Modelo optimizado BFOA/ANFIS	59
5.21	Variabilidad de los modelos ANFIS para diferentes estaciones	60

## ÍNDICE DE TABLAS

<b>Tabla</b>		<b>Página</b>
2.1	Estado del arte de Inteligencia Artificial en aplicaciones ambientales	13
4.1	Datos para generar el modelo	18
4.2	Estaciones de Monitoreo	20
5.1	Variabilidad de los datos de entrenamiento	39
5.2	Variabilidad de los datos y sus factores ambientales	40
5.3	Variabilidad entre estaciones mismo día y hora	40
5.4	Relación de cantidad de bacterias, RMSE y tiempo de ejecución	45
5.5	Valores de $N_c$ y el error RMSE	48
5.6	Valores de $N_{re}$ y RMSE	49
5.7	Valores de $N_{ed}$ y RMSE	53
5.8	Diferentes valores de $C(i)$	55
5.9	Configuración final de parámetros	56
5.10	RMSE para ANFIS y BFOA	56
5.11	Datos de la complejidad computacional en minutos del Modelo BFOA/ANFIS	60
5.12	Relación de $N_{re}$ y $N_c$	62

## RESUMEN

El monitoreo de la calidad del aire es un problema que se ha vuelto elemental para la humanidad, ya que cuando la calidad del aire es deficiente, constituye una amenaza seria para la salud, concretamente, el material particulado es uno de los contaminantes con mayor impacto en la calidad del aire. El material particulado se encuentra compuesto de partículas de materia sólida y líquida que es causante de enfermedades del sistema respiratorio y cardiovascular, esto debido a su tamaño, en el orden de los micrómetros ( $\mu\text{m}$ ), y que se clasifica como PM10 ( $10 \mu\text{m}$ ) y PM2.5 ( $2.5 \mu\text{m}$ ), además cabe mencionar que su comportamiento ha demostrado ser no lineal.

Dada la complejidad del problema, la Inteligencia artificial es una herramienta ideal para encontrar una solución, en el presente trabajo, el objetivo principal es la optimización de un modelo base, en este caso generado por el método ANFIS, para predecir la concentración de partículas contaminantes, en específico PM10, utilizando algoritmos de inteligencia de enjambre. Específicamente, se utiliza BFOA para la optimización, demostrando que es posible disminuir el error del modelo ANFIS. Y es por esto que, para comprobar la efectividad del modelo propuesto, se llevaron a cabo varios experimentos con los datos y los parámetros propios del algoritmo BFOA, esto, con el fin de en un futuro poder utilizar este método para generar herramientas de predicción del comportamiento del material particulado.

**(Palabras clave: Contaminación del aire, algoritmo de optimización de forrajeo por bacterias (BFOA), Inteligencia de enjambre, ANFIS)**

## SUMMARY

The monitoring of air quality is a problem that has become essential for humanity, when air quality is poor, it constitutes a serious threat to health, specifically; the particulate matter is one of the pollutants with the greatest impact in air quality. It is composed of particles of solid and liquid matter being the causative agent of diseases of the respiratory and cardiovascular system, due to its size, in the order of micrometers ( $\mu\text{m}$ ), and which is classified as PM10 (10  $\mu\text{m}$ ) and PM2.5 (2.5 $\mu\text{m}$ ). In addition, it is also worth mentioning that its behavior has been shown to be non-linear.

Given the complexity of the problem, artificial intelligence is an ideal tool to get a solution, in the present work; the main objective is the optimization of a base model, in this case, generated by the ANFIS method, to predict the concentration of pollutant particles, in specific PM10, using swarm intelligence algorithms. Specifically, BFOA is used for optimization, demonstrating that it is possible to reduce the error of the ANFIS model. For this reason, in order to verify the effectiveness of the proposed model, several experiments were carried out with the data and own parameters of the BFOA algorithm, with the purpose of using this method in the future to generate predictive tools for the behavior of particulate matter.

**(Key words: Air pollution, Bacterial Foraging Optimization Algorithm (BFOA), swarm intelligence, ANFIS)**

## 1. INTRODUCCIÓN

La calidad del aire es un tema de vital importancia para la salud, que data desde el episodio de Londres en 1952, en el cual llegaron a morir alrededor de 12,000 personas por la excesiva contaminación del aire, y aunque no se ha repetido un episodio tan agudo, sí han existido algunos otros eventos de contaminación ambiental de gravedad como el ocurrido en Augsburgo (Alemania) en 1985 que tuvo una duración de 5 días, fue que entonces el monitoreo de la contaminación se centró en registrar las variaciones diarias y su relación con la mortalidad y otros asuntos de salud pública como el ingreso a los hospitales por complicaciones en las vías respiratorias. Desde entonces los contaminantes que son monitoreados incluyen gases como: el ozono (O<sub>3</sub>), dióxido de nitrógeno (NO<sub>2</sub>), dióxido de azufre (SO<sub>2</sub>) y material particulado (PM 2.5 y PM10) (Brunekreef, 2002).

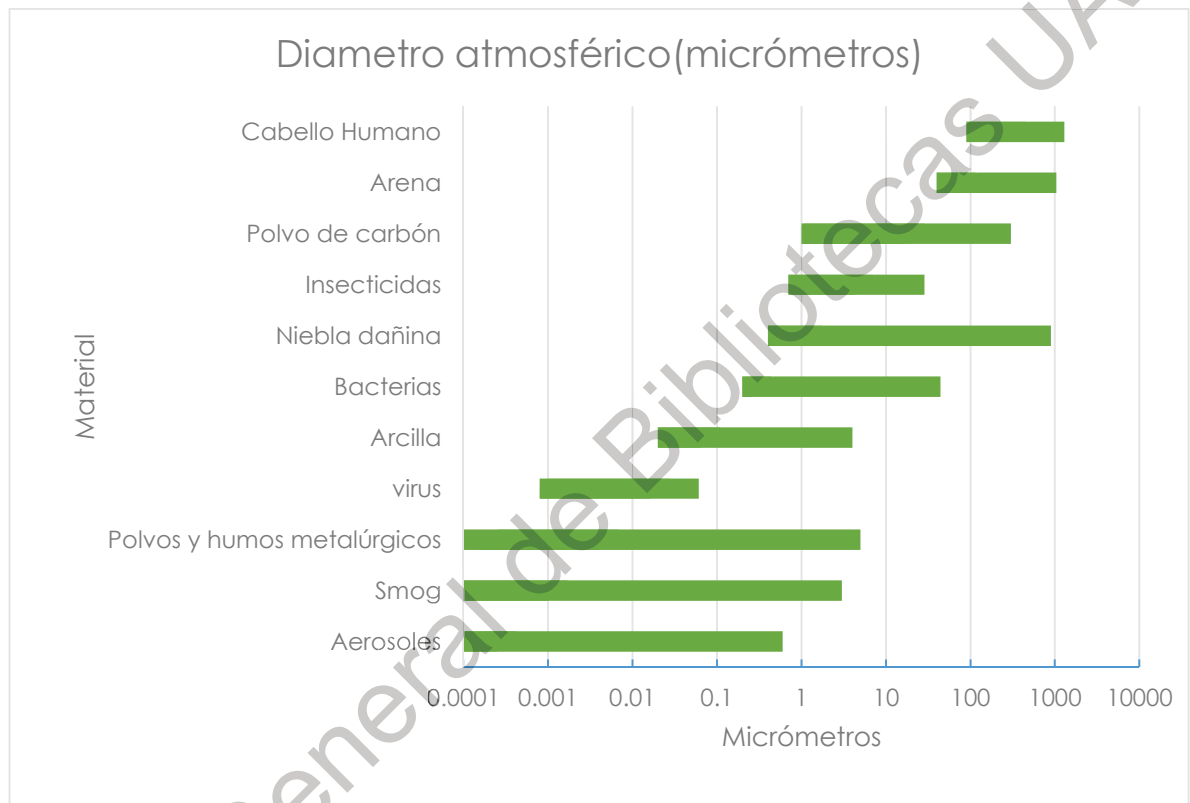
El material particulado (MP) o partículas atmosféricas, son pequeñas piezas de materia suspendidas en la atmósfera que bien pueden ser materia sólida o pequeñas gotas, cuyo origen puede ser por procesos naturales (polen, erosión terrestre, etc.) o puede ser originado a partir de las actividades humanas (construcción, actividad industrial, etc.), (Kelly, 2012). Así mismo el material particulado se desprende directamente de la fuente, o se origina por reacciones químicas ocurridas directamente en la atmósfera.

Las partículas atmosféricas se pueden clasificar según su tamaño, hay partículas con un diámetro atmosférico de 10 µm (micras de metro) llamadas PM10 y otras más finas que son de 2,5 µm (PM2.5), es decir son incluso más pequeñas que un cabello humano, figura 1.1, lo que las hace adecuadas para ser inhaladas fácilmente por los seres humanos y causar graves problemas de salud en las vías respiratorias y sistemas cardiovasculares (Brook, 2010).

Es importante destacar que, aunque se monitorean gases como el ozono (O<sub>3</sub>), dióxido de nitrógeno (NO<sub>2</sub>), dióxido de azufre (SO<sub>2</sub>) y PM10 como parte de los contaminantes atmosféricos, PM10 ha demostrado en ser uno de los contaminantes



que causan mayores problemas de salud, incluso siendo un factor de mortalidad en un sector de la población (Tellez, 2000) de ahí la importancia del desarrollo de herramientas más precisas para modelar su concentración en el medio ambiente



**Figura 1.1** Comparativa del tamaño de partículas atmosféricas y otros materiales (adaptado de Alley, 1998)

El comportamiento de las partículas atmosféricas no se puede modelar de manera sencilla ya que depende de factores atmosféricos propios de cada región, así como de la actividad humana que se desarrolla en esa zona es por eso que se ha recurrido a algoritmos y técnicas que manejen comportamientos complejos y no lineales.

El desarrollo de un modelo capaz de predecir de manera eficaz el comportamiento del material particulado es de vital importancia en un mundo donde la actividad humana aporta cada día más y más material nocivo para la salud de la población

mundial y es por eso que se desea contar con una herramienta de predicción que sea certera y que emita alertas que permitan informar cuando el material particulado suspendido en la atmósfera han alcanzado un nivel nocivo sobre todo para la población vulnerable.

Dirección General de Bibliotecas UAQ

## 2. MARCO TEÓRICO

En la última década se han desarrollado modelos para realizar predicciones de la concentración de contaminantes basados en Inteligencia Artificial, con el objetivo de mejorar su certeza, las técnicas que se han utilizado son redes neuronales artificiales (Moustris et al 2009, Alkasassbeh et al 2013, Cortina-Januchs et al 2015), lógica difusa de tipo 2 (Zarandi et al 2012), Algoritmo de optimización de Colonia de Hormigas (Aceves-Fernández et al, 2015), éste último arrojó resultados alentadores acerca del funcionamiento del algoritmo de Colonia de Hormigas con un 90% de certeza en el mejor caso.

El algoritmo de Colonia de Hormigas pertenece a los algoritmos de Swarm Intelligence o Inteligencia de Enjambre (Beni, 1989) en donde un conjunto de agentes trabajan interactuando entre sí y su medio para lograr un comportamiento integral con cierto grado de complejidad que no hubiera podido lograr de manera individual cada agente y así encontrar soluciones como conjunto, estos algoritmos están inspirados principalmente en comportamientos naturales como una colonia de hormigas, colonia de abejas o de una colonia de bacterias.

En este trabajo se va a implementar el algoritmo de Optimización de Forrajeo bacteriano (Bacterial Foraging Optimization o BFO), el cual está inspirado en la manera en que la bacteria E. Coli realiza su proceso de alimentación y reproducción (Passino, 2002), para crear un modelo para buscar una predicción eficaz sobre el comportamiento del material particulado.

El algoritmo de optimización BFO se considera un algoritmo de optimización global ya que en los trabajos en donde se ha aplicado se ha demostrado su eficiencia en diversas áreas, por ejemplo su aplicación en el campo en la ingeniería eléctrica y de control donde ha sido aplicado frecuentemente, como en encontrar la ubicación y control óptimos de un controlador de flujo de potencia unificado (Tripathy, 2007), así mismo en la misma área, para desarrollar un controlador de frecuencia de carga (Ali, 2011), para resolver problemas de programación de horarios (Wu et al 2007),

en el reconocimiento de patrones (Dasgupta et al 2008), otra aplicación ha sido en la predicción de los índices bursátiles, en donde se probó que el algoritmo de BFO tiene mayor eficiencia que los algoritmos genéticos utilizados en el mismo problema (Majhi, 2009), así mismo, se ha utilizado en la planificación de una red de comunicación de Identificación por Radiofrecuencia (RFID) que fuera precisa y confiable, en dicho trabajo el desempeño de BFO fue comparado con algoritmos genéticos y Optimización de enjambre de partículas (PSO), y cuyos resultados mostraron la superioridad de BFO (Chen, 2010), entre otros. Es por esto, que el algoritmo de BFO ha ganado popularidad, ya que ha mostrado mejores tener mejor desempeño en diversas áreas, incluso comparado con respecto a otros algoritmos de inteligencia de enjambre.

El utilizar un algoritmo que cuenta con varios agentes o de inteligencia de enjambre como lo es el algoritmo BFO nos da la oportunidad de encontrar una solución óptima ya que se trata de varios agentes, relativamente simples, explorando el área de estudio, teniendo así mayor probabilidad de encontrar los valores óptimos globales evitando así el problema de estancarse en una solución local como sucede con otros métodos como las redes neuronales, además de que son sistemas robustos, flexibles y sin un control central que emita órdenes a los agentes del sistema (Yang, 2013).

En la tabla 2.1 se presenta una lista de los trabajos hechos en el área del análisis del comportamiento de las partículas atmosféricas utilizando técnicas de inteligencia artificial, entre ellas se encuentran las Redes neuronales, lógica difusa y el algoritmo de Colonia de Hormigas (ACO), éste último pertenece a los algoritmos de inteligencia de Enjambre al igual que el algoritmo BFO, el cual será aplicado en este trabajo para obtener un modelo para las partículas PM10, cabe mencionar que no se ha encontrado un trabajo parecido en la literatura en el que se haya aplicado en el problema de modelado del comportamiento de partículas atmosféricas.

Título	Autor	Año	Descripción	Resultados
Air pollution modelling with the aid of computational intelligence methods in Thessaloniki, Greece. Simulation Modelling Practice and Theory	Karatzas, K. D, et al	2007	Redes neuronales artificiales, análisis de componentes principales (PCA)	Valores de ozono con 24 h de anticipación con un 83% de certeza
3-Day-Ahead Forecasting of Regional Pollution Index for the Pollutants NO <sub>2</sub> , CO, SO <sub>2</sub> , and O <sub>3</sub> Using Artificial Neural Networks in Athens, Greece	Moustris K., et al	2009	Artificial Neural Network (ANN), Dioxido de Nitrógeno(NO <sub>2</sub> ), Monóxido de Carbono CO, dióxido de azufre(SO <sub>2</sub> ), Ozono(O <sub>3</sub> )	Predicciones de 3 días de anticipación
Forecasting concentrations of air pollutants by logarithm support vector regression with immune algorithms.	Lin, K. P., et al	2011	Regresión de Vectores de Soporte con algoritmos inmunes.	partículas (PM <sub>10</sub> ), óxido de nitrógeno, (NO <sub>x</sub> ), y dióxido de nitrógeno (NO <sub>2</sub> ), se desempeñó mejor que las redes neuronales
Forecasting PM <sub>10</sub> in metropolitan areas: Efficacy of neural networks	Fernando, H. J., et al	2012	Redes neuronales artificiales	Predicciones con un 25% de error
Prediction of PM <sub>10</sub> and TSP Air Pollution Parameters Using Artificial Neural Network Autoregressive, External Input Models: A Case Study in Salt, Jordan	Alkasassbeh M., et al	2013	Artificial Neural Network (ANN), Material Particulado < 10 $\mu$ (PM <sub>10</sub> ), Total Suspended Particles (TSP)	Modelo: red neuronal autorregresiva(ANN), Parámetros utilizados: Concentraciones de contaminantes, velocidad del viento, temperatura y humedad relativa

Forecast Urban Air Pollution in Mexico City by Using Support Vector Machines: A Kernel Performance Approach	Sotomayor-Olmedo, A, et al	2013	support vector machines	Ozono (O3), partículas (PM10) y dióxido de nitrógeno (NO2), predicciones con certeza entre 80% y 87%
Analysis of Key Features of Non-Linear Behaviour Using Recurrence Quantification. Case Study: Urban Airborne Pollution at Mexico City	Marco A. Aceves-Fernandez, et al	2013	Recurrence analysis, Ozono(O3), Monóxido de Carbono CO Dioxido de Nitrogeno(NO2) dióxido de azufre(SO2), Material Particulado < 10µ (PM10)	Extracción de información sobre el comportamiento de las partículas de grandes volúmenes de datos
Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation.	Feng, X., et al	2015	Redes neuronales artificiales, PM 2.5	Efectivo para la predicción de concentraciones de PM 2.5
Artificial intelligence based approach to forecast PM2.5 during haze episodes: A case study of Delhi, India. Atmospheric Environment	Mishra, D., et al	2015	Red neuronal artificial, lógica neuro-difusa, regresión lineal múltiple	Las redes neuronales se desempeñaron mejor que la regresión lineal múltiple
Development of a model for forecasting of PM <sub>10</sub> concentrations in Salamanca, Mexico	Cortina-Januchs M. G., et al.	2015	Partículas: (PM10) artificial neural networks combined with clustering algorithms k-means fuzzy c means Neural networks	Modelo para predecir concentraciones de PM10 en las próximas 24 horas: Regresión lineal múltiple Parámetros utilizados: Concentraciones de PM10, dirección del

				viento, velocidad del viento, temperatura y humedad relativa
Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models	Aceves-Fernandez M.A. , et al	2015	Método: Ant Colony Partículas: Ozono(O3), Dioxido de Nitrogeno(NO2 ) Material Particulado< 10µ (PM10)	Predicción con precisión de 93% para PM10 72% para NO2 37% para O3
A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine.	Wang, D., et al	2017	Optimizado con evolución diferencial (DE) Extreme Learning Machine (ELM)	Mostro eficiencia sobre otros métodos para el índice de Calidad del aire (air quality index, AQI)
A deep cnn-lstm model for particulate matter (PM2. 5) forecasting in smart cities	Huang, C. J., et al	2018	Red neuronal convolucional (CNN) y Gran Memoria de Corto Plazo (LSTM)	Pronóstico de la concentración de PM2.5 de la siguiente hora según la concentración de PM2.5, la velocidad del viento acumulada y las horas de lluvia acumuladas durante las últimas 24 h

FuSSFFra, a fuzzy semi-supervised forecasting framework: the case of the air pollution in Athens	Bougoudis, I., et al	2018	Lógica difusa, Aprendizaje semi-supervisado, Framework de predicción Ozono(O3), Monóxido de Carbono CO Dioxido de Nitrogeno(NO2 ) dióxido de azufre(SO2)	Pronóstico exitoso de valores extremos para algunos contaminantes
Air pollution prediction with clustering-based ensemble of evolving spiking neural networks and a case study on London area.	Maciąg, P. S., et al	2019	Red neuronal de impulsos, Ozono , PM10, Clustering	Predicción de series de 12 horas para PM10
Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts	Zhou, Y., et al	2019	Multi-output LSTM Deep learning	Predicción con un 2.18% a 13.91% de error para PM10 y PM2,5
An improved particle swarm optimization (PSO): method to enhance modeling of airborne particulate matter (PM10)	Ordóñez-De León, B., et al	2019	Particle Swarm Optimization	

**Tabla 2.1** Estado del arte de Inteligencia Artificial en aplicaciones ambientales



### 3. HIPÓTESIS Y OBJETIVOS

#### 3.1 Hipótesis

El algoritmo de optimización de forrajeo bacteriano será capaz de encontrar un mejor modelo para predecir el comportamiento de las partículas atmosféricas, ya que al tratarse de un algoritmo que trabaja con una población de agentes que trabajan de manera colectiva para encontrar la mejor solución, así mismo es pertinente mencionar que el algoritmo ha demostrado tener mejores resultados que otras técnicas de inteligencia artificial en otras aplicaciones.

#### 3.2 Objetivo General:

Generar un modelo capaz de predecir el comportamiento de las partículas atmosféricas de manera certera y que mejore los resultados obtenidos con las técnicas y algoritmos que ya han sido implementados para este problema.

#### 3.3 Objetivo Particular:

- Obtener/adquirir los datos que serán utilizados como datos de entrenamiento y procesarlos para que sean aptos para ser analizados con el algoritmo de BFO.
- Implementar el algoritmo de Bacterial Foraging utilizando la variante que ha mostrado mejores resultados.
- Crear modelo de predicción basado en método.
- Comparar resultados con otros métodos aplicados anteriormente en el mismo problema.

## 4. MATERIALES Y MÉTODOS

### 4.1 Datos

Los datos utilizados para construir el modelo fueron tomados de la base de datos del Sistema de Monitoreo Atmosférico (SIMAT), institución responsable de la medición de los niveles de contaminantes atmosféricos en la Ciudad de México y Área Metropolitana (SIMAT, 2019). El SIMAT tiene a su vez un subsistema, la Red Automática para el Monitoreo Atmosférico (RAMA), que cuenta con equipo de medición continua de contaminantes atmosféricos, dióxido de azufre, monóxido de carbono, dióxido de nitrógeno, ozono, PM10 y PM2.5. Así mismo se tomaron datos de la Red de Meteorología y Radiación Solar (REDMET) que también forma parte del SIMAT, de este subsistema se utilizan los datos de Temperatura (TMP), humedad relativa(RH), dirección del viento (WDR) y Velocidad del viento (WSP), estos datos forman parte de los factores para modelar la concentración del contaminante(REDMET,2019).

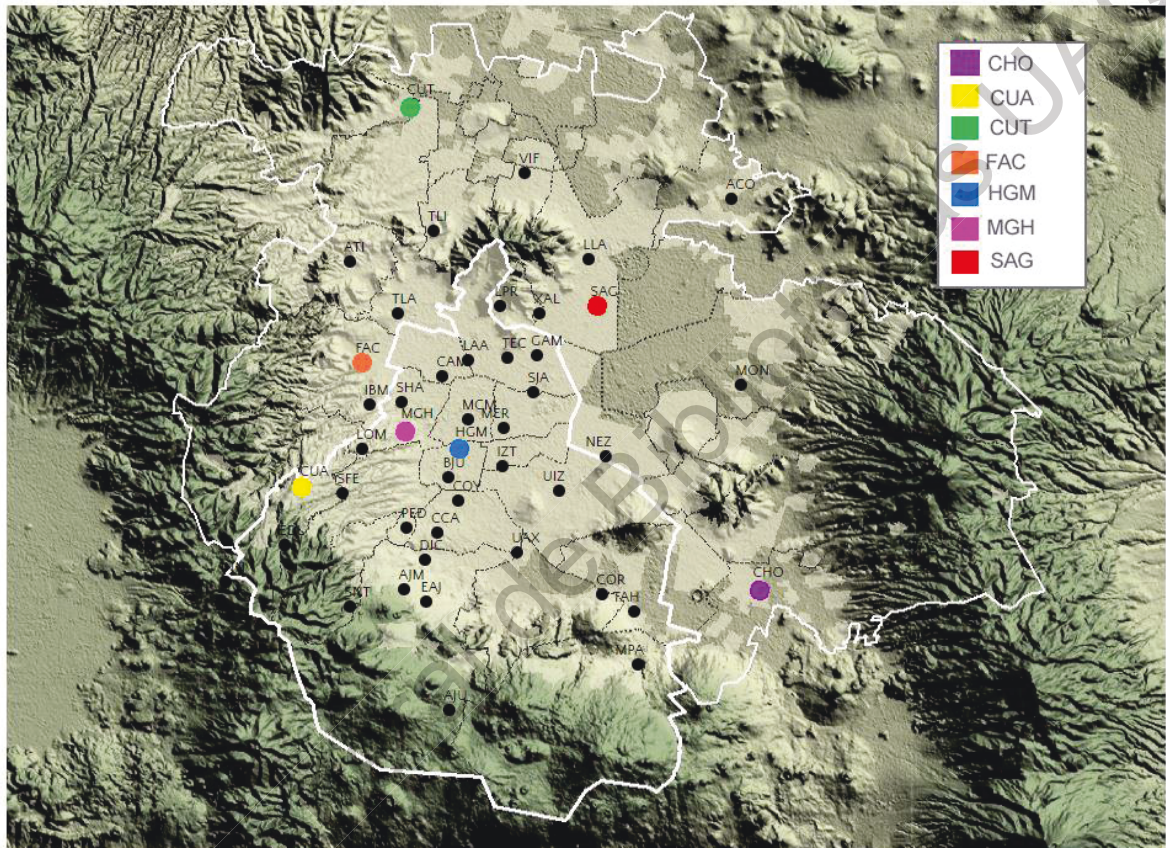
En la tabla 4.1 se encuentra la información de los datos usados para construcción del modelo y sus unidades de medición.

Dato	Unidad de medición	Subsistema
PM10	µg/m <sup>3</sup> .	RAMA
TMP	Grados Celsius (°C)	REDMET
RH	Porcentaje (%)	REDMET
WDR	Grados Azimut	REDMET
WSP	Metros/segundo(m/s)	REDMET

**Tabla 4.1.** Datos para generar el modelo

El SIMAT tiene distribuidas las estaciones de monitoreo por diferentes zonas de la ciudad, pertenecen al subsistema RAMA, estas estaciones de monitoreo recolectan

información de las concentraciones de los contaminantes y condiciones atmosféricas cada hora.



**Figura 4.1.** Mapa de las estaciones de RAMA, Fuente: (SIMAT,2018)

En la figura 4.1 se pueden observar las estaciones de monitoreo. El criterio para la selección de las estaciones cuyos datos se utilizaron para la construcción del modelo, fue tomando en cuenta aquellas que tenían mayor cantidad de datos válidos para el año 2016, y las estaciones se listan en la tabla 4.2.

Clave	Nombre	Alcaldía o municipio	Entidad
<b>CHO</b>	Chalco	Chalco	Estado de México
<b>CUA</b>	Cuajimalpa	Cuajimalpa de Morelos	CDMX
<b>CUT</b>	Cuautitlán	Tepotzotlán	Estado de México
<b>FAC</b>	FES Acatlán	Naucalpan de Juárez	Estado de México
<b>HGM</b>	Hospital General de México	Cuauhtémoc	CDMX
<b>MGH</b>	Miguel Hidalgo	Miguel Hidalgo	CDMX
<b>SAG</b>	San Agustín	Ecatepec de Morelos	Estado de México

Tabla 4.2. Estaciones de Monitoreo

## 4.2 Construcción del modelo

El planteamiento de este trabajo es la optimización de un modelo existente (modelo base), aplicando el algoritmo de optimización de forrajeo bacteriano (BFOA, por sus siglas en inglés) como método de optimización.

La idea principal de la optimización del modelo utilizando BFOA, es tomar un modelo ya existente (ANFIS) cuya precisión puede ser mejorada utilizándolo como punto de partida, es decir, se usan los datos generados por el modelo ANFIS, a partir de este modelo se generan de forma aleatoria las posibles soluciones al problema y el algoritmo optimizador BFO se encarga de ir buscando las soluciones que se desempeñen mejor para nuestro modelo.

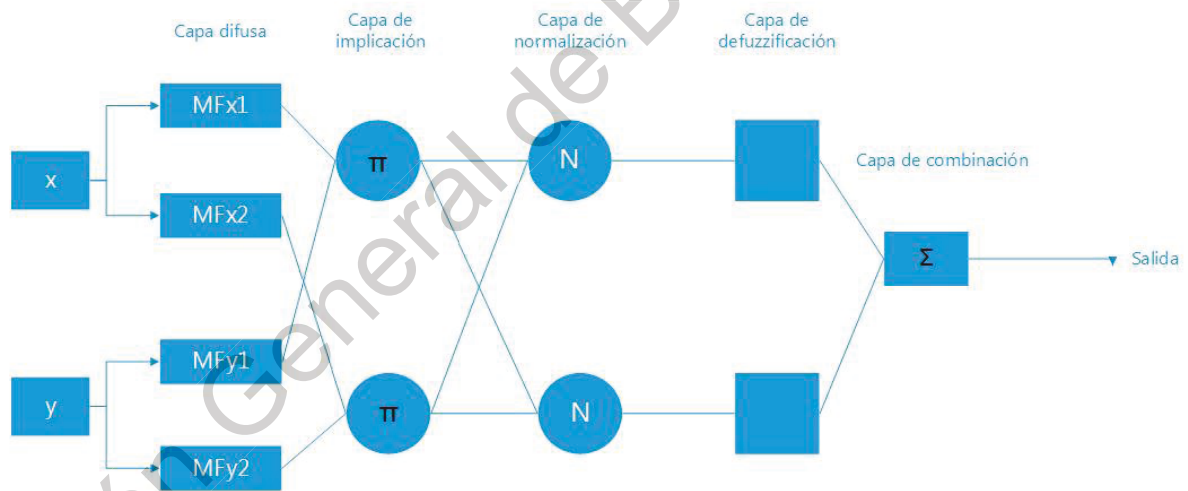
El modelo fue validado utilizando los datos de las mismas estaciones del año 2015 y del año 2017 para evaluar el desempeño del modelo.

### 4.2.1 Sistema adaptativo de inferencia neurodifusa (ANFIS)

La propuesta de modelo base es un modelo generado por un Sistema Adaptativo de Inferencia Neurodifusa, ANFIS (por sus siglas en inglés). ANFIS es un tipo de red neuronal artificial que contiene un sistema de inferencia difusa Takagi-Sugeno.

El Sistema de inferencia difusa (FIS) se construye dado un conjunto de datos de tipo entradas/salida, así mismo contiene funciones de membresía propias de los sistemas difusos y cuyos parámetros se ajustan utilizando un algoritmo de retropropagación, que puede ser usado en combinación con un método de mínimos cuadrados (Jang,1993).

Un FIS puede definirse como un conjunto de reglas difusas de tipo SI-ENTONCES (if- then), estas reglas tienen la forma SI  $A$  ENTONCES  $B$ , donde  $A$  y  $B$  son etiquetas de conjuntos difusos (Jang,1993). Estas reglas difusas encapsulan la forma imprecisa en que los humanos toman decisiones, como se muestra en la figura 4.2.



**Figura 4.2.** Estructura de ANFIS. Fuente: adaptado de (Abdulshahed,2015)

Tomaremos como ejemplo un conjunto de datos con dos entradas ( $x$ ,  $y$ ) y una salida  $f$ . Así mismo ANFIS utiliza 5 capas para construir el modelo, cada una de estas capas contiene varios nodos, algunos son adaptativos (nodos cuadrados en la figura 2) o fijos (nodos circulares) (Abdulshahed,2015).

#### 4.2.1.1 Capa 1

Esta se conoce como la capa difusa y se encarga de convertir las entradas del modelo en conjuntos difusos a través de las funciones de membresía (MF), las funciones del nodo se describen de la siguiente manera

$$O_{1,i} = \mu_{A_i}(X_1) \text{ para } i = 1,2 \quad (4.1)$$

$$O_{1,i} = \mu_{B_{1-2}}(Y_1) \text{ para } i = 3,4 \quad (4.2)$$

Donde  $X_1$  y  $Y_1$  son los nodos de entrada, A y B son las etiquetas lingüísticas asociadas a los nodos de entrada, y  $\mu(X_1)$  y  $\mu(Y_1)$  son las funciones de membresía (MF), existen diferentes tipos de funciones de membresía. Los parámetros en esta capa se llaman parámetros de premisa (Singh, 2012).

#### 4.2.1.2 Capa 2

Los nodos en esta capa son fijos y se etiquetan con  $\Pi$ , la función en cada nodo se multiplica por las señales de entrada, esto produce una señal de salida.

Es representado por la siguiente fórmula:

$$O_{2,i} = w_i = \mu_{A_i}(X_1) \cdot \mu_{B_{1-2}}(Y_1) \text{ para } i = 1,2 \quad (4.3)$$

Donde  $O_{2,i}$  es la salida de esta capa y la señal de salida  $w_i$  representa la fuerza de disparo de la regla. La fuerza de disparo se define como el peso de cada regla.

#### 4.2.1.3 Capa 3.

En esta capa los nodos también son fijos, están etiquetados con N, y su función es normalizar la fuerza del disparo calculando la proporción de la fuerza de disparo del nodo  $i$ -ésimo para sumar la fuerza de disparo de todas las reglas.

$$O_{3,i} = \bar{w} = \frac{w_i}{w_1 + w_2} \text{ para } i = 1,2 \quad (4.4)$$

Donde  $O_{3,i}$  es la salida de la capa 3 y  $\bar{w}$  es la fuerza de disparo pero ya normalizada.

#### 4.2.1.4 Capa 4

Los nodos de esta capa se adaptan y se definen como

$$O_{4,i} = \bar{w}_i \cdot f_i \text{ para } i = 1,2 \quad (4.5)$$

Donde  $f_1$  y  $f_2$  son las reglas difusas SI- ENTONCES definidas de la siguiente manera:

- Regla 1. SI  $X_1$  es  $A_1$  y  $Y_1$  es  $B_1$ , ENTONCES  $f_1 = p_1 X_1 + q_1 Y_1 + r_1$  (4.6)
- Regla 2. SI  $X_1$  es  $A_2$  y  $Y_1$  es  $B_2$ , ENTONCES  $f_2 = p_2 X_1 + q_2 Y_1 + r_2$  (4.7)

Donde  $p_i, q_i$  y  $r_i$  son parámetros establecidos y referidos como parámetros consecuentes.

#### 4.2.1.5 Capa 5,

En esta capa los nodos son fijos, están etiquetados con  $\Sigma$  y su función es calcular la salida total y que se define por:

$$O_{5,i} = \sum_i \bar{w}_i \cdot f_i = \frac{\sum_i w_i f_i}{w_i} = f_{out} = \text{salida total} \quad (4.8)$$

ANFIS tiene una regla de aprendizaje muy sencilla, que es la "propagación hacia atrás", esta regla calcula las señales de error de forma recurrente, comenzando desde la capa de salida (Capa 5) hacia los nodos de entrada (Capa 1).

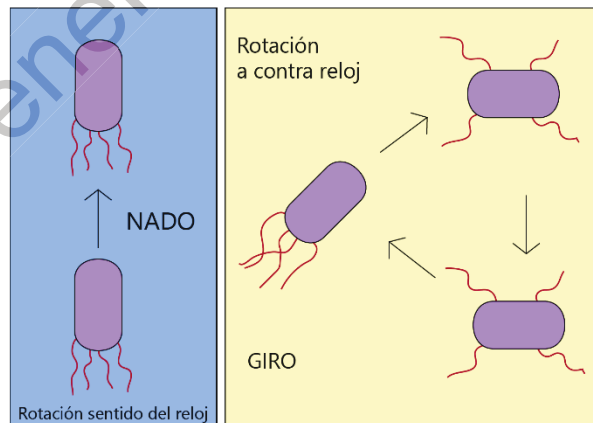
### 4.3 Optimización del modelo

La parte de optimización del modelo se llevará a cabo con el algoritmo de optimización forrajeo bacteriano BFOA (por sus siglas en inglés). BFOA es un algoritmo que simula el proceso de forrajeo de una bacteria real, la cual logra su locomoción en el medio ambiente a través del movimiento de sus flagelos, este

movimiento genera las operaciones básicas del forrajeo que son el nado y el giro. Para realizar la operación de giro, las bacterias giran sus flagelos en el sentido de las manecillas del reloj, si el ambiente es nocivo, la bacteria va a dar más giros para encontrar nutrientes, en cambio si los flagelos giran en el sentido contrario a las manecillas del reloj la bacteria va a realizar la operación de nado o avance, si el ambiente es propicio para la bacteria entonces el movimiento de nado abarcará mayores distancias, (Hansen,2008).

Las operaciones de nado y giro forman parte del proceso de quimiotaxis (figura 4.3), en el cual la bacteria va a buscar moverse en un ambiente con nutrientes suficientes, evitando zonas nocivas.

Si el ambiente tiene suficientes nutrientes y una temperatura adecuada, entonces la bacteria se reproducirá dividiéndose en dos, creando una réplica de sí misma, formando así una colonia de bacterias. Sin embargo, si la colonia se ve amenazada por un ataque o hay un cambio en el ambiente, como el que los nutrientes se estén terminando, entonces un grupo de bacterias será dispersado hacia otras áreas del ambiente o son eliminadas, este evento es llamado eliminación-dispersión.



**Figura 4.3.** Quimiotaxis de una bacteria traducido de (Das, 2009)



Supongamos que tenemos un problema en donde buscamos encontrar el mínimo de  $J(\theta)$  donde  $\theta \in \mathbb{R}^p$  (en donde  $\theta$  es un vector  $p$ -dimensional) pero ignoramos la naturaleza del gradiente  $\nabla J(\theta)$  porque no tenemos una descripción analítica o mediciones de  $\nabla J(\theta)$ .

Entonces el algoritmo BFO es una implementación artificial de los mecanismos principales de una colonia real de bacterias *E. Coli*, los cuales son: quimiotaxis, formación de la colonia o proceso de enjambre, reproducción y eliminación-dispersión, con los cuales se aspira a resolver el problema de optimización sin gradiente. Para explicar concretamente en qué consiste una bacteria virtual dentro de BFOA, es que representa una solución de prueba, que se va mover en el espacio de búsqueda para ubicar el óptimo global (Das, et al. 2009).

Para la implementación del algoritmo es necesario definir algunos parámetros siendo estos:

$p$ : Dimensión del espacio de búsqueda

$S$ : Bacterias que forman la población

$N_c$ : Pasos quimiotácticos

$N_s$ : Longitud de nado

$N_r$ : pasos de reproducción

$N_e$ : Eventos de eliminación dispersión

$P_{ed}$ : Probabilidad de que una bacteria sea eliminada o dispersada

$C(i)$ : Tamaño del paso tomado en una dirección aleatoria especificada por el giro.

Si un paso quimiotáctico se define como un giro seguido de otro giro, o un giro seguido de un nado, entonces  $j$  es el índice del paso quimiotáctico,  $k$  será el índice del paso de reproducción, y  $l$  es el índice del evento de eliminación dispersión.

Entonces digamos que  $P(j, k, l) = \theta^i(j, k, l)$  donde  $i = 1, 2, \dots, S$  es la posición de cada miembro en la población de  $S$  bacterias que se encuentran en el paso quimiotáctico  $j$ -ésimo, el paso  $k$ -ésimo de reproducción y el evento  $l$ -ésimo de eliminación-dispersión, entonces a cada posición  $\theta^i(j, k, l)$  se le puede asociar un costo  $J(j, k, l)$  que se busca optimizar. Cada etapa de BFOA se describe brevemente a continuación.

#### 4.3.1.1 Quimiotaxis:

Anteriormente se estableció que  $\theta^i(j, k, l)$  donde  $i = 1, 2, \dots, S$  es la posición de cada miembro de la población de  $S$  bacterias en el paso quimiotáctico  $j$ -ésimo, el paso  $k$ -ésimo de reproducción y el evento  $l$ -ésimo de eliminación-dispersión, así mismo  $C(i)$  es el paso tomado en una dirección aleatoria es decir, es lo que especifica el giro, entonces el movimiento de quimiotaxis artificial se representa con:

$$\theta^i(j, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (4.9)$$

Donde  $\Delta$  es un vector que contiene una dirección aleatoria cuyos elementos se encuentran entre  $[-1, 1]$ .

#### 4.3.1.2 Enjambre:

Las células reales responden a estímulos químicos para formar grupos de células o enjambres, y así viajar como colonia en el ambiente, cuando las células son estimuladas por un alto nivel de succinato, entonces lanzan un aspartato que es un atrayente que ayuda a que formen el enjambre en patrones concéntricos con una alta densidad de bacterias. En las bacterias artificiales de BFOA las señales de

célula a célula para formar el enjambre y así lograr la inteligencia colectiva, se representan de la siguiente manera:

$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S [-d_{attractant} \exp(-w_{attractant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] + \sum_{i=1}^S [-h_{repellant} \exp(-w_{repellant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \quad (4.10)$$

Donde  $J_{cc}$  es la señal que se enviará entre las bacterias para formar el enjambre y su valor se agregará a la función objetivo que será minimizada, así mismo  $w_{attractant}$  es la cuantificación de la tasa de difusión del atrayente,  $d_{attractant}$  es una cuantificación del agente de atracción que se va a liberar, de la misma manera una célula repele a cualquier célula cerca en el sentido de que no es físicamente posible tener dos células en la misma ubicación para modelar esto se utiliza la altura del repelente  $h_{repellant}$  que es la magnitud de su efecto y cuyo valor se define como  $h_{repellant} = d_{attractant}$ , y sea  $w_{repellant}$  la medida de la tasa de difusión del repelente. Estos son coeficientes que deben escogerse de manera adecuada de acuerdo a nuestro espacio de búsqueda.

#### 4.3.1.3 Reproducción.

El criterio general para esta etapa es que las bacterias menos saludables deben morir mientras las bacterias más saludables, que son las que tienen un valor más bajo en la función objetivo, se van a reproducir dividiéndose en dos, manteniendo el tamaño de la población constante.

#### 4.3.1.4 Eliminación/dispersión

Para simular los eventos de dispersión y eliminación, se eliminan un grupo de bacterias de manera aleatoria con una pequeña probabilidad, y se inicializan los reemplazos de manera aleatoria sobre el espacio de búsqueda.

La figura 4.4 muestra el diagrama de flujo del algoritmo BFOA.

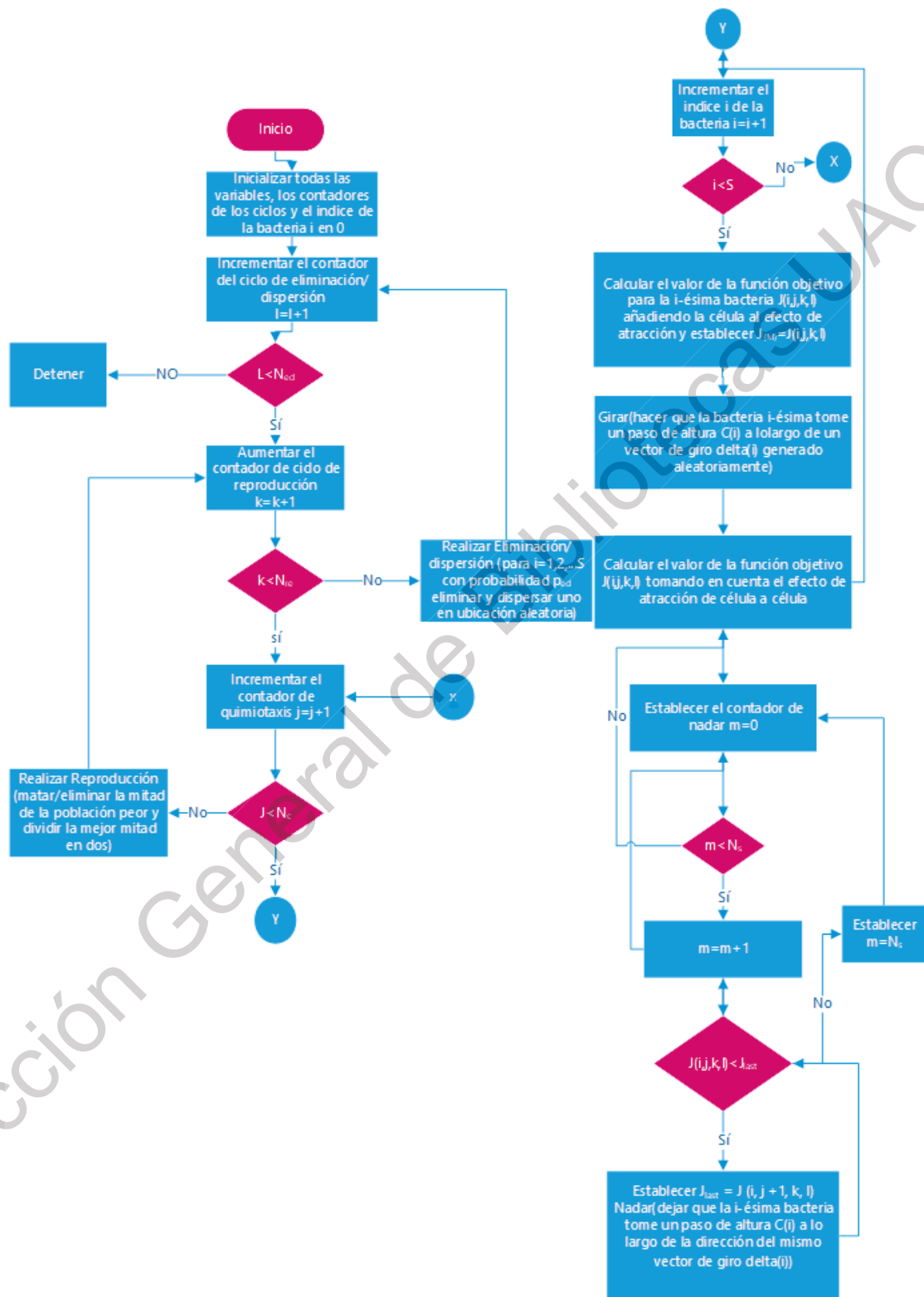


Figura 4.4. Diagrama de flujo BFOA.

#### 4.3.1.5 Pseudocódigo BFOA.

Paso 1.- Inicializar parámetros  $S, p, N_c, N_r, N_s, N_{ed}, P_{ed}, \theta^i, C(i)(i=1,2\dots S)$

Paso 2.- Ciclo de eliminación-dispersión:  $l=l+1$

Paso 3.- Ciclo de reproducción:  $k=k+1$

Paso 4.- Ciclo de quimiotaxis:  $j=j+1$

a) Para  $i=1,2\dots S$  tomar un paso quimiotáctico para la bacteria  $i$  de la siguiente manera.

b) Calcular costo  $J(i, j, k, l)$

Sea  $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta, P(j, k, l))$  donde  $J_{cc}$  es el perfil de atracción/repelente definido en (2).

c) Sea  $J_{last} = J(i, j, k, l)$  para guardar este valor, dado que podría encontrarse un costo mejor en la siguiente ejecución del algoritmo.

d) Giro: se genera un vector aleatorio  $\Delta(i) \in \mathbb{R}^p$  donde cada elemento  $\Delta_m(i), m = 1, 2, \dots, p$  es un número aleatorio entre  $[-1, 1]$

e) Movimiento: Sea

$$\theta^i(j, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

Lo cual resulta en un paso de tamaño  $C(i)$  en la dirección del giro para la bacteria  $i$

f) Se debe de calcular el costo  $J(i, j, k, l)$  siendo

$$J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta, P(j, k, l))$$

g) Nado

Contador para la longitud de nado  $m=0$

Mientras  $m < N_s$

$m=m+1$

si  $J(i, j + 1, k, l) < J_{last}$  entonces  $J_{last} = J(i, j + 1, k, l)$  y

$$\theta^i(j + 1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

Usar  $\theta^i(j + 1, k, l)$  para calcular  $J(i, j + 1, k, l)$  como en f)

De lo contrario  $m=N_s$ , fin del ciclo mientras.

h) Ir a la siguiente bacteria (i+1), ir a b)

Paso 5.- Si  $j < N_c$  ir al paso 4 dado que la bacteria no ha terminado su ciclo de vida

Paso 6.- Reproducción:

a) dados  $k$  y  $l$  para cada  $i=1, 2, \dots, S$  entonces

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l) \quad (4.11)$$

Representa la salud de la bacteria, es decir cuántos nutrientes obtuvo durante su ciclo de vida y que tanto éxito tuvo para evitar sustancias nocivas. Ordenar bacteria y parámetros de quimiotaxis en orden ascendente del costo  $J_{health}$ , en donde un mayor costo significa una salud mala.

b) Las bacterias  $S_r$  con el valor más alto de  $J_{health}$  mueren y las restantes con los mejores valores se dividen, creando copias y colocadas en el mismo lugar que las bacterias que les dieron origen.

Paso 7.- si  $k < N_r$  ir al paso 3, lo cual significa que no se han realizado los pasos de reproducción especificados en la inicialización de parámetros, así que se inicia la siguiente generación del ciclo de quimiotaxis.

Paso 8.- Eliminación-dispersión: para  $i=1, 2, \dots, S$  se tiene la probabilidad  $P_{ed}$  de eliminar/dispersar cada bacteria para mantener la población de bacterias constante.

Cuando una bacteria se elimina, se dispersa otra en una ubicación aleatoria en el dominio de optimización.

Si  $l < N_{ed}$ , entonces ir al paso 2, de lo contrario terminar.

#### 4.4 Metodología Propuesta

En la figura 4.5, se presentan los pasos que conforman la metodología para obtener el modelo optimizado, los cuales son los siguientes:

1. Adquisición de datos: Los datos en bruto necesarios para crear el modelo son obtenidos de una base de datos. El 10% de los datos totales se omitió del modelo de manera aleatoria para validar la precisión del modelo, pero sin caer en el overfitting del modelo o sesgar los resultados. Estos datos seleccionados al azar se eligieron teniendo en cuenta una distribución uniforme durante todo el mes para garantizar la coherencia de los datos de la prueba.
2. Filtrado de datos: La base de datos contiene algunos datos no válidos, o que pudieran ser parciales, y que deben de ser filtrados para no genera un modelo sesgado o incierto. En la figura 4.5 se observa una muestra de los datos usados para construir el modelo, en las celdas marcadas en rojo se ve el ejemplo de los datos que la estación no pudo registrar por alguna falla en el sistema de monitoreo y de manera automática le asigna un valor de -99 que debe ser filtrado ya que no corresponde a un dato real.

FECHA	HORA	RH	TMP	WDR	WSP	PM10		
13/01/2016	1	-99	-99	315	2.2	28		
13/01/2016	2	-99	-99	27	2	55	RH	Humedad Relativa
13/01/2016	3	-99	-99	13	2	59	TMP	Temperatura
13/01/2016	4	-99	-99	348	2	56	WDR	Dirección del viento
13/01/2016	5	-99	-99	336	2.3	56	WSP	Velocidad del viento
13/01/2016	6	-99	-99	135	2.7	54	PM10	Concentración PM10
13/01/2016	7	-99	-99	185	2.1	48		
13/01/2016	8	-99	-99	191	3.7	46		

Figura 4.5. Muestra de los datos

3. Entrada de datos: una vez que los datos han sido revisados y son válidos, entonces pueden ser alimentados al modelo. Este modelo preliminar o base es el generado por ANFIS. Para generarlo se utilizaron las aplicaciones Fuzzy Logic y Neuro Fuzzy Designer de Matlab® que contienen una interfaz gráfica para generar modelos basados en conjunto de datos de tipo entradas/salidas. En la figura 3.6 se muestran las interfaces del editor de funciones de membresía, de lógica difusa y la aplicación de Neuro fuzzy designer necesarios para construir el modelo desde las aplicaciones.

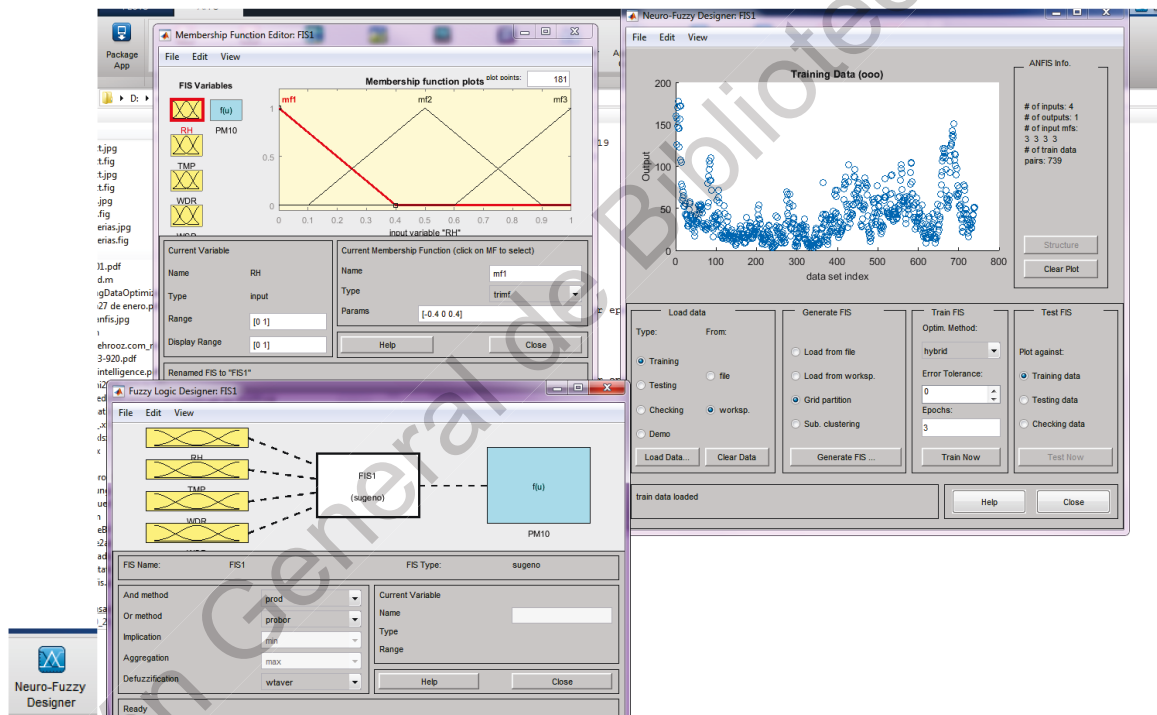


Figura 4.6. Herramienta Neuro-Fuzzy designer

Estas aplicaciones también contienen funciones para generar el modelo fuera de la interfaz gráfica del modelo FIS (generado por ANFIS).

4. Fuzzificación: Es el proceso de convertir en los datos de entrada en un valor lingüístico, este depende de las funciones de membresía. En este caso se utilizaron funciones de membresía generalizadas en forma de campana, figura 4.7.



La función de campana depende de tres parámetros a, b y c, y está dada por.

$$f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (4.12)$$

Evaluación de las reglas: las reglas del modelo son evaluadas con respecto a las reglas difusas y los valores de las funciones de membresía.

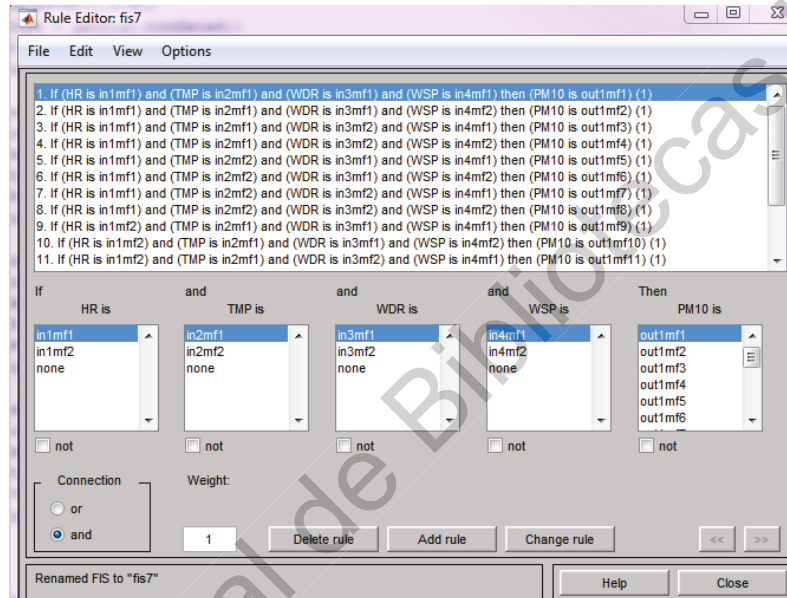
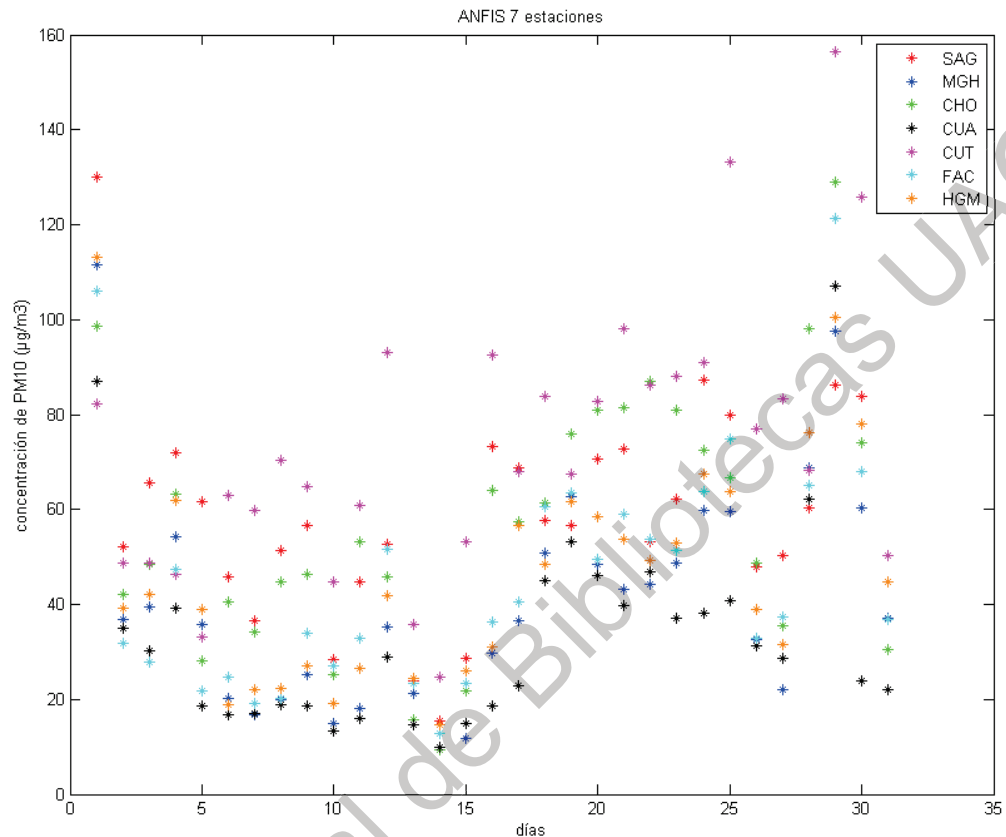


Figura 4.7. Reglas del modelo ANFIS

5. Defuzzificación: El método de defuzzificación utilizado fue el de promedio ponderado de todas las salidas de reglas.
6. Modelo Sistema de inferencia Difuso: Una vez que se completaron los pasos de evaluación y defuzzificación, el modelo se construye con las ecuaciones propias de ANFIS expresadas en la sección de construcción del modelo.
7. Definición del espacio de búsqueda: Un espacio de búsqueda se define como todas las soluciones factibles dentro de nuestro problema, es así que para encontrar el espacio de búsqueda en donde se va a optimizar el modelo utilizando

el BFOA tenemos que visualizar que en esta propuesta se busca minimizar la diferencia entre el modelo generado con ANFIS y los datos reales, es por esto que nuestro espacio de búsqueda debe de encontrarse primero, entre los datos válidos para los factores ambientales Humedad relativa, temperatura, dirección y velocidad del viento así como también dentro de los valores factibles de PM10, ya que dada la naturaleza del problema, otra forma de definir el espacio de búsqueda es muy difícil ya que las posibilidades son muy amplias.

8. Generación de la población: se deben de generar la población con S bacterias, en posiciones iniciales aleatorias dentro de un rango de posibles valores que puedan tener los datos reales para la hora en que se están generando las bacterias. El objetivo es realizar experimentos utilizando diferentes valores de S para determinar cómo afecta el tamaño de la población al modelo optimizado. Se utilizaron los modelos de ANFIS generados con los datos de siete estaciones que contaban con suficientes datos para construir y optimizar el modelo, se tomaron los datos del mismo periodo de tiempo y así generar el espacio de búsqueda donde las bacterias van a minimizar la diferencia, como se muestra en la figura 4.8. Por lo tanto, la generación de la población inicial de bacterias se hizo de manera aleatoria, tomando como referencia el rango de los datos de cada día del mes de las 7 estaciones.



**Figura 4.8.** ANFIS de 7 estaciones

9. Inicialización de parámetros: Los parámetros de BFOA se deben inicializar, estos parámetros incluyen inicializar los contadores de los ciclos de eliminación/dispersión  $j$ , reproducción  $k$ , quimiotaxis  $l$ , y el índice  $s$  de la bacteria  $i$ , así como los parámetros de atracción y repelente ( $w_{attractant}$ ,  $d_{attractant}$ ,  $h_{repellant}$ ,  $w_{repellant}$ ), que son los que generan el efecto de enjambre, en este caso se utilizaron los valores de  $d_{attractant} = 0.05$ ,  $w_{attractant} = 0.15$ ,  $h_{repellant} = d_{attractant}$ ,  $w_{repellant} = 10$

Estos valores fueron seleccionados de acuerdo a estudios realizados por otros autores (Passino, 2002) mismos que recomiendan se seleccionen, por ejemplo

la altura o magnitud del repelente debe ser igual al ancho de la señal de atracción para evitar tener dos bacterias en el mismo punto y esto se modela haciendo que  $h_{repellant} = d_{attractant}$ , por otro lado, también recomienda que la señal de atracción sea muy pequeña en comparación a los valores de concentración de nutrientes en nuestro espacio de búsqueda y por eso el repelente debe ser lo suficientemente grande para evitar que las bacterias estén muy cerca.

10. Cálculo de la función objetivo: Se debe de calcular el criterio que nos dirá que tan saludable está nuestra bacteria, en este algoritmo se agrega el cálculo del factor de enjambre  $J_{cc}$  con la ecuación (4.10) y agregarse al costo  $J$  de la bacteria en la posición actual.

Actualizar posición de las bacterias: Se hace una actualización de las bacterias de acuerdo al costo de la bacteria en la posición actual y comparando la posición siguiente, para lograr esto se calcula el giro de la bacteria y se realiza un nado en esa dirección, se calcula el costo de la nueva posición, si esta tiene menor costo entonces se convierte en la mejor posición de la bacteria y se sigue moviendo en esa dirección, de lo contrario si termina el ciclo de nado y no encontró una mejor posición significa que no está ubicada en un ambiente adecuado y continua con la siguiente bacteria.

11. Se realizan los eventos de reproducción, dispersión y eliminación en donde la mitad mejor de la población se reproduce, es decir se hace una copia exacta de las bacterias con menor costo total y la otra mitad es reemplazada por bacterias generadas aleatoriamente, al igual un grupo con probabilidad baja son dispersadas en el espacio de búsqueda de manera aleatoria.

12. Si los ciclos del algoritmo BFO han sido completados, se cumplen los criterios de optimización, es decir las bacterias han convergido a un valor para el dato actual, en este caso, se refiere al valor de la concentración de PM10 para esa hora, el criterio para determinar que las bacterias han convergido es por medio de los parámetro de Nre y Ned, que son los eventos que influyen en la convergencia del algoritmo (Brabazon, 2018). Lo siguiente es regresar al paso 8 pero tomando los datos correspondientes de la siguiente hora, así hasta que se terminen los datos del mes.
  
13. Modelo optimizado: si los ciclos del algoritmo BFO y de las horas del mes han sido completados y se cumplen los criterios de optimización se dice que se ha creado un Modelo Optimizado.

En la figura 4.9 se muestra el diagrama de flujo de la metodología propuesta para la obtención del modelo optimizado.

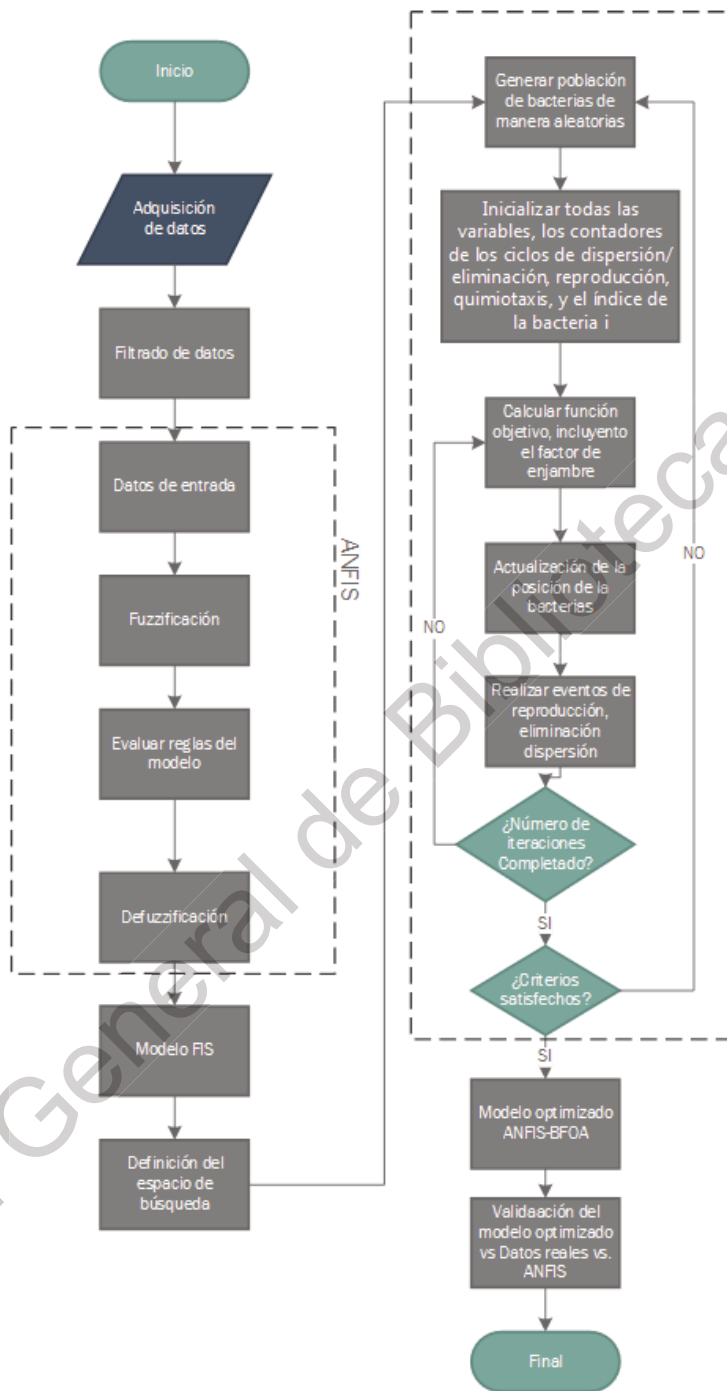
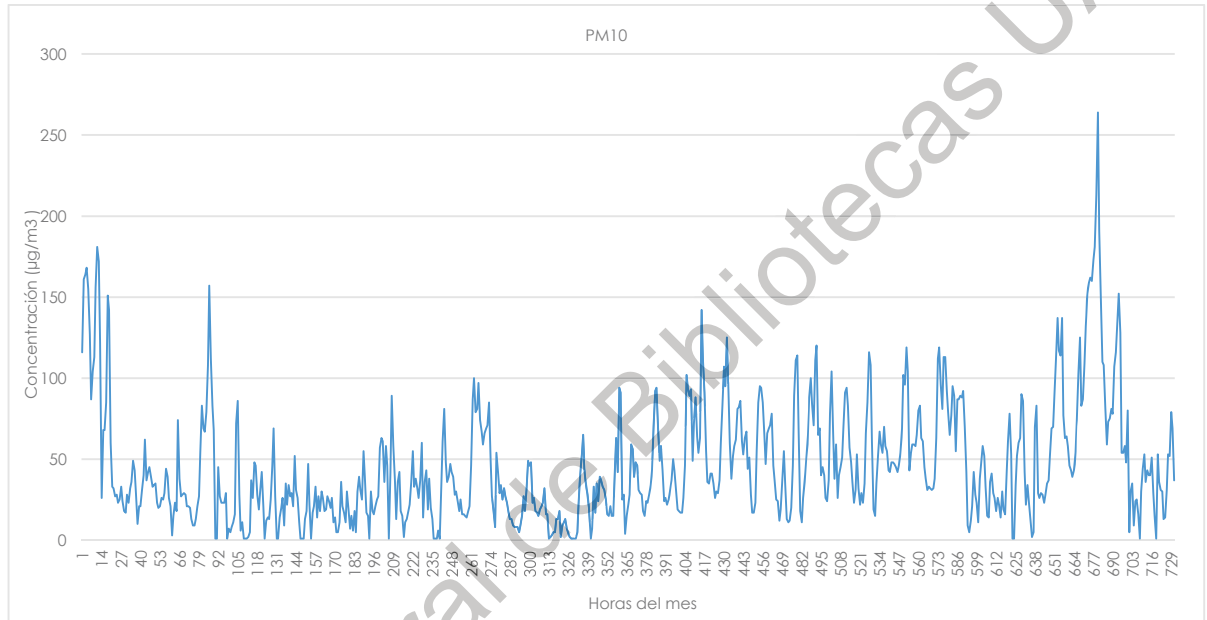


Figura 4.9. Metodología propuesta

## 5. RESULTADOS Y DISCUSIÓN

### 5.1 Resultados

Para ilustrar la complejidad del problema en la figura 5.1 se pueden observar los datos reales de concentración de PM10 y su comportamiento no lineal.



**Figura 5.1.** Concentración de PM10 enero 2016 estación FAC

Para explicar un poco más acerca de la complejidad de los datos es necesario hablar de la variabilidad de los datos provenientes de las diferentes estaciones de monitoreo, esta variabilidad tiene origen en la naturaleza del fenómeno del comportamiento de las partículas atmosféricas, en la tabla 5.1 se pueden observar algunas medidas de dispersión de tres estaciones, como muestra de esta variabilidad, en ella la desviación estándar nos indica que los datos se distribuyen en un rango más amplio.

Estación	Mínimo	Máximo	Rango	Desviación estándar	Media
hgm	5	196	191	30.74678334	46.325452
fac	1	264	263	36.87219649	45.9357045
sag	3	238	235	34.85784066	58.5607094

**Tabla 5.1.** Variabilidad de los datos de entrenamiento

La complejidad de los datos de entrenamiento, se puede observar fácilmente en las veces en que los factores atmosféricos pueden presentar valores muy alejados entre sí y aun así la concentración del contaminante, PM10, es la misma que con otra configuración de factores ambientales, esta variabilidad se presenta incluso en la misma estación, tabla 5.2, y así como también se presenta en la concentración de PM10 en la misma hora en diferentes estaciones, tabla 5.3.

FECHA	HORA	RH	TMP	WDR	WSP	PM10
01/08/2016	1	75	13.3	118	2	24
05/08/2016	20	77	17.1	319	3.7	24
06/08/2016	16	65	20.6	108	2.1	24
13/08/2016	13	39	22.9	327	1.6	24
13/08/2016	15	32	25.1	327	1.8	24
15/08/2016	6	83	12.2	107	1.2	24
15/08/2016	23	90	15.6	158	1	24
16/08/2016	21	91	15.6	125	0.8	24

**Tabla 5.2.** Variabilidad de los datos y sus factores ambientales

Estación	Fecha	Hora	HR	TEMP	WSP	WDR	PM10
sag	02/01/2015	1	75	17.4	95	1.2	50
mgh	02/01/2015	1	74	15.7	330	2.2	20
hgm	02/01/2015	1	86	14.3	98	0.4	1
fac	02/01/2015	1	88	14.2	214	1	17

**Tabla 5.3.** Variabilidad entre estaciones mismo día y hora

Una vez establecida la complejidad del problema, podemos expresar que el propósito de utilizar el método de optimización BFOA es la reducción del error que existe al aplicar el modelo creado con ANFIS, es por eso que los experimentos que se realizaron están orientados a probar la eficiencia del algoritmo BFO y las diferentes configuraciones de sus parámetros.



Es importante calcular el error que se obtiene al utilizar ANFIS para crear el modelo, el cual se puede observar en las figuras 5.1 y 5.2, y que posteriormente servirá para hacer la comparación con el modelo optimizado. La cuantificación del error entre los datos reales y los calculados por el modelo ANFIS se llevó a cabo utilizando la raíz del error cuadrático medio (RMSE por sus siglas en inglés) que es un método históricamente utilizado para medir la precisión de los pronósticos de datos (Hyndman, 2006).

En el caso del modelo generado con ANFIS se obtiene un  $RMSE = 24.147$ , el cual se espera será minimizado con BFOA, es por eso que se comparará con el modelo generado con ANFIS para establecer si el modelo optimizado con BFOA presenta mejores resultados con respecto a una técnica basada en redes neuronales y lógica difusa como lo es ANFIS.

En cuanto a la optimización del modelo, el objetivo es generar un modelo más preciso, y esto podría lograrse variando los parámetros propios del algoritmo de optimización de Forrajeo de Bacterias, como lo son el número de bacterias que se moverán en el espacio de búsqueda, la cantidad de pasos de quimiotaxis y de pasos de reproducción, sin embargo, la variación en los parámetros de BFOA podría generar un alto costo computacional, dada la naturaleza de BFOA, ya que es un algoritmo que cuenta con ciclos anidados, es por eso que también se tomó en cuenta en los experimentos el tiempo de procesamiento de cada configuración de parámetros. Como contexto para las pruebas de tiempo de ejecución, se utilizó una PC con Windows 7, de 64 bits, Procesador Intel Core i3-2100 3.10 GHz y memoria RAM de 12Gb.

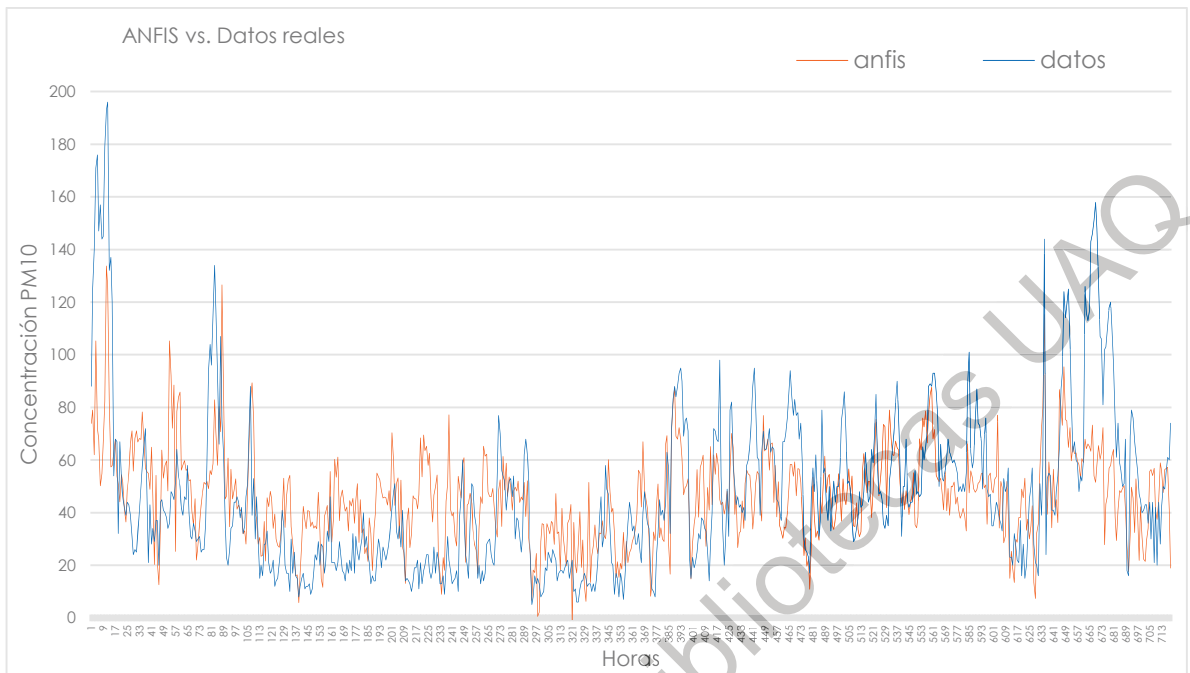


Figura 5.2. Comparación datos reales vs ANFIS

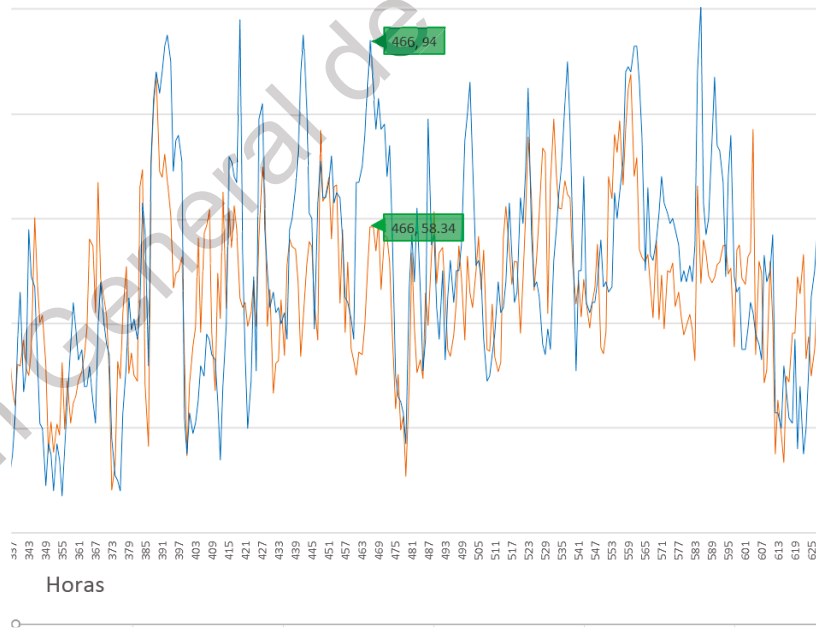
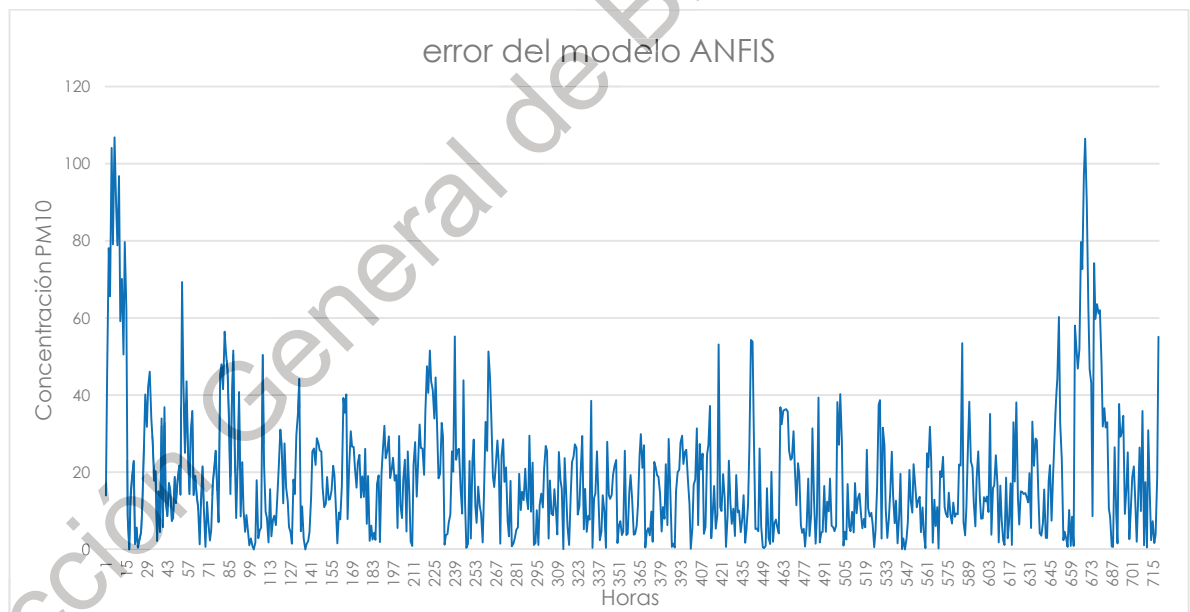


Figura 5.3. Detalle del error del modelo generado con ANFIS (creación propia)

El problema de usar ANFIS para generar un modelo de concentración del contaminante son que los valores obtenidos con ANFIS, presentan grandes diferencias en comparación con los valores reales, en la figura 5.3, se observa un acercamiento de la figura 5.2, para poder observar a detalle como en la hora 466 para el modelo ANFIS (línea naranja) la concentración es de 58.34  $\mu\text{g}$ , sin embargo, la concentración real (línea azul) fue de 94  $\mu\text{g}$ , el error de la predicción ya no es desestimable, y es pertinente notar que este fenómeno no se presenta exclusivamente en esa hora, en las horas 13, 469, 601, y de la hora 661 a la 686 se observan un errores de grandes proporciones, esto, sin tomar en cuenta en donde el error es moderado aunque no insignificante, el error neto para cada una de las horas del mes también se puede apreciar en la figura 5.4.



**Figura 5.4.** Error del modelo generado con ANFIS.

### 5.1.1 Variación del número de bacterias

El número de bacterias en la población S es tal vez el primer parámetro en el cual se piensa a la hora de elegir qué configuración se le dará al algoritmo dado que

cada una de las bacterias representa una posible solución al problema de optimización.

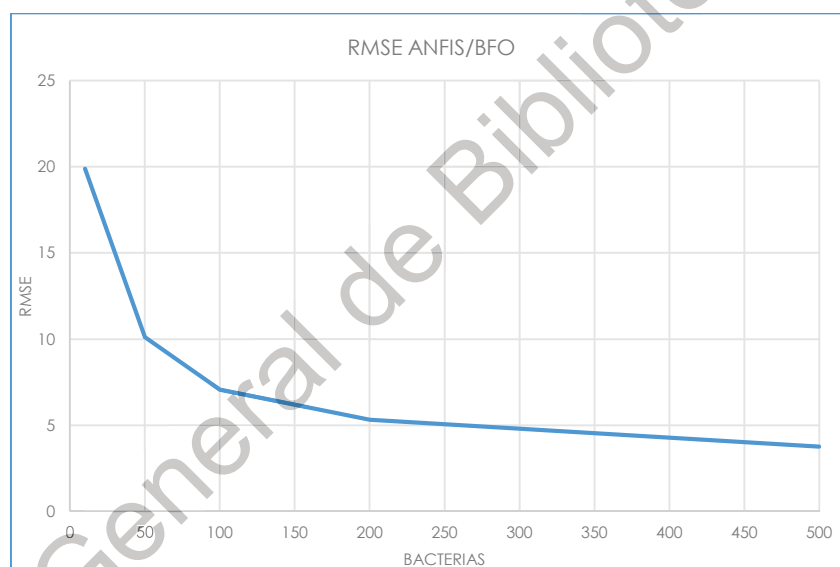
Aunque hay que tomar en cuenta que al aumentar el tamaño de  $S$  también puede aumentar el tiempo de ejecución. Sin embargo, si  $S$  tiene un valor grande y al distribuir la población inicial de manera aleatoria en el espacio de búsqueda, se tiene una probabilidad mayor de que algunas de esas bacterias hayan quedado posicionadas cerca de un punto óptimo, y que durante la ejecución del algoritmo es también mayor la probabilidad de que haya una mayor densidad de bacterias en dicha región óptima, ya sea porque se desplazaron ahí por medio del proceso de quimiotaxis o porque se reprodujeron otras bacterias que ya estaban ahí desde el inicio.

Los valores propuestos para experimentar con el número de bacterias fueron  $S = \{10, 50, 100, 200, 500\}$ , en la tabla 5.4 se puede observar cómo la cantidad de bacterias influye la obtención de un modelo optimizado al obtener una medición menor del error (RMSE), no obstante, es importante señalar cómo el efecto que tiene el tamaño de la población  $S$  en la obtención del error tiene una razón de cambio mayor cuando la población es menor a 50 y después de ese valor la razón de cambio va disminuyendo y va llegando a su límite, figura 5.5, en la configuración mostrada en la tabla es de  $RMSE \approx 3.75$ .

Así mismo, se puede observar que por la naturaleza de BFOA, ya que es un algoritmo que cuenta con ciclos anidados, el tiempo de ejecución aumenta de manera considerable, y aunque el error se ve reducido al aumentar el tamaño de la población, figura 5.6, esta diferencia en la optimización no parece importante, tomando en cuenta los recursos computacionales que requiere su cálculo.

Bacterias	RMSE	tiempo/seg
10	19.8851195	419.84
50	10.1072	2268.786
100	7.07398003	4185.117
200	5.32684396	8062.675
500	3.75769854	20625.8073

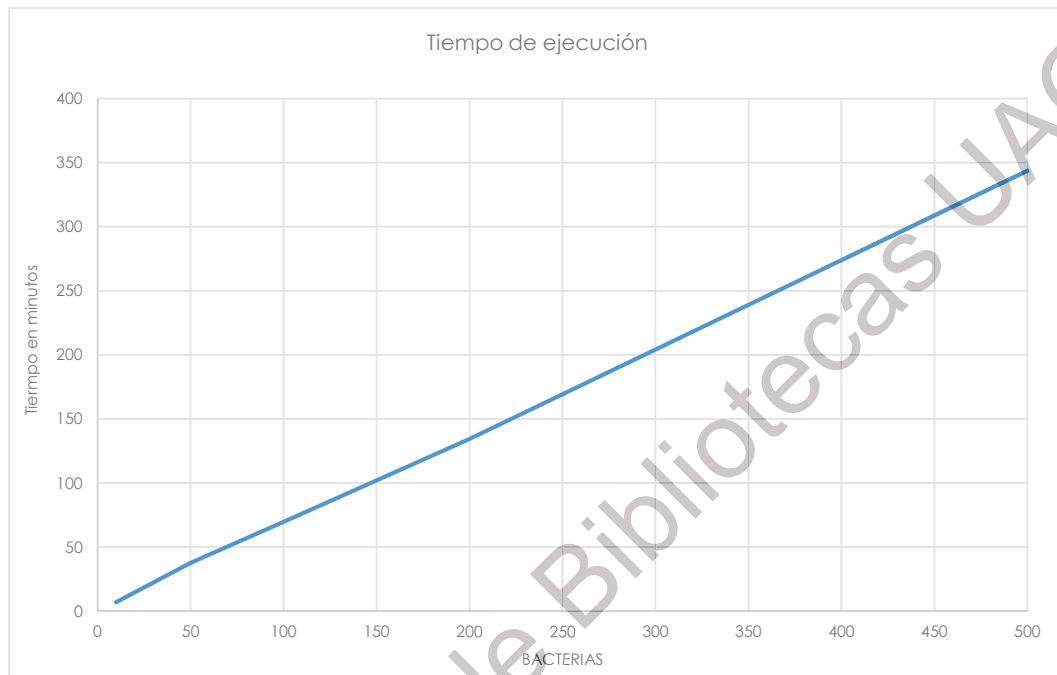
**Tabla 5.4.** Relación de cantidad de bacterias, RMSE y tiempo de ejecución



**Figura 5.5.** Error RMSE de acuerdo a la población S

De igual modo, se puede observar en la figura 5.6, que la relación del tiempo de ejecución con respecto al tamaño de la población tiene un comportamiento lineal, el cual se mantiene en las diferentes variaciones de la población S, dicho comportamiento es importante, ya que nos da un mejor panorama de la relación de las bacterias con el tiempo de ejecución del algoritmo y además puede servir para

calcular una aproximación del tiempo que podría tardar con otros tamaños de la población S.



**Figura 5.6.** Tiempo de ejecución y el número de bacterias (creación propia)

### 5.1.2 Variación de los pasos de quimiotaxis

Ahora bien, cuando aumentamos los pasos de quimiotaxis, es decir,  $N_c$  tiene un valor grande, lo cual resulta en un mayor avance de optimización al tener más oportunidades de llegar a un punto óptimo. Es por lo anterior, que en la figura 5.7 podemos observar una disminución en el error cuando  $N_c=10$  y en especial cuando la población S es mayor ( $S=500$ ), pero, esto supone también un mayor tiempo de ejecución, en la figura 5.8, si se compara la diferencia en el RMSE entre  $N_c=6$  y  $N_c=10$  para el mismo número de bacterias, se puede observar en la tabla 5.5, que sí existe influencia al aumentar el número de  $N_c$  pero no es de gran impacto su influencia en el resultado final. En cambio, si elegimos que el tamaño de  $N_c$  muy pequeño, el algoritmo va a depender de más de la suerte, ya que la población se genera de manera aleatoria, así como también dependerá de la reproducción de las

mejores bacterias, incluso en algunos casos, podría quedar atorado en un mínimo local (convergencia prematura).

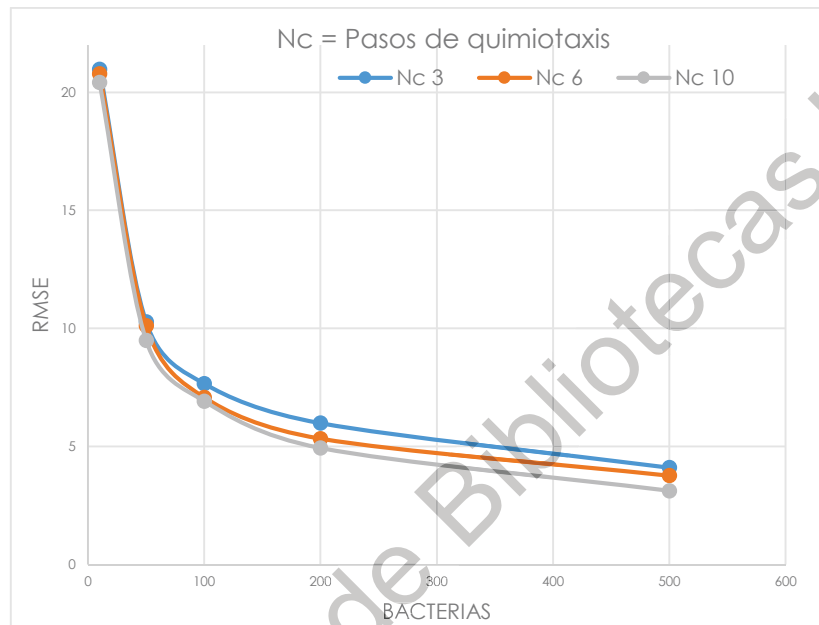


Figura 5.7. Diferentes valores para Nc (pasos de quimiotaxis)

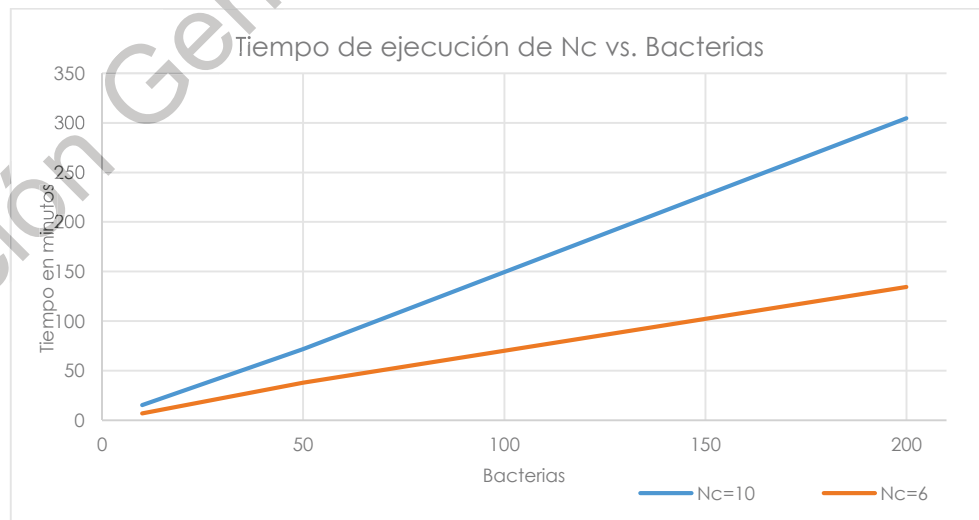


Figura 5.8. Tiempo de ejecución para diferentes valores para Nc

Bacterias	Nc=3	Nc=6	Nc=10
10	20.9758077	20.78511952	20.4111207
50	10.269331	10.10720003	9.48635717
100	7.65544865	7.073980025	6.91341043
200	5.98221459	5.326843964	4.92607912
500	4.09642276	3.757698536	3.11426491

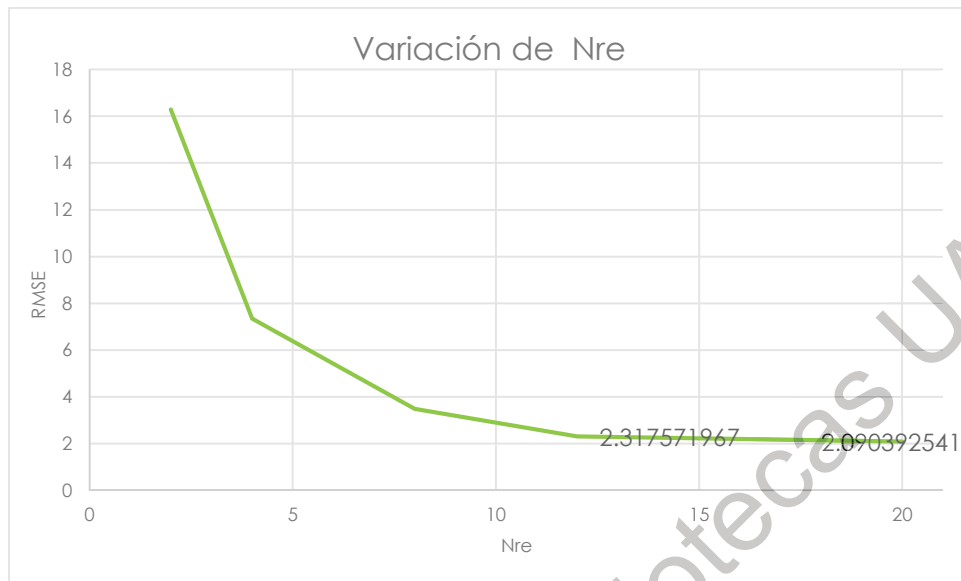
**Tabla 5.5.** Valores de Nc y el error RMSE

### 5.1.3 Variación de los pasos de reproducción

Los pasos de reproducción, Nre, dan un indicativo de cómo el algoritmo ignora las regiones con pocos nutrientes, y se va enfocando en las regiones con alto contenido nutricional para la bacteria, es decir, si está encontrando mejores soluciones, dado que las bacterias en regiones malas mueren y las bacterias que están en buenas regiones suelen reproducirse con mayor velocidad. Sin embargo, Nre depende de los pasos quimiotácticos Nc, sólo es cuando Nc tiene un valor lo suficientemente grande, que el valor de Nre es más significativo a la hora de encontrar mejores soluciones. Cabe agregar que, si Nre es muy pequeño, el algoritmo converge de manera prematura, y si por el contrario el valor de los pasos de reproducción es muy alto, aumenta de manera clara el tiempo de ejecución.

En la figura 5.9, se puede apreciar que el valor de Nre realmente contribuye a la optimización del modelo al aumentar, aunque llega un punto, en este caso es cuando Nre=12, en donde la optimización ya no mejora y el error empieza a estacionarse, aunque Nre aumente a 20 la diferencia ya no es significativa, tabla 5.6.





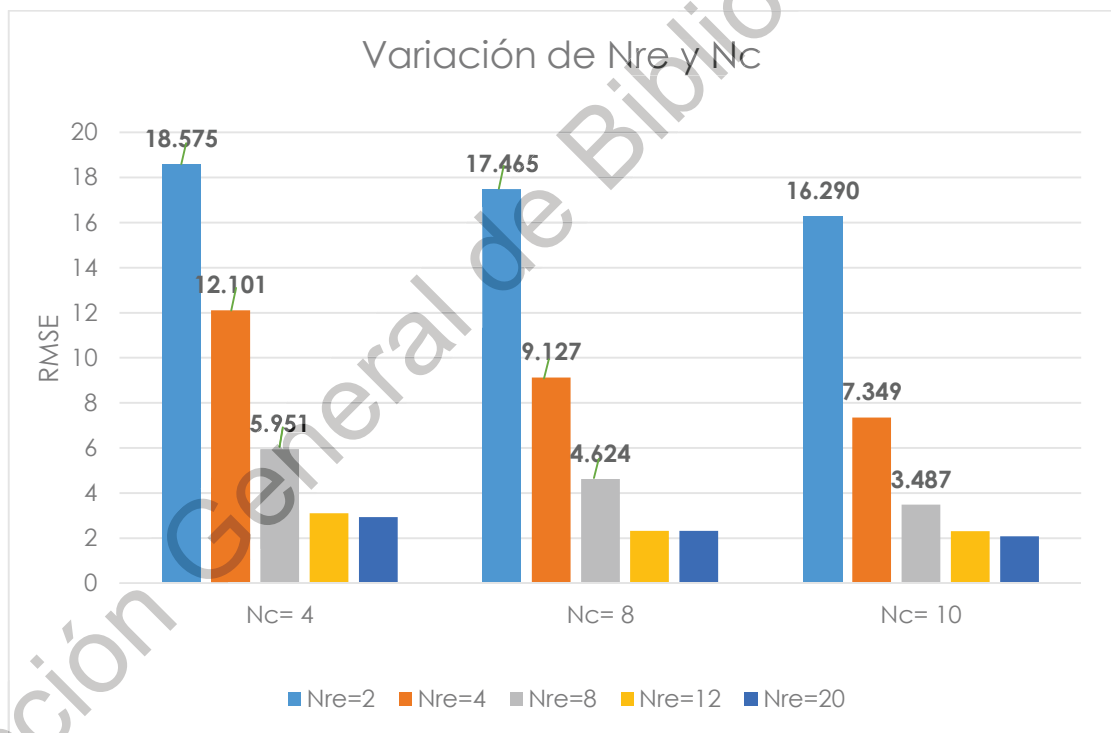
**Figura 5.9.** Variación de Nre con respecto al error RMSE

RMSE	Nre
16.2904319	2
7.34944189	4
3.48697112	8
2.31757197	12
2.09039254	20

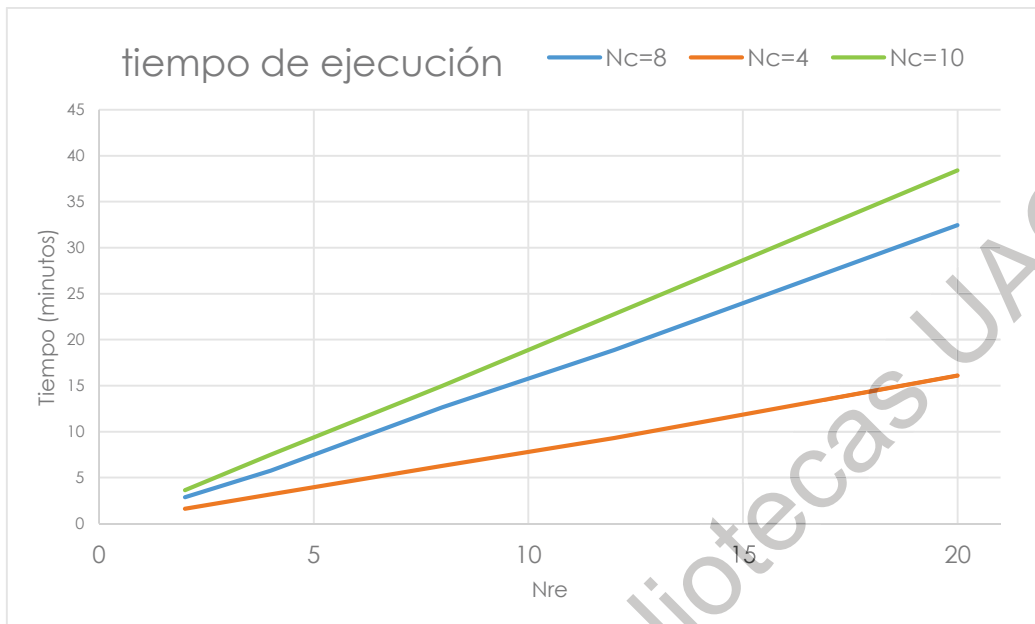
**Tabla 5.6.** Valores de Nre y RMSE

Por otro lado, se mencionaba que Nre depende de que Nc tenga valores grandes para que los cambios de la variable de pasos de reproducción sean significativos, sin embargo en la figura 5.10, se puede ver que cuando Nc= 10 es donde se ve una mejora significativa, y en específico para los valores de Nre={2, 4 , 8}, en los valores de Nre={12 , 20} la diferencia no es tan significativa ya que por la naturaleza del algoritmo empiezan a converger al valor mínimo obtenido con esa configuración para el problema planteado en este trabajo.

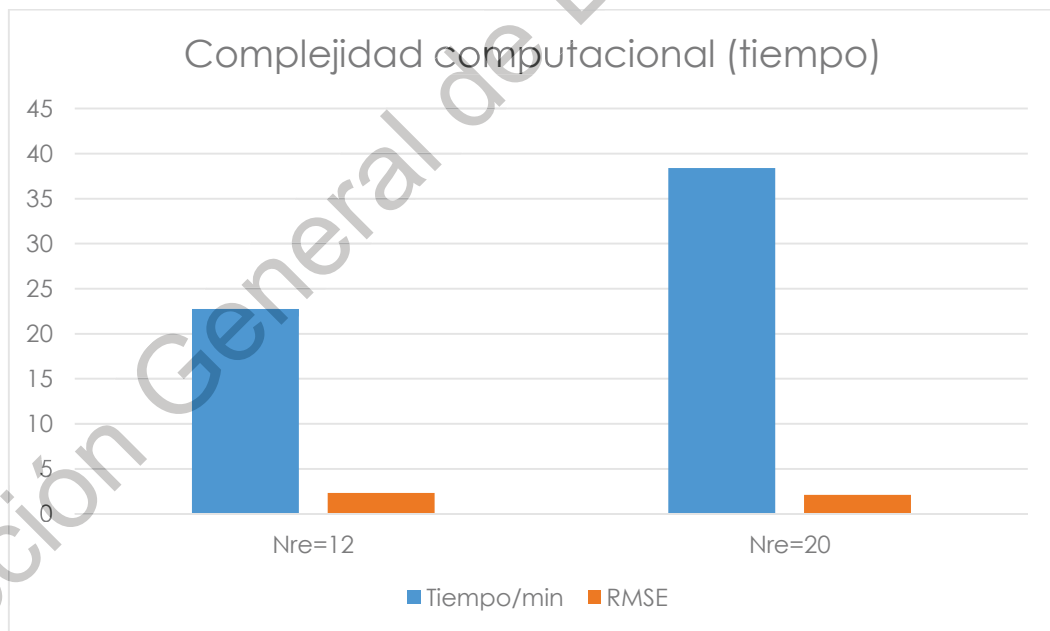
En cuanto al tiempo de ejecución, en la figura 5.11 se muestran los tiempos para los diferentes valores de Nre en combinación con Nc, en esta gráfica se puede observar que indudablemente la duración de la ejecución aumenta de manera significativa al aumentar los valores de ambas variables, reproducción y pasos de quimiotaxis, para los pasos de quimiotaxis se nota el aumento sobre todo en los valores de Nc=8 y Nc=10, y en cuanto a los pasos de reproducción hay una diferencia considerable entre los valores Nr=12 y Nre=20, la cual debe de tomarse en cuenta a la hora de seleccionar los parámetros, ya que la mejora en la optimización no es muy grande en cuanto al valor del error pero sí aumenta mucho el tiempo, figura 5.12.



**Figura 5.10.** Variación de Nre y Nc con respecto a sus respectivos errores RMSE



**Figura 5.11.** Costo de tiempo de variación de Nre y Nc

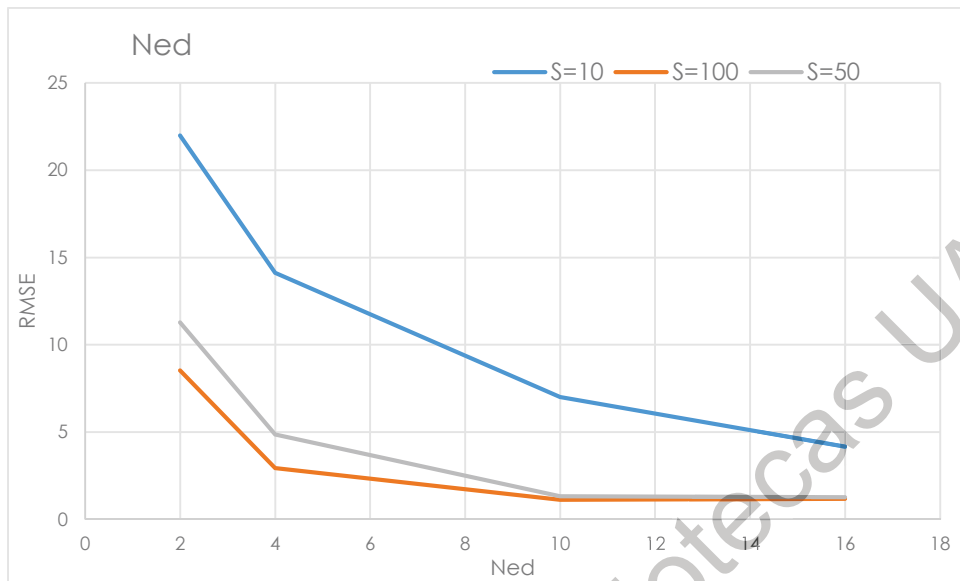


**Figura 5.12.** Costo/beneficio de Nre

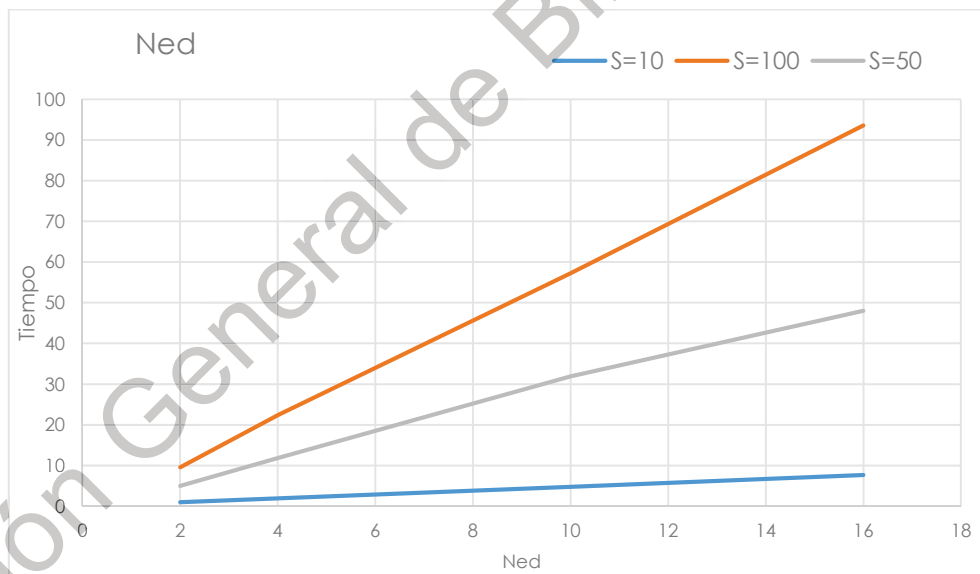
#### 5.1.4 Variación de los eventos de eliminación/dispersión

El valor de los eventos de eliminación/dispersión, Ned, se refiere a cuantas veces se eliminará un grupo de bacterias y se generarán nuevas bacterias en posiciones aleatorias en todo el espacio de búsqueda, por eso es que un valor bajo para Ned no tendrá que basarse en los eventos aleatorios de eliminación/dispersión para encontrar regiones favorables, pero, un valor mayor de Ned permitirá que las bacterias puedan tener acceso a más regiones del espacio de búsqueda en las cuales podrían encontrar mayores concentraciones de nutrientes, aunque hay que tomar en cuenta que al aumentar la cantidad de estos eventos puede aumentar el tiempo de ejecución.

Como resultado de las pruebas que se realizaron con diferentes valores para Ned, se puede apreciar en la figura 5.13, que el aumento de Ned contribuye al resultado de obtener un error menor en el modelo optimizado. Sin embargo, se puede observar en la gráfica, que para cuando el tamaño de población es mayor, la contribución de Ned ya no es tan significativa, ya que el valor del error para cuando Ned=10 está muy cerca del error cuando Ned=16, ya que el algoritmo está convergiendo en el valor límite, tabla 5.7, pero el tiempo de ejecución aumenta de manera considerable, figura 5.14, lo cual debe tomarse en cuenta para seleccionar el valor del parámetro.



**Figura 5.13** Variación de los eventos de eliminación/dispersión



**Figura 5.14** Tiempo de ejecución de Ned

Ned	S=10	S=100
2	21.9864716	8.53933847
4	14.1218685	2.93864778
10	7.00605537	1.12131925
16	4.17984705	1.18270545

**Tabla 5.7.** Valores de Ned y RMSE

### 5.1.5 Variación del tamaño del paso tomado durante el giro

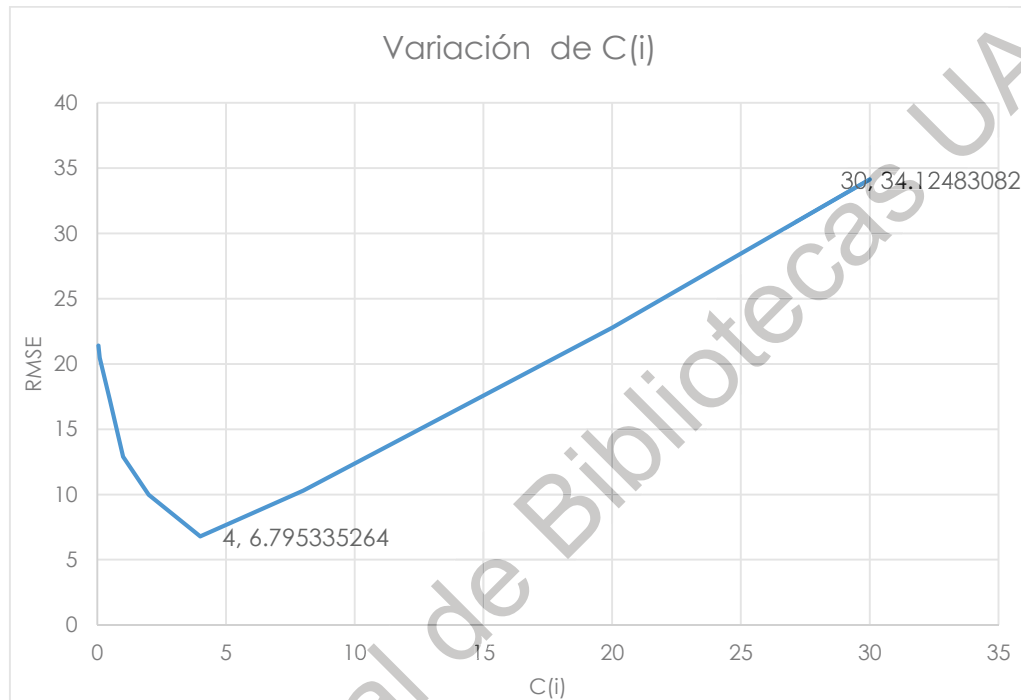
$C(i)$ , donde  $i=1,2,\dots,S$ , se define como el tamaño del paso tomado en una dirección aleatoria especificada por el giro. Se podría decir que  $C(i)$  es el tamaño del paso con el que avanza el algoritmo BFO, esto lo hace uno de los parámetros principales para experimentar utilizando diferentes valores, dado que una pregunta importante que se plantea es ¿Cuáles deben de ser los valores que puede tomar  $C(i)$ ? Una respuesta puede ser que podemos tomar los valores que utiliza una bacteria real, pero podrían ser no aptas para el problema de optimización planteado en este trabajo. Sin embargo, se puede intuir que si los valores del paso  $C(i)$  son muy grandes, y además el valor óptimo se encuentra dentro de un valle con bordes muy pronunciados, la búsqueda podría saltar el valle sin detenerse, o también puede omitir mínimos locales nadando a través de ellos. En cambio, si los valores de  $C(i)$  son muy pequeños, la convergencia se hace más lenta, pero si se encuentra con un mínimo local durante la búsqueda no se desvía demasiado de él.

En la figura 5.15 se puede observar cómo el valor de  $C(i)$  al tomar valores diferentes si afecta en el resultado que obtiene el algoritmo, en esta configuración en específico, alcanza su valor óptimo siendo  $C(i)=4$  con un  $RMSE=6.7$ , tabla 5.8, después de ese valor, el error va aumentando considerablemente, por ejemplo cuando  $C(i)=30$ , el error es incluso mayor que el que se obtiene con el modelo sin optimizar, tabla, como se esperaba, al ser más grande el paso, es más fácil que pase de largo por regiones con altos nutrientes si estas se encuentran dentro de un valle.

$C(i)$	Rmse
0.05	21.4181091
0.1	20.4720087
0.5	17.2198997
1	12.8932434
2	9.96778264
4	6.79533526

8	10.29197
20	22.7732265
30	34.1248308

**Tabla 5.8.** Diferentes valores de C(i)



**Figura 5.15.** Variación del tamaño de paso tomado durante el giro

### 5.1.6 Configuración en conjunto de los parámetros

En las secciones anteriores se explicaron los parámetros más significativos y los efectos que tienen cada uno de ellos en la obtención del modelo optimizado, y basados en dichos resultados, se seleccionó la configuración adecuada de los parámetros que nos ayudarán a obtener el modelo optimizado para las concentraciones del contaminante PM10, evitando que la configuración tenga un tiempo de ejecución elevado.

La configuración final, propuesta para el modelo optimizado con BFO se muestra en la tabla 5.9.

	Valor
S	300
Nc	10
Ned	10
Nre	12
C(i)	4

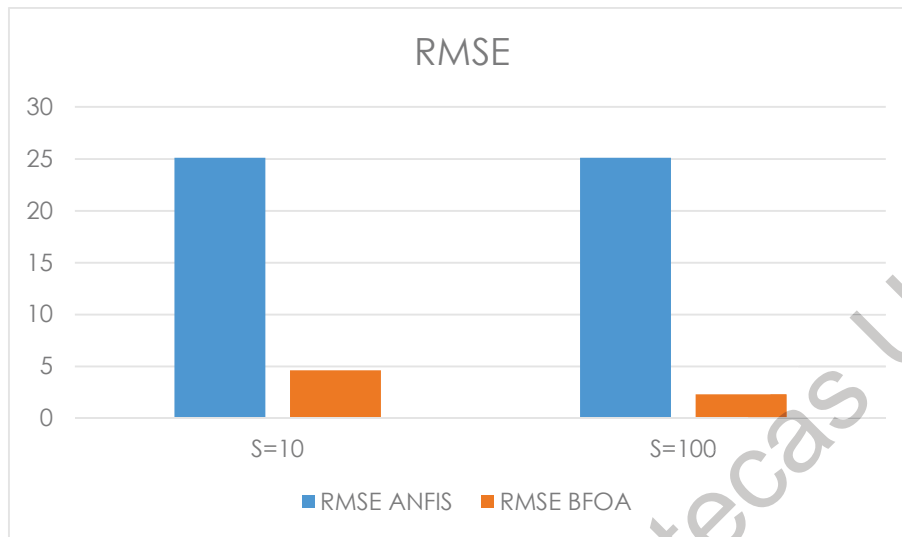
**Tabla 5.9.** Configuración final de parámetros

En la figura 5.16 se muestra una comparativa entre los errores del modelo generado con ANFIS y el modelo optimizado con BFOA para cuando el tamaño de la población S varía, siendo S=10 y S=100, igualmente en la figura 5.17 se muestra el error por cada hora del mes, correspondiente al método ANFIS con respecto a los datos reales (línea azul), en comparación con la tasa de error del modelo optimizado ANFIS/ BFOA respecto a los datos reales (línea naranja), esta comparación demuestra que al ser menor su tasa de error, el modelo optimizado con BFOA, se ajusta mejor a los datos reales que el método ANFIS por sí solo, en la tabla 5.10 se presentan los errores calculados con RMSE, de cada modelo para cada uno de los diferentes valores de la población S.

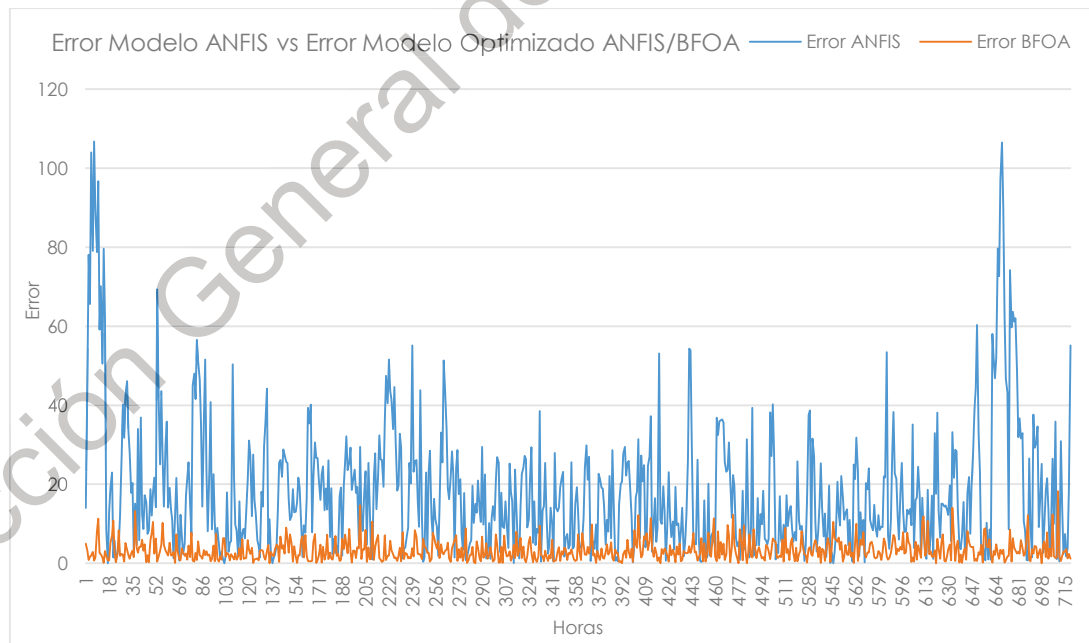
S	RMSE ANFIS	RMSE BFOA
S=10	25.1144672	4.62980105
S=100	25.1144672	2.32101394

**Tabla 5.10.** RMSE para ANFIS y BFOA



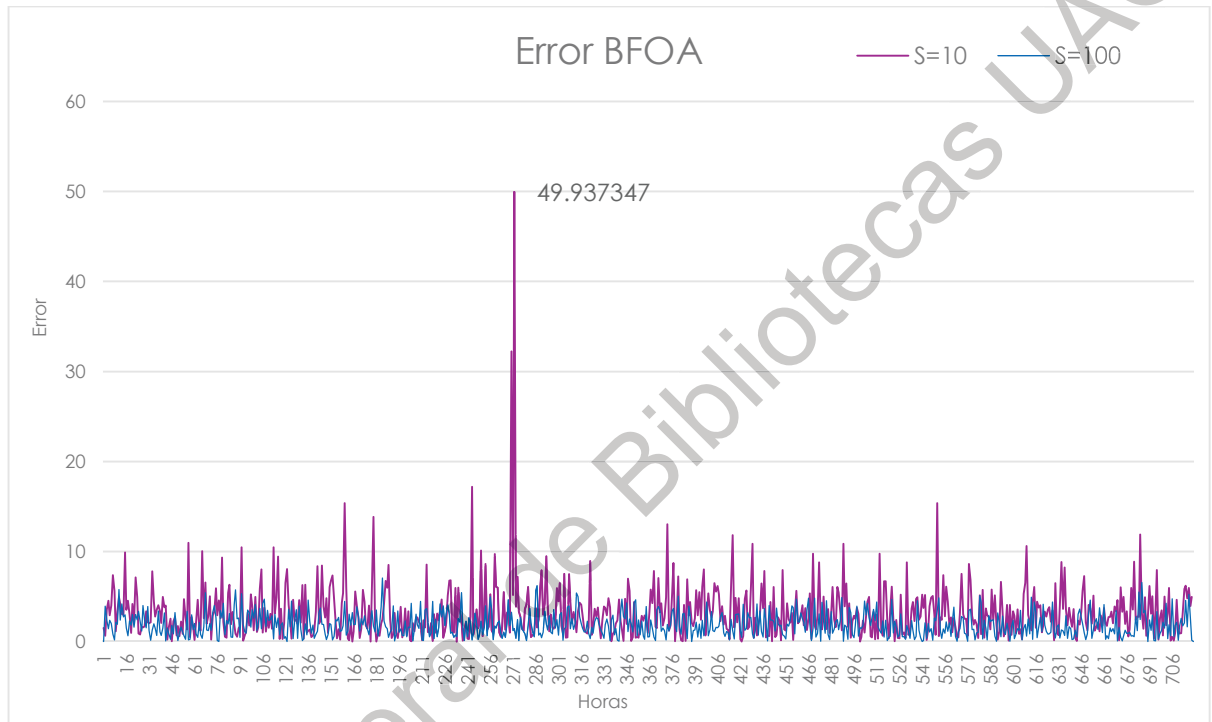


**Figura 5.16.** Errores de ANFIS y Optimizado con BFOA con población diferente



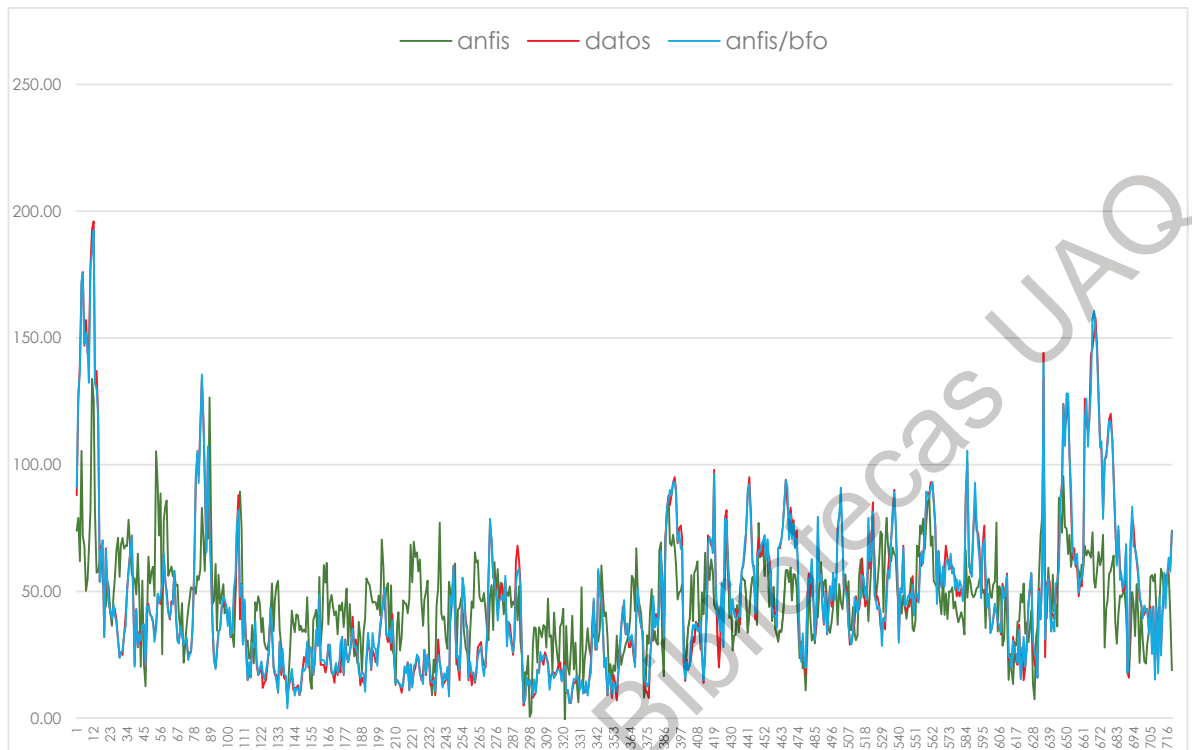
**Figura 5.17.** Comparación de errores del Modelo ANFIS y Modelo optimizado BFOA/ANFIS

En la figura 5.18 se muestra la diferencia de errores entre dos modelos optimizados con BFOA, pero con diferentes tamaños de población  $S$ , en donde se puede apreciar que el valor significativo sigue siendo el tamaño de  $S$ , ya que vemos que incluso se reducen errores de gran magnitud (figura 5.18, etiquetado 49.93).



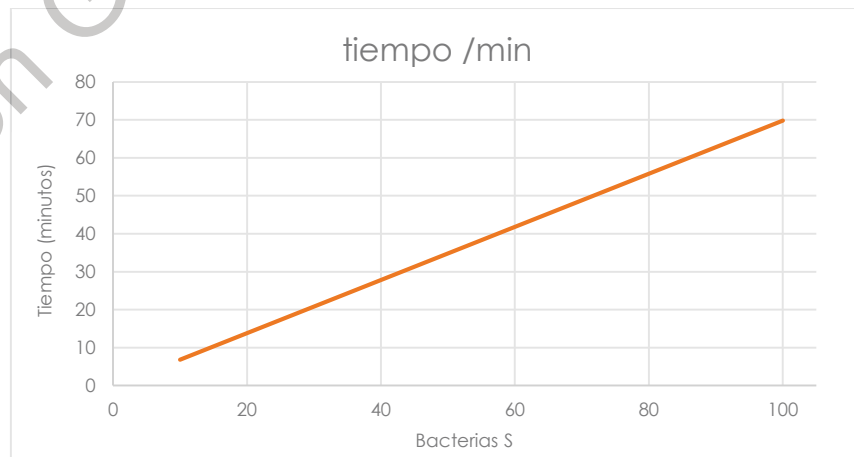
**Figura 5.18.** Comparación de errores de Modelo optimizado BFOA/ANFIS con variación en la población  $S$

De igual modo, en la figura 5.19, se muestra una comparativa entre los datos reales, el modelo generado con ANFIS y el modelo optimizado con la metodología propuesta en el presente trabajo, en esta gráfica se puede observar como el modelo optimizado BFOA/ANFIS se ajusta a los datos reales del comportamiento de la concentración de PM10, de forma que en algunas partes de la gráfica es indistinguible un punto de otro, haciendo visible que la diferencia o error es menor que el existente con ANFIS.



**Figura 5.19.** Comparación Modelo optimizado BFOA/ANFIS vs. Datos reales vs. ANFIS

Aunque cabe mencionar que el tiempo de ejecución en minutos (como se muestra en la figura 5.20 y tabla 5.11), sigue siendo un factor que podría ser un obstáculo para obtener un modelo optimizado si no se cuentan con los recursos de procesamiento necesarios.



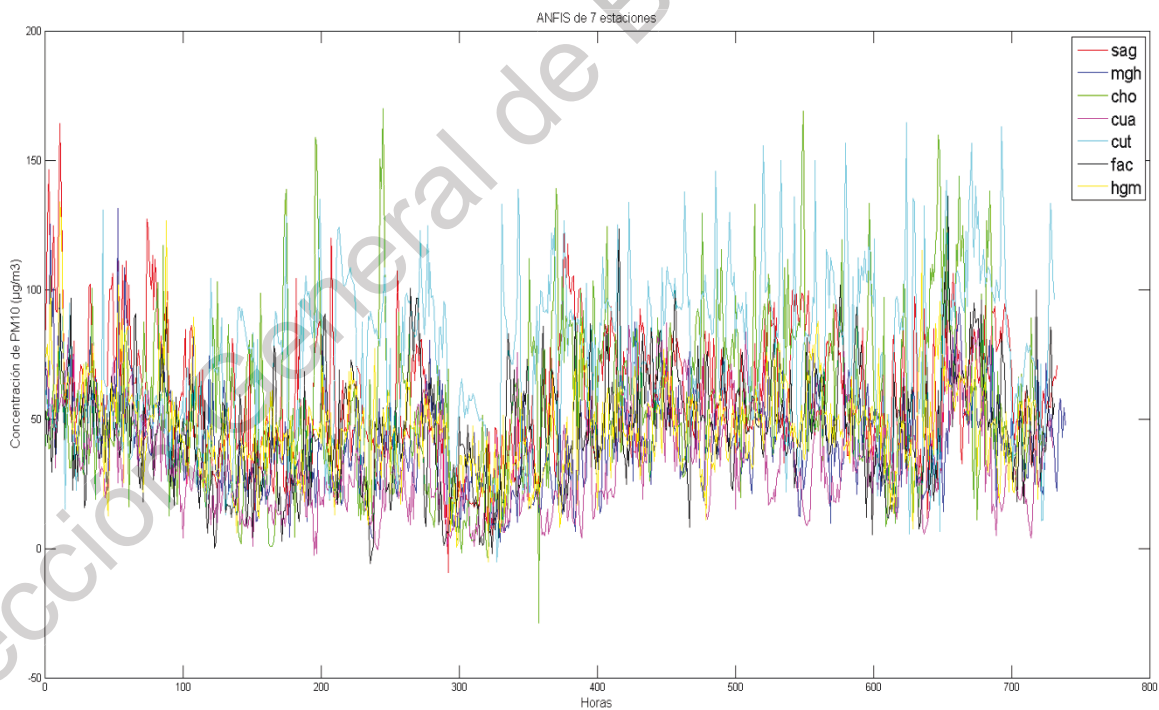
**Figura 5.20.** Tiempo de ejecución del Modelo optimizado BFOA/ANFIS

Tiempo /min	S
6.83141069	10
69.8039942	100

**Tabla 5.11.** Datos de ejecución en minutos del Modelo BFOA/ANFIS

## 5.2 Discusión de resultados

En la sección 5.1 se estableció la variabilidad de los datos de entrenamiento, y por lo tanto podría decirse que es por eso que los modelos, tanto el modelo generado con ANFIS, figura 5.21, y el modelo Optimizado con BFOA presentan una variabilidad inherente al problema.



**Figura 5.21.** Variabilidad de los modelos ANFIS para diferentes estaciones

Una vez establecida la variabilidad de los modelos, se concluye que la utilización del algoritmo de forrajeo bacteriano, BFOA, resultó ser útil para la optimización del modelo de concentración de PM10, es decir, la aportación de este trabajo es el uso exitoso de BFOA aplicado a un problema ambiental, ya que ha sido aplicado a problemas de planeación de tareas (Narendar, 2012), Clasificación de datos (Hossin, 2017), Sistemas de energía (Panda, 2013), entre otros.

De manera específica al realizar las pruebas se concluye que el tamaño de la población  $S$  es uno de los parámetros que influye de manera importante en la optimización ya que, al tener más bacterias o posibles soluciones, se cuenta con una mayor probabilidad de encontrar una solución óptima. Sin embargo, se debe seleccionar un tamaño de  $S$  adecuado debido a que el aumento de  $S$  conlleva un aumento en el tiempo de ejecución.

Después de la realización de varias pruebas se concluye que el número ideal para  $S$  para este problema se encuentra entre los 400 y 500 bacterias, ya que, dentro de este rango, el error empieza a converger en el valor de  $RMSE \approx 3.7$ , y no mejora aunque se aumenten las bacterias, en cambio el tiempo de ejecución sigue aumentando.

Otro parámetro que se probó es el número de pasos de quimiotaxis,  $N_c$ , en donde se observó que un aumento en los pasos de quimiotaxis, sí supone una mejora en la optimización y que no es desdeñable. Sin embargo, es importante tomar en cuenta que el aumento de  $N_c$  también aumenta el tiempo de ejecución de forma considerable, y si se cuentan con recursos limitados, ya sea recursos de tiempo o computacionales, entonces tal vez convenga conservar un valor  $N_c$  no muy alto, la conclusión que se obtiene de las pruebas es que el valor óptimo para  $N_c$  se encuentra entre los valores de  $N=7$  y  $N=10$ , ya que se obtiene una mejoría en la reducción del error sin tener un tiempo de ejecución muy alto.

Los pasos de reproducción Nre, conforman otro parámetro importante para tomar en cuenta al configurar el algoritmo para encontrar una solución óptima, ya que la reproducción es el mecanismo que se encarga de ir descartando las regiones con nutrientes bajos, por esa razón es que en los resultados presentados en este trabajo mostraron que el error (RMSE) comenzaba a disminuir conforme se aumentaban los pasos de reproducción, a partir del valor de Nre=12 el error comenzaba a estacionarse, y las diferencias al aumentar el valor de Nre ya no eran significativas pero el tiempo de ejecución si aumentó lo cual podría indicarnos que aunque se aumente Nre éste tiene un límite donde ya no puede mejorar el error pero sí ocupa los recursos computacionales sin obtener un beneficio apreciable.

Así como también se realizaron pruebas en conjunto con los pasos de quimiotaxis, Nc, ya que se mencionaba en la literatura que el efecto de aumentar Nre era mayor en presencia de un Nc mayor (Passino, 2002); y efectivamente, el incremento de Nre generaba un descenso mayor en el valor del error al existir un valor de Nc mayor. Este descenso es más notable entre Nc=2 y Nc=4, como se muestra en la tabla 5.12, donde en valores mayores a estos el error comienza a convergir y la diferencia ya no es tan notoria, además el tiempo de ejecución aumenta considerablemente entre Nc=4 y Nc=8, pero entre Nc=8 y Nc=10 la diferencia no es tan acentuada, sugiriendo que mantiene una tendencia clara.

Nc	RMSE Nre=2	RMSE Nre=4	RMSE Nre=8	RMSE Nre=12	RMSE Nre=20
Nc= 4	18.5747155	12.1014324	5.95122125	3.11115948	2.94364583
Nc= 8	17.4649532	9.12747131	4.62367454	2.32953639	2.33246408
Nc= 10	16.2904319	7.34944189	3.48697112	2.31757197	2.09039254

**Tabla 5.12.** Relación de Nre y Nc

En lo referente a los eventos de eliminación/dispersión, Ned, que son los que se encargan de eliminar un grupo de bacterias y dispersar nuevas bacterias de manera

aleatoria por todo el espacio de búsqueda lo cual permite acceder a más regiones del espacio de búsqueda que podrían ser mejores. Sin embargo, en las pruebas se observó que Ned tiene mayor influencia en encontrar soluciones óptimas cuando es mayor, Ned=10 y Ned=16 y además la población S es pequeña, S=10, lo cual nos indica que la optimización se está basando en los eventos aleatorios de Ned en lugar de cualquier otro mecanismo del algoritmo BFO.

Otra observación que es pertinente hacer acerca de los resultados de la configuración final, se trata de la presencia de valores de errores muy grandes, llamados outliers o valores atípicos, como se puede observar en la figura 5.17, y aunque son escasos, estos valores afectan el resultado final ya que el error está medido a través de RMSE y esta medida es sensible a los valores atípicos (Chai, 2014), haciendo que el RMSE sea un poco más elevado si no se contara con estos valores. Una medida para evitar la influencia de los outliers en el RMSE podría ser no tomar en cuenta los valores que no se encuentren en cierto rango y esos valores atípicos sean catalogados de otra manera.

Por último, uno de los parámetros que tiene un comportamiento interesante es el tamaño del paso tomado durante el giro,  $C(i)$ , que es el tamaño del paso con el que avanza el algoritmo BFO, para decidir qué valor debe de tomar se experimentó empezando con valores pequeños,  $C(i) < 1$ , ya que el tamaño que debe de tomar depende del problema o espacio de búsqueda. De los resultados obtenidos se puede concluir que este valor debe ser elegido de manera cuidadosa ya que un valor inadecuado nos puede alejar del óptimo. En este caso el tamaño ideal de  $C(i)=4$ , que fue donde se encontraron mejores soluciones, valores menores y mayores nos alejaban de la zona óptima ya que cuando  $C(i)$  es pequeño la convergencia es más lenta, en cambio cuando  $C(i)$  es muy amplio, podría omitir mínimos locales.

## **6. CONCLUSIONES**

En cuanto al algoritmo de BFO de manera general, puede decirse que se reafirma como un algoritmo válido para aplicaciones de optimización, ya que además de

probarse su efectividad al obtener un modelo optimizado para PM10, también se realizaron pruebas para los diferentes parámetros básicos que lo definen y que se comportaron como se esperaba, basados en la teoría.

Se podría decir también que la hipótesis planteada en este trabajo se prueba como verdadera, ya que el algoritmo sí fue capaz de generar un modelo optimizado con respecto a otras soluciones conocidas con lo es ANFIS.

Así mismo, como conclusión de las simulaciones con el algoritmo y sus parámetros, se puede observar que esta implementación de BFOA y el algoritmo en sí tiene sus límites y registrar cuales son para esta aplicación en específico, ya que todos los parámetros de BFOA, al aumentar su valor, convergían o se alejaban de una región óptima, sin dejar de lado el costo en tiempo de ejecución que implicaba este aumento en los parámetros.

Una vez seleccionados los valores ideales para cada parámetro, teniendo como objetivo que se obtengan mejores resultados, pero tomando en cuenta el tiempo de ejecución del algoritmo, se realizaron varias pruebas, en donde los resultados muestran que definitivamente el número de bacterias en la población S es el parámetro que más influye en el algoritmo BFO para encontrar una solución óptima. Por otro lado, si la selección de algunos parámetros en valores muy altos (Nre, Nc, Ned) aumentan de manera considerable el tiempo de ejecución y se cuentan con recursos limitados, es una situación que podría compensarse limitando algunos parámetros que tengan menos influencia, asignándoles un valor medio para que no afecte de manera apreciable en la solución óptima pero que no aumenten el tiempo de ejecución.

Cabe mencionar que la implementación del algoritmo BFO se basó en la versión original del autor (Passino, 2002). Sin embargo, existe un campo muy amplio en donde como futuros trabajos de investigación, se podrían hacer pruebas con más datos, por ejemplo, probar su desempeño con datos de otras ciudades alrededor del mundo e incluso generar un modelo optimizado para partículas PM2.5, también



se podrían incluir más estaciones de monitoreo, así como también la implementación de versiones híbridas para esta misma aplicación. Por ejemplo, implementando la versión híbrida de BFO con Optimización de Enjambre de partículas (PSO, por sus siglas en inglés) como se han hecho en otros trabajos (Panda, 2013) o la versión híbrida (HBFO) en donde se implementa una parte del algoritmo de colonia de Hormigas (ACO, por sus siglas en inglés) en el mecanismo de giro de la bacteria artificial (Narendar, 2012). Así mismo, se podría utilizar el algoritmo BFO pero en su versión multiobjetivo (Panigrahi, 2011), todo esto con el fin de obtener mejores resultados en el modelado de consolidar a BFOA como un algoritmo de optimización y su aplicación exitosa en un problema de datos ambientales.

## REFERENCIAS

Abdulshahed, A. M., Longstaff, A. P., & Fletcher, S. (2015). The application of ANFIS prediction models for thermal error compensation on CNC machine tools. *Applied Soft Computing*, 27, 158-168.

Aceves-Fernandez MA, Sotomayor-Olmedo A, Gorrostieta-Hurtado E, Pedraza-Ortega JC, Ramos-Arreguín JM, Canchola-Magdaleno S, Vargas-Soto E (2011). *Advances in Airborne Pollution Forecasting Using Soft Computing Techniques, Air Quality-Models and Applications*, Prof. Dragana Popovic (Ed.), ISBN: 978-953-307-307-1, intech, DOI: 10.5772/16273.

Aceves-Fernandez MA, Estrada AL, Pedraza-Ortega JC, Gorrostieta-Hurtado E, Tovar-Arriaga S (2015). Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models. *Int. J. Environ. Sci. Toxic. Vol. 3(1)*.

Acharya, D.P., Panda, G., Mishra, S., Lakhshmi, Y.V.S.: *Bacteria Foaging Based Independent Component Analysis*. In: *International Conference on Computational Intelligence and Multimedia Applications*. IEEE Press, Los Alamitos (2007).

Alkasassbeh, M., Sheta, A. F., Faris, H., & Turabieh, H. (2013). Prediction of PM10 and TSP air pollution parameters using artificial neural network autoregressive, external input models: a case study in salt, Jordan. *Middle-East Journal of Scientific Research*, 14(7), 999-1009.

Ali, E. S., & Abd-Elazim, S. M. (2011). Bacteria foraging optimization algorithm based load frequency controller for interconnected power system. *International Journal of Electrical Power & Energy Systems*, 33(3), 633-638.

Alley, E. R., Stevens, L. B., & Cleland, W. L. (1998). *Air quality control handbook*. McGraw-Hill.

Biswas, A., Dasgupta, S., Das, S., Abraham, A.: A Synergy of Differential Evolution and Bacterial Foraging Algorithm for Global Optimization. *Neural Network World* 17(6), 607–626 (2007).

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. SFI Studies in the Science of Complexity, Oxford University Press.

Bougoudis, I., Demertzis, K., Iliadis, L., Anezakis, V. D., & Papaleonidas, A. (2018). FuSSFFra, a fuzzy semi-supervised forecasting framework: the case of the air pollution in Athens. *Neural Computing and Applications*, 29(7), 375-388.

Brabazon, A., & McGarraghy, S. (2018). *Foraging-Inspired Optimisation Algorithms*. Berlin: Springer.

Brook, R. D., Rajagopalan, S., Pope, C. A., Brook, J. R., Bhatnagar, A., Diez-Roux, A. V., Kaufman, J. D. (2010). Particulate Matter Air Pollution and Cardiovascular Disease. *Circulation*, 121(21), 2331–2378.

Brunekreef, B., & Holgate, S. T. (2002). Air pollution and health. *The lancet*, 360(9341), 1233-1242.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3), 1247-1250.

Chatterjee, A., Matsuno, F.: Bacteria Foraging Techniques for Solving EKF-Based SLAM Problems. In: *Proc. International Control Conference (Control 2006)*, Glasgow, UK, August 30- September 01 (2006).

Chen Y., Li Y., Wang G., Zheng Y., Xu Q. et. Al. A novel bacterial foraging optimization algorithm for feature selection. (2017). *Expert Systems with Applications* vol: 83 pp: 1-17.

Chen, H., Zhu, Y., & Hu, K. (2010). Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning. *Applied Soft Computing*, 10(2), 539-547.

Cortina–Januchs M., Quintanilla–Dominguez J., Vega–Corona A., Andina D. Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. (2015). *Atmospheric Pollution Research* vol: 6 (4) pp: 626-634.

Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In *Foundations of Computational Intelligence Volume 3* (pp. 23-55). Springer, Berlin, Heidelberg.

Dorigo M (1993). "Optimization, Learning and Natural Algorithms". Ph.D. thesis, Politecnico di Milano, Italia.

EPA (Environmental Protection Agency). (2000). Air Quality Criteria for Carbon Monoxide, EPA 600/p-99/001F. EEUU.

Feng, X., Li, Q., Zhu, Y., Hou, J., Jin, L., & Wang, J. (2015). Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation. *Atmospheric Environment*, 107, 118-128.

Fernando, H. J., Mammarella, M. C., Grandoni, G., Fedele, P., Di Marco, R., Dimitrova, R., & Hyde, P. (2012). Forecasting PM10 in metropolitan areas: Efficacy of neural networks. *Environmental pollution*, 163, 62-67.

Hansen, C. H., Endres, R. G., & Wingreen, N. S. (2008). Chemotaxis in *Escherichia coli*: a molecular model for robust precise adaptation. *PLoS computational biology*, 4(1), e1.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), 679-688.

Hossin, M. & Mohd Suria, F., (2017). A Conceptual Framework of Bacterial Foraging Optimization Algorithm for Data Classification. *Journal of Engineering and Applied Sciences*, 12: 8531-8536.

Huang, C. J., & Kuo, P. H. (2018). A deep cnn- lstm model for particulate matter (PM2. 5) forecasting in smart cities. *Sensors*, 18(7), 2220.

Karaboga, D. (2005). An idea based on honeybee swarm for numerical optimization. In Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.

Karatzas, K. D., & Kaltsatos, S. (2007). Air pollution modelling with the aid of computational intelligence methods in Thessaloniki, Greece. *Simulation Modelling Practice and Theory*, 15(10), 1310-1319.

Kelly, F. J., & Fussell, J. C. (2012). Size, source and chemical composition as determinants of toxicity attributable to ambient particulate matter. *Atmospheric environment*, 60, 504-526.

Kennedy, J.; Eberhart, R. (1995). «Particle Swarm Optimization». *Proceedings of IEEE International Conference on Neural Networks Vol. IV: 1942–1948*.

Laden, F., Neas, L. M., Dockery, D. W., & Schwartz, J. (2000). Association of fine particulate matter from different sources with daily mortality in six U.S. cities. *Environmental Health Perspectives*, 108(10), 941–947.

Lin, K. P., Pai, P. F., & Yang, S. L. (2011). Forecasting concentrations of air pollutants by logarithm support vector regression with immune algorithms. *Applied Mathematics and Computation*, 217(12), 5318-5327.

Maciąg, P. S., Kasabov, N., Kryszkiewicz, M., & Bembenik, R. (2019). Air pollution prediction with clustering-based ensemble of evolving spiking neural networks and a case study on London area. *Environmental Modelling & Software*.

Majhi, R., Panda, G., Majhi, B., & Sahoo, G. (2009). Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. *Expert Systems with Applications*, 36(6), 10097-10104.

Manahan, Stanley E. *Environmental Science, Technology and Chemistry*. CRC Press LLC, 2000.

Mishra, D., Goyal, P., & Upadhyay, A. (2015). Artificial intelligence based approach to forecast PM2.5 during haze episodes: A case study of Delhi, India. *Atmospheric Environment*, 102, 239-248.

Mishra, S., Bhende, C.N. (2007). Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation. *IEEE Transactions on Power Delivery* 22(1), 457–465.

Moustris, K. P., Ziomas, I. C., & Paliatsos, A. G. (2010). 3-Day-ahead forecasting of regional pollution index for the pollutants NO<sub>2</sub>, CO, SO<sub>2</sub>, and O<sub>3</sub> using artificial neural networks in Athens, Greece. *Water, Air, & Soil Pollution*, 209(1-4), 29-43.

Munoz, M.A., Lopez, J.A., Caicedo, E.: Bacteria Foraging Optimization for Dynamical Resource Allocation in a Multizone temperature Experimentation Platform. In: *Anal. and Des. of Intel. Sys. using SC Tech.*, ASC, vol. 41, pp. 427–435 (2007).

Narendhar, S., & Amudha, T. (2012). A hybrid bacterial foraging algorithm for solving job shop scheduling problems. *arXiv preprint arXiv:1211.4971*.

Ordóñez-De León, B., Aceves-Fernandez, M. A., Fernandez-Fraga, S. M., Ramos-Arreguín, J. M., & Gorrostieta-Hurtado, E. (2019). An improved particle swarm optimization (PSO): method to enhance modeling of airborne particulate matter (PM<sub>10</sub>). *Evolving Systems*, 1-10.

Panda, S., Mohanty, B., & Hota, P. K. (2013). Hybrid BFOA–PSO algorithm for automatic generation control of linear and nonlinear interconnected power systems. *Applied Soft Computing*, 13(12), 4718-4730.

Panigrahi, B. K., Pandi, V. R., Sharma, R., Das, S., & Das, S. (2011). Multiobjective bacteria foraging algorithm for electrical load dispatch problem. *Energy Conversion and Management*, 52(2), 1334-1342.

Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine, 52–67 (2002).

Red de Meteorología y Radiación Solar, REDMET  
<http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBi%27>, fecha de consulta: 15 de marzo del 2018.

Singh, R., Kainthola, A., & Singh, T. N. (2012). Estimation of elastic constant of rocks using an ANFIS approach. Applied Soft Computing, 12(1), 40-45.

Sistema de Monitoreo Atmosférico, SIMAT,  
<http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnml=%27>, fecha de consulta: 15 de marzo del 2018.

Sotomayor-Olmedo, A., Aceves-Fernández, M. A., Gorrostieta-Hurtado, E., Pedraza-Ortega, C., Ramos-Arreguín, J. M., & Vargas-Soto, J. E. (2013). Forecast urban air pollution in Mexico City by using support vector machines: A kernel performance approach. International Journal of Intelligence Science, 3(03), 126.

Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham<sup>1</sup>, Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications, Foundations of Computational Intelligence Volume 3: Global Optimization. Springer Berlin Heidelberg (2009).

Tellez-Rojo, M. M., Romieu, I., Ruiz-Velasco, S., Lezana, M. A., & Hernandez-Avila, M. M. (2000). Daily respiratory mortality and PM10 pollution in Mexico City: importance of considering place of death. European Respiratory Journal, 16(3), 391-396.



Tripathy, M., Mishra, S.: Bacteria Foraging-Based to Optimize Both Real Power Loss and Voltage Stability Limit. *IEEE Transactions on Power Systems* 22(1), 240–248 (2007).

U.S.EPA (U.S. Environmental Protection Agency)  
<https://www.epa.gov/environmental-topics/air-topics>. Visitada en 2017.

Volk HE, Lurmann F, Penfold B, Hertz-Picciotto I, McConnell R. Traffic-related air pollution, particulate matter, and autism. *JAMA Psychiatry*. Published online November 26, 2012. doi:10.1001/jamapsychiatry.2013.266.

Wang, D., Wei, S., Luo, H., Yue, C., & Grunder, O. (2017). A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine. *Science of The Total Environment*, 580, 719-733.

Yang, X. S., Cui, Z., Xiao, R., Gandomi, A. H., & Karamanoglu, M. (Eds.). (2013). *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes.

Zarandi M Faraji M Karbasian M. Interval type-2 fuzzy expert system for prediction of carbon monoxide concentration in mega-cities.(2012).*Applied Soft Computing*, vol: 12 (1) pp: 291-301.

Zhou, Y., Chang, F. J., Chang, L. C., Kao, I. F., & Wang, Y. S. (2019). Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. *Journal of Cleaner Production*, 209, 134-145.

## APÉNDICE



Sociedad Mexicana de Inteligencia Artificial, A. C.  
www.smia.org.mx

Metepec, Estado de México, 08 de Mayo de 2019

**A Quien Corresponda,  
Presente**

Estimados Autores,

A nombre del comité organizador del XI Congreso Mexicano de Inteligencia Artificial (COMIA 2019, <https://smia.mx/comia2019/>) me es grato informarle que después del proceso de revisión de su artículo con los siguientes datos:

**ID del artículo:** 2

**Título:** "Propuesta de Mejora de Modelado de Partículas Contaminantes PM10 Mediante Algoritmos de Inteligencia Colectiva de Bacteria BFOA"

**Autores:** María Del Carmen Cabrera-Hernandez, Julio Alberto Ramírez Montañez, Marco Antonio Aceves-Fernandez, Jesus Carlos Pedraza-Ortega and Juan Manuel Ramos-Arreguin.

HA SIDO ACEPTADO para presentación oral en el congreso mencionado y para publicación en las memorias del congreso, en un número especial de la revista Research in Computing Science, RCS, de publicación periódica con ISSN, indizada en LatinIndex y DBLP. Por tal razón, extendemos la invitación para que realice la presentación de su artículo durante la realización del evento.

COMIA 2019 se celebrará en las instalaciones de la Universidad Autónoma de Nayarit en Tepic, Nayarit del 4 al 7 de Junio del presente año. Es importante aclarar que la organización del evento no se hace responsable de los gastos de traslado y manutención de los participantes en dicho evento.

Agradeciendo, de antemano, la atención brindada a la presente y poniéndome a su disposición para cualquier aclaración sobre el particular, me despido de ustedes enviándoles un cordial saludo.

Dr. Félix Agustín Castro Espinoza  
Presidente General del COMIA 2019 y  
Presidente de la Sociedad Mexicana de Inteligencia Artificial



EZEQUIEL MONTES 56 FRACC. LOS PILARES METEPEC MEXICO 52159. RFC SMI871231F87

## **Propuesta de mejora de modelado de partículas contaminantes PM10 mediante algoritmos de inteligencia colectiva de bacteria BFOA**

Cabrera-Hernandez M.C.<sup>1</sup>, Aceves-Fernandez M.A.<sup>1,\*</sup>, Pedraza-Ortega J.C.<sup>1</sup>, Ramos-Arreguín J.M.<sup>1</sup>

<sup>1</sup> Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas S/N, 76000, Querétaro, Mexico

\*Autor por correspondencia: marco.aceves@uaq.mx

### **Resumen.**

La necesidad de monitorear la calidad del aire se ha vuelto fundamental para la humanidad ya que una calidad deficiente representa una amenaza grave en materia de salud, en especial para cierto tipo de partícula cuyo comportamiento no es lineal como lo es la partícula PM10. En esta contribución el objetivo es optimizar el modelo generado por ANFIS para partículas contaminantes utilizando algoritmos de inteligencia de enjambre. Específicamente, se utiliza BFOA para la optimización, demostrando que es posible disminuir el error del modelo ANFIS, para comprobar la efectividad del modelo propuesto, se llevaron a cabo varios experimentos con los datos y los parámetros propios del algoritmo BFOA.

**Palabras clave:** Contaminación del aire, algoritmo de optimización de forrajeo por bacterias (BFOA), Inteligencia de enjambre, ANFIS.

### **Proposal to improve modeling of PM10 pollutant particles through collective intelligence algorithms of BFOA bacteria**

#### **Abstract.**

The need to monitor air quality has become fundamental for humanity, since poor quality represents a serious threat in terms of health, especially for a certain type of particle whose behavior is not linear, such as the PM10 particle. In this contribution, the objective is to optimize the model generated by ANFIS for pollutant particles using swarm intelligence algorithms. Specifically, BFOA is used for optimization, demonstrating that it is possible to reduce the error of the ANFIS model, to verify the effectiveness of the proposed model; several experiments were carried out with the data and parameters of the BFO algorithm.

**Keywords:** Air pollution, Bacterial foraging optimization algorithm (BFOA), Swarm Intelligence, ANFIS.

### **Introducción**

#### **Material particulado**

El material particulado (MP) o partículas atmosféricas son piezas pequeñas de materia suspendidas en la atmosfera que bien pueden encontrarse en forma de partículas sólidas o

gotas. El origen de esta materia bien puede ser debido a procesos naturales (polen, erosión de la tierra), o actividades humanas (construcciones, actividad industrial, etc.) [1]. Así mismo el MP es liberado directamente de la fuente, o puede ser producto de reacciones químicas que ocurren directamente en la atmósfera.

Las partículas atmosféricas se pueden clasificar según su tamaño, hay partículas con un diámetro de 10  $\mu\text{m}$  (micras de metro) llamadas PM10 y otras más finas que son de 2,5  $\mu\text{m}$  (PM2.5), lo que las hace adecuadas para ser inhaladas fácilmente por los seres humanos y causar graves problemas de salud en las vías respiratorias. y sistemas cardiovasculares [2]. Es importante destacar que, aunque se monitorean gases como el ozono (O<sub>3</sub>), dióxido de nitrógeno (NO<sub>2</sub>), dióxido de azufre (SO<sub>2</sub>) y PM10 como parte de los contaminantes atmosféricos, PM10 ha demostrado en ser uno de los contaminantes que causan mayores problemas de salud, incluso siendo un factor de mortalidad en un sector de la población [3], de ahí la importancia del desarrollo de herramientas más precisas para modelar su concentración en el medio ambiente.

- **Contaminantes atmosféricos y PM10**

- Para el del modelado comportamiento de las partículas PM10 ya se han utilizado varias técnicas como redes neuronales para predecir el comportamiento del contaminante con 24 horas de anticipación [4], en general las redes neuronales han sido de los métodos más utilizados para el modelado de contaminantes atmosféricos [5][6], incluyendo algunas variantes de dicho método como las redes neuronales autorregresivas (ARNN, autoregressive neural network) [7]. Algunos otros métodos utilizados para el modelado de contaminantes ambientales incluyen, lógica difusa de intervalo tipo 2 [8]; así como el uso de cuantificación de recurrencia para investigar los cambios en el comportamiento no lineal de los contaminantes atmosféricos [9] Uno de los métodos innovadores en esta área es el uso de algoritmos de inteligencia de enjambre, por ejemplo, el algoritmo de Colonia de Hormigas [10] que mostró resultados alentadores para los algoritmos basados en la naturaleza aplicados a este problema.

### **Inteligencia de enjambre**

Los algoritmos de inteligencia de enjambre existen desde finales de los años 80 del siglo pasado, Beni y Wang [11] introdujeron el término. La inteligencia de enjambre está basada en un conjunto de agentes que interactúan entre sí, así como con su ambiente, con el objetivo de alcanzar un comportamiento integral y cooperativo que da como resultado una interacción con cierto grado de complejidad que de otra manera no podría ser completado por los agentes de manera individual, de esa manera encontrar mejores soluciones como una comunidad. Estos algoritmos están inspirados principalmente en fenómenos naturales como las colonias de Hormigas [12], de abejas [13] o de bacterias, gotas de agua [14], el comportamiento de los murciélagos [15], termitas [16], etc.

- **Algoritmo de optimización de forrajeo bacteriano**

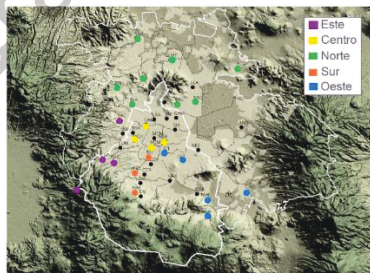
El algoritmo de optimización de forrajeo bacteriano (BFOA, Bacterial Foraging Optimization Algorithm) [17] está inspirado en el proceso de forrajeo de la bacteria E. Coli. La forma en que esta bacteria realiza los procesos de forrajeo y reproducción, maximiza la energía que obtiene del ambiente, haciendo de este un método de optimización.

BFOA ha sido aceptado como un algoritmo de optimización global, su eficiencia se ha demostrado en varias áreas. Por ejemplo, su aplicación en el campo de la ingeniería eléctrica y el control [18], reconocimiento de patrones [19], diseño PID [20] programación de horarios de trabajo utilizando sólo BFOA [21], así como el desarrollo de aplicaciones para resolver ecuaciones simultáneas [22], incluso se ha utilizado para la selección de características en problemas de reducción de dimensionalidad de conjuntos de datos que así lo requieran [23]. El algoritmo de BFO se ha utilizado también de manera híbrida junto con otros métodos por ejemplo junto con el algoritmo de Colonia de Hormigas para el mismo problema de programación de horarios [24] y se ha utilizado en conjunto con el Algoritmo de Optimización de Enjambre de Partículas PSO (Particle Swarm Optimization) para diseño óptimo de estabilizadores del sistema de potencia [25] y en problemas de generación de energía [26].

## Metodología y materiales

### Materiales

Los datos usados para construir el modelo provienen del Sistema de Monitoreo Atmosférico (SIMAT) que es responsable de la medición permanente de los niveles de contaminantes en la atmósfera en la Ciudad de México y el área Metropolitana [27]. El SIMAT a su vez tiene un subsistema, la Red Automática para el Monitoreo Atmosférico (RAMA), este sistema usa equipo de medición continua de contaminantes del aire, tales como dióxido de azufre, monóxido de carbono, dióxido de nitrógeno, ozono, PM10 y PM2.5. La ubicación de las estaciones de monitoreo están divididas en 5 áreas, Este, Centro, Norte, Sur y Oeste, la organización de las estaciones se puede observar en la Fig. 1.



**Fig. 1.** Mapa de las estaciones de RAMA

Las estaciones de monitoreo recolectan información de las concentraciones de los contaminantes cada hora, estas concentraciones son medidas en  $\mu\text{g}/\text{m}^3$ .

### Construcción del modelo

- **Sistema adaptativo de inferencia neurodifusa (ANFIS).**

El enfoque propuesto en esta contribución es optimizar un modelo existente (modelo base), utilizando BFOA como un optimizador. El modelo base propuesto es un sistema adaptativo

de inferencia neurodifusa, ANFIS (Adaptive neuro fuzzy inference system) que es un tipo de red neuronal artificial que incluye un sistema de inferencia difusa Takagi-Sugeno (Fig. 2).

ANFIS construye un Sistema de inferencia difusa (FIS) dado un conjunto de datos de tipo entradas/salida, los parámetros de las funciones de membresía se ajustan utilizando un algoritmo de retropropagación o en combinación con un método de mínimos cuadrados.

Un FIS se puede definir como un conjunto de reglas difusas de tipo SI-ENTONCES (if-then), que son expresiones de la forma SI A ENTONCES B, donde A y B son etiquetas de conjuntos difusos [28]

Para simplificar el ejemplo supongamos que tenemos dos entradas  $(x,y)$  y una salida  $f$ , y se utilizan cinco capas para construir el modelo. Cada una de las capas contienen varios nodos que pueden ser adaptativos (nodos cuadrados) o fijos (nodos circulares) [29].

- *Capa 1.* Es la capa difusa y convierte las entradas del modelo en conjuntos difusos por medio de funciones de membresía (MF) y las funciones del nodo se describen así:

$$O_{1,i} = \mu_{A_i}(X_1) \text{ para } i = 1,2 \quad (1)$$

$$O_{1,i} = \mu_{B_{1-2}}(Y_1) \text{ para } i = 3,4 \quad (2)$$

Donde  $X_1$  y  $Y_1$  son los nodos de entrada, A y B corresponden a las etiquetas lingüísticas asociadas a estos nodos,  $\mu(X_1)$  y  $\mu(Y_1)$  son las funciones de membresía (MF), los parámetros en esta capa se llaman parámetros de premisa.

- *Capa 2.* Los nodos en esta capa son fijos, la función de cada nodo se multiplica por las señales de entrada, lo cual sirve como señal de salida y se etiquetan con  $\Pi$ .

$$O_{2,i} = w_i = \mu_{A_i}(X_1) \cdot \mu_{B_{1-2}}(Y_1) \text{ para } i = 1,2 \quad (3)$$

Donde  $O_{2,i}$  es la salida de esta capa y la señal de salida  $w_i$  representa la fuerza de disparo de la regla.

- *Capa 3.* Los nodos de esta capa también son fijos y están etiquetados con N, y su función es normalizar la fuerza del disparo calculando la proporción de la fuerza de disparo del nodo  $i$ -ésimo para sumar la fuerza de disparo de todas las reglas.

$$O_{3,i} = \bar{w} = \frac{w_i}{w_1 + w_2} \text{ para } i = 1,2 \quad (4)$$

Donde  $O_{3,i}$  es la salida de la capa 3 y  $\bar{w}$  es la fuerza de disparo pero ya normalizada.

- *Capa 4.* Los nodos de esta capa se ajustan y se definen como

$$O_{4,i} = \bar{w}_i \cdot f_i \text{ para } i = 1,2 \quad (5)$$

Donde  $f_1$  y  $f_2$  son las reglas difusas SI-ENTONCES definidas de la siguiente manera:

Regla 1. SI  $X_1$  es  $A_1$  y  $Y_1$  es  $B_1$ , ENTONCES  $f_1 = p_1 X_1 + q_1 Y_1 + r_1$

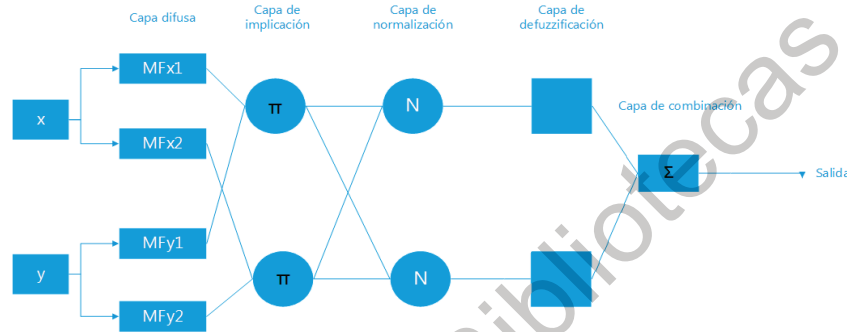
Regla 2. SI  $X_1$  es  $A_2$  y  $Y_1$  es  $B_2$ , ENTONCES  $f_2 = p_2 X_1 + q_2 Y_1 + r_2$

Donde  $p_i, q_i$  y  $r_i$  son parámetros establecidos y referidos como parámetros consecuentes.

- **Capa 5.** En esta capa los nodos son fijos etiquetados con  $\Sigma$  y su función es calcular la salida total y que se define por:

$$O_{5,i} = \sum_i \bar{w}_i \cdot f_i = \frac{\sum_i w_i f_i}{w_i} = f_{out} = \text{salida total} \quad (6)$$

ANFIS tiene una regla de aprendizaje muy sencilla, que es la "propagación hacia atrás", esta regla calcula las señales de error de forma recurrente, comenzando desde la capa de salida (Capa 5) hacia los nodos de entrada (Capa 1).

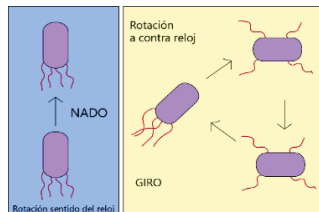


**Fig. 2.** Estructura ANFIS (adaptado de [25])

### Modelo de Optimización

El algoritmo de BFO imita el proceso de forrajeo de una bacteria real cuya locomoción es lograda a través del movimiento de sus flagelos que ayudan a la bacteria a nadar o girar, estas operaciones son básicas en el proceso de forrajeo. Si los flagelos giran en el sentido de las manecillas del reloj genera un movimiento de giro, en un ambiente nocivo la bacteria va a dar más giros para encontrar nutrientes y cuando los flagelos giran en el sentido contrario a las manecillas del reloj la bacteria realiza un nado, en un ambiente adecuado para la bacteria el movimiento de nado recorre mayores distancias [30]. Los movimientos mencionados anteriormente forman parte del proceso de quimiotaxis en donde la bacteria buscará moverse en un ambiente con nutrientes y evitará las zonas dañinas. En la figura se puede observar los movimientos de giro y nado.

Cuando la bacteria se encuentre con nutrientes suficientes y la temperatura del ambiente lo permita la bacteria se reproducirá dividiéndose en dos y creando una réplica de sí misma creando así una colonia de bacterias. Igualmente, si ocurre un ataque o el ambiente cambia repentinamente un grupo de bacterias es dispersado hacia otras áreas del ambiente o son eliminadas, este evento se llama eliminación-dispersión.



**Fig. 3.** Quimiotaxis de una bacteria (traducido de [17])

Si suponemos que buscamos encontrar el mínimo de  $J(\theta)$  donde  $\theta \in \mathbb{R}^p$  (en donde  $\theta$  es un vector  $p$ -dimensional) y se ignora la naturaleza del gradiente  $\nabla J(\theta)$  ya que no contamos con una descripción analítica o incluso mediciones de  $\nabla J(\theta)$ . BFOA implementa una imitación de los mecanismos principales presentes en una colonia de bacterias *E. Coli* real: quimiotaxis, formación de la colonia o enjambre, reproducción y eliminación-dispersión con el cual se puede resolver el problema de optimización sin gradiente. La forma de explicar en qué consiste una bacteria virtual es que es realidad una solución de prueba que se mueve en la superficie funcional para ubicar el óptimo global [23].

Para poder implementar BFOA es necesario definir el paso quimiotáctico como un giro seguido de un nado o un giro seguido de un giro. Entonces  $j$  es el índice del paso quimiotáctico,  $k$  el índice del paso de reproducción, y por último  $l$  es el índice del evento de eliminación-dispersión.

El algoritmo cuenta con ciertos parámetros que deben inicializarse y de los cuales depende el desempeño del algoritmo dependiendo del problema. Sean:

$p$ : Dimensión del espacio de búsqueda

$S$ : Bacterias que forman la población

$N_c$ : Pasos quimiotácticos

$N_s$ : Longitud de nado

$N_r$ : pasos de reproducción

$N_e$ : Eventos de eliminación dispersión

$P_{ed}$ : Probabilidad de que una bacteria sea eliminada o dispersada

$C(i)$ : Tamaño del paso tomado en una dirección aleatoria especificada por el giro.

Sea entonces  $\theta^i(j, k, l) = \theta^i(j, k, l)$  donde  $i = 1, 2, \dots, S$  la posición de cada miembro en la población de  $S$  bacterias en el paso quimiotáctico  $j$ -ésimo, el pas  $k$ -ésimo de reproducción y el evento  $l$ -ésimo de eliminación-dispersión, entonces podemos asociar un costo  $J(j, k, l)$  a esa posición  $\theta^i(j, k, l)$ .

A continuación, se describe cada una de las etapas de BFOA

- *Quimiotaxis*. Supongamos que  $\theta^i(j, k, l)$  donde  $i = 1, 2, \dots, S$  la posición de cada miembro en la población de  $S$  bacterias en el paso quimiotáctico  $j$ -ésimo, el paso  $k$ -ésimo de reproducción y el evento  $l$ -ésimo de eliminación-dispersión y  $C(i)$  es el paso tomado en una dirección aleatoria especificada por el giro, entonces el movimiento de quimiotaxis artificial se representa con:

$$\theta^i(j, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (7)$$

Donde  $\Delta$  es un vector que contiene una dirección aleatoria cuyos elementos se encuentran entre  $[-1, 1]$ .

- *Enjambre*. Las celdas reales responden a estímulos químicos para formar grupos de células y así viajar en el ambiente. Las señales de célula a célula se representan de la siguiente manera:



$$J_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S [-d_{attractant} \exp(-w_{attractant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] + \sum_{i=1}^S [-h_{repellant} \exp(-w_{repellant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \quad (8)$$

Donde  $J_{cc}$  es el valor que se agregará a la función objetivo que será minimizada  $w_{attractant}$ ,  $d_{attractant}$ ,  $h_{repellant}$ ,  $w_{repellant}$  son coeficientes que deben escogerse de manera adecuada.

- *Reproducción.* Las bacterias menos saludables deben morir mientras las bacterias más saludables que son las que tienen un valor más bajo en la función objetivo se van a reproducir dividiéndose en dos, manteniendo el tamaño de la población constante.
- *Eliminación/dispersión.* Para simular los eventos de dispersión y eliminación, se eliminan un grupo de bacterias de manera aleatoria con una pequeña probabilidad, y se inicializan los reemplazos de manera aleatoria sobre el espacio de búsqueda.

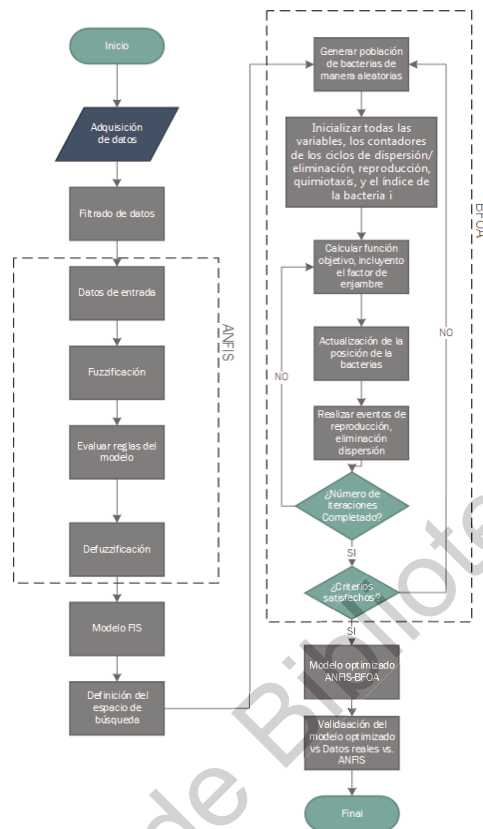
### Metodología Propuesta

A continuación, se presentan los pasos que conforman la metodología para obtener el modelo optimizado (Fig. 4).

Adquisición de datos: Los datos puros para crear el modelo son obtenidos de una base de datos.

Filtrado de datos: La base de datos contiene algunos datos no válidos o que pudieran ser parciales y deben de ser filtrados para no tener un modelo sesgado.

Entrada de datos: una vez que los datos han sido revisados y son válidos, entonces pueden ser alimentados al modelo.



**Fig. 4.** Modelo optimizado

**Fuzzificación:** Es el proceso de convertir en los datos de entrada en un valor lingüístico, esto depende de las funciones de membresía.

**Evaluación de las reglas:** las reglas del modelo son evaluadas con respecto a las reglas difusas y los valores de las funciones de membresía.

**Defuzzificación:** El método de defuzzificación utilizado fue el de promedio ponderado de todas las salidas de reglas.

**Modelo Sistema de inferencia Difuso:** Una vez que se completaron los pasos de evaluación y defuzzificación, el modelo se construye con las ecuaciones propias de ANFIS expresadas en la sección de construcción del modelo.

**Definición del espacio de búsqueda:** Un espacio de búsqueda se define como todas las soluciones factibles dentro de nuestro problema, es por esto que nuestro espacio de búsqueda debe de encontrarse primero, entre los datos válidos para los factores ambientales Humedad relativa, temperatura, dirección y velocidad del viento así como también dentro de los valores factibles de PM10, ya que dada la naturaleza del problema, otra forma de definir el espacio de búsqueda es muy difícil ya que las posibilidades son muy amplias

**Generación de la población:** se deben de generar la población con S bacterias, en posiciones iniciales aleatorias dentro de un rango de posibles valores que puedan tener los datos reales.

Inicialización de parámetros: Los parámetros de BFOA se deben inicializar, estos parámetros incluyen inicializar los contadores de los ciclos de eliminación/dispersión  $j$ , reproducción  $k$ , quimiotaxis  $l$ , y el índice  $s$  de la bacteria  $i$ , así como los parámetros de atracción y repelente ( $w_{attractant}$ ,  $d_{attractant}$ ,  $h_{repellant}$ ,  $w_{repellant}$ ), que son los que generan el efecto de enjambre, en este caso se utilizaron los valores de  $d_{attractant} = 0.05$ ,  $w_{attractant} = 0.15$ ,  $h_{repellant} = d_{attractant}$ ,  $w_{repellant} = 10$

Estos valores fueron seleccionados en un principio de acuerdo a estudios realizados por algunos autores [17] mismos que recomiendan se seleccionen siguiendo ciertos criterios, por ejemplo la altura o magnitud del repelente debe ser igual al ancho de la señal de atracción para evitar tener dos bacterias en el mismo punto y esto se modela haciendo que  $h_{repellant} = d_{attractant}$ , por otro lado, también recomienda que la señal de atracción sea muy pequeña en comparación a los valores de concentración de nutrientes en nuestro espacio de búsqueda y por eso el repelente debe ser lo suficientemente grande para evitar que las bacterias estén muy cerca, sin embargo, el experimentar con diferentes variantes de estos valores es parte de trabajos futuros para localizar el valor óptimo de este parámetro.

Cálculo de la función objetivo: Se debe de calcular el criterio que nos dirá que tan saludable está nuestra bacteria, en este algoritmo se agrega el cálculo del factor de enjambre  $J_{cc}$  con la ecuación (8) y agregarse al costo  $J$  de la bacteria en la posición actual. Actualizar posición de las bacterias: Se hace una actualización de las bacterias de acuerdo al costo de la bacteria en la posición actual y comparando la posición siguiente, para lograr esto se calcula el giro de la bacteria y se realiza un nado en esa dirección, se calcula el costo de la nueva posición, si esta tiene menor costo entonces se convierte en la mejor posición de la bacteria y se sigue moviendo en esa dirección, de lo contrario si termina el ciclo de nado y no encontró una mejor posición significa que no está ubicada en un ambiente adecuado y continúa con la siguiente bacteria.

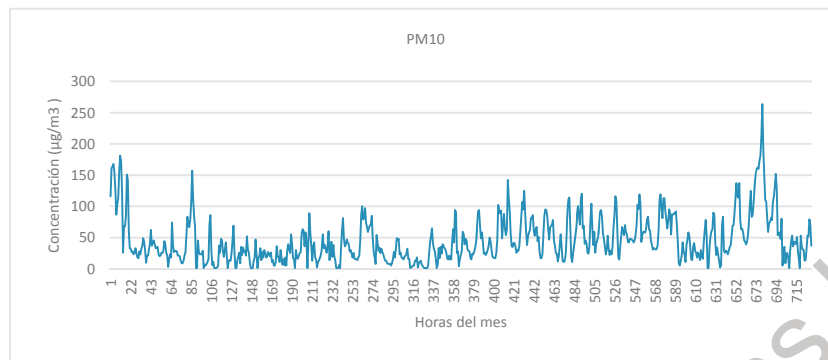
Se realizan los eventos de reproducción, dispersión y eliminación en donde la mitad mejor de la población se reproduce, es decir se hace una copia exacta de las bacterias con menor costo total y la otra mitad es reemplazada por bacterias generadas aleatoriamente, al igual un grupo con probabilidad baja son dispersadas en el espacio de búsqueda de manera aleatoria.

Modelo optimizado: si los ciclos del algoritmo BFO han sido completados y se cumplen los criterios de optimización se dice que se ha creado un Modelo Optimizado.

## Resultados

En la Fig. 5 se pueden observar los datos reales de concentración de PM10 y su comportamiento no lineal.

El objetivo de los experimentos realizados con el método de optimización BFOA es la reducción del error que existe al aplicar el modelo creado con ANFIS. Este error se puede observar en la Fig. 6, la cuantificación del error entre los datos reales y los calculados por el modelo ANFIS se llevó a cabo utilizando la raíz del error cuadrático medio (RMSE por sus siglas en inglés) que es un método históricamente utilizado para medir la precisión de los pronósticos de datos [31].



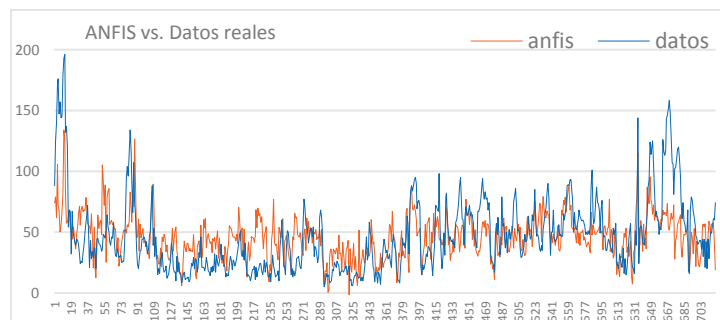
**Fig. 5.** Concentración de PM10 enero 2016 estación FAC

En el caso del modelo generado con ANFIS se obtiene un  $RMSE = 24.147$ , el cual debe de ser minimizado con BFOA.

Por otro lado, si se busca generar un modelo más preciso, podemos realizar más pruebas variando los parámetros del algoritmo de optimización de Forrajeo de Bacterias, como lo son el número de bacterias que se moverán en el espacio de búsqueda, la cantidad de pasos de quimiotaxis y de pasos de reproducción para determinar cuáles son los valores óptimos de tales parámetros.

En la Fig. 7 y la tabla 1 observamos cómo la cantidad de bacterias, afecta en el rendimiento del modelo optimizado. Como puede observarse, al tener más agentes que busquen en el espacio de búsqueda, es más probable que encuentren una solución óptima a nuestro caso de estudio.

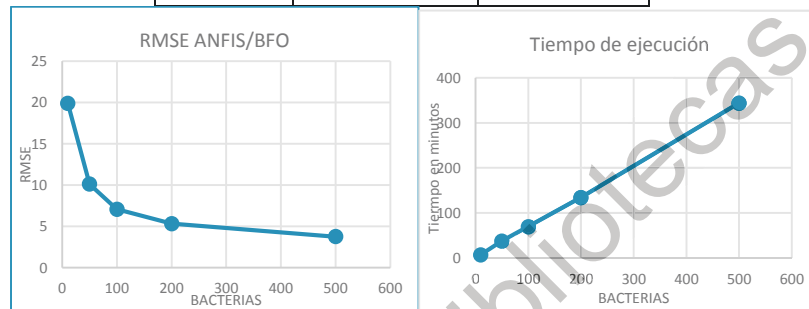
Así mismo, se puede observar que por la naturaleza de BFOA, ya que es un algoritmo que cuenta con ciclos anidados, el tiempo de ejecución aumenta de manera considerable, y aunque el error se ve reducido al aumentar el tamaño de la población, esta diferencia en la optimización no parece importante, tomando en cuenta los recursos computacionales que requiere su cálculo. Como contexto para las pruebas de tiempo de ejecución, mencionamos que se utilizó una PC con Windows 7, de 64 bits, Procesador Intel Core i3-2100 3.10 GHz y memoria RAM de 12Gb.



**Fig. 6.** Comparación datos reales vs ANFIS

**Tabla 1.** Relación de cantidad de bacterias, RMSE y tiempo de ejecución

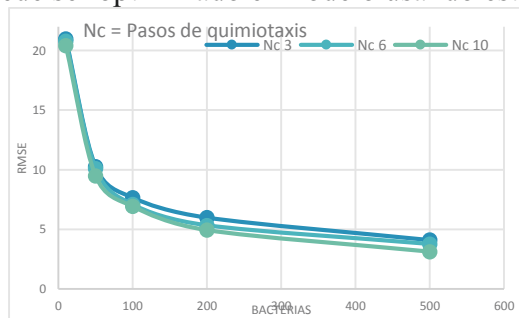
Bacterias	RMSE	tiempo/seg
10	19.8851195	419.84
50	10.1072	2268.786
100	7.07398003	4185.117
200	5.32684396	8062.675
500	3.75769854	20625.8073



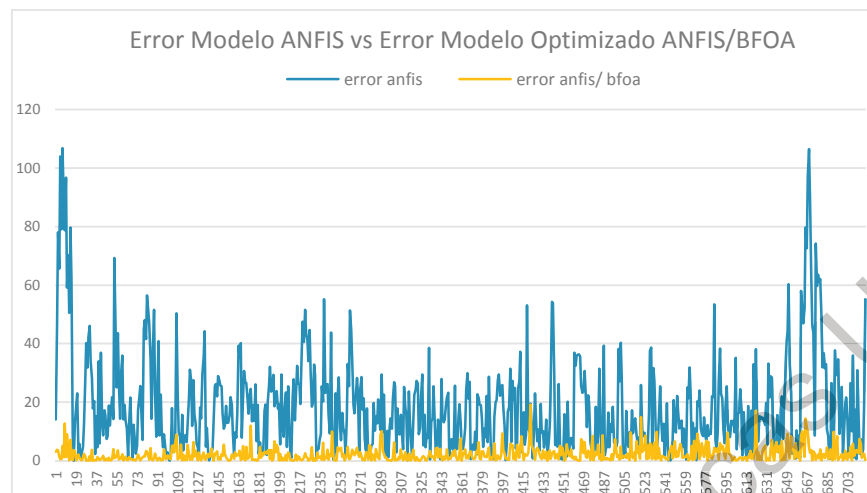
**Fig. 7.** Aumento de tiempo de ejecución

Otro parámetro con el que se experimentó es el número de pasos de quimiotaxis,  $N_c$ , cuando este parámetro tiene un valor grande significa que tendremos numerosos pasos quimiotácticos lo cual podría significar un mayor avance de optimización, en la Fig. 8 podemos observar una disminución en el error cuando  $N_c=10$  y en especial cuando la población  $S$  es mayor ( $S=500$ ), pero, esto supone también una mayor complejidad computacional. Si elegimos que el tamaño de  $N_c$  va a ser muy pequeño, el algoritmo va a depender de más de la suerte al generarse la población aleatoria y la reproducción de las mejores bacterias, incluso en algunos casos, podría quedar atorado en un mínimo local (convergencia prematura).

En la Fig. 9 se observa el error correspondiente al método ANFIS con respecto a los datos reales (línea azul), en comparación con la tasa de error del modelo optimizado ANFIS/ BFOA respecto a los datos reales (línea naranja), esta comparación muestra que al ser menor su tasa de error, el modelo optimizado con BFOA se ajusta mejor a los datos reales que el método ANFIS, por lo tanto, sí puede ser optimizado el modelo usando éste método.



**Fig. 8.** Diferentes valores para  $N_c$  (pasos de quimiotaxis)



**Fig. 9.** Comparación de errores del Modelo ANFIS y Modelo optimizado BFOA/ANFIS

## Conclusiones

Este trabajo muestra que se puede optimizar un modelo previo construido en ANFIS utilizando el algoritmo de optimización de forrajeo bacteriano (BFOA). El incremento de ajuste del modelo optimizado depende en su mayoría de los parámetros de tamaño de la población  $S$  y número de pasos quimiotácticos, pasos de nado, por mencionar algunos. Y aunque se demostró que el método sí presentó una mejora en el modelo en cuanto a error, dependerá del problema, su dimensionalidad y los requerimientos de optimización donde sea aplicado ya que también demostró aumentar la complejidad del tiempo de cómputo lo cual podría ser un problema si se cuenta con recursos limitados para construir el modelo.

En futuros trabajos se puede utilizar el algoritmo de forma híbrida con otros métodos para buscar una forma de reducir el tiempo de ejecución e investigar si su rendimiento mejora.

## Referencias

- Kelly, F. J., & Fussell, J. C. (2012). Size, source and chemical composition as determinants of toxicity attributable to ambient particulate matter. *Atmospheric environment*, 60, 504-526.
- Brook, R. D., Rajagopalan, S., Pope III, C. A., Brook, J. R., Bhatnagar, A., Diez-Roux, A. V., & Peters, A. (2010). Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the American Heart Association. *Circulation*, 121(21), 2331-2378.
- Tellez-Rojo, M. M., Romieu, I., Ruiz-Velasco, S., Lezana, M. A., & Hernandez-Avila, M. M. (2000). Daily respiratory mortality and PM10 pollution in Mexico City: importance of considering place of death. *European Respiratory Journal*, 16(3), 391-396.
- Cortina-Januchs, M. G., Quintanilla-Dominguez, J., Vega-Corona, A., & Andina, D. (2015). Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. *Atmospheric Pollution Research*, 6(4), 626-634.

- Moustris, K. P., Ziomas, I. C., & Paliatsos, A. G. (2010). 3-Day-ahead forecasting of regional pollution index for the pollutants NO<sub>2</sub>, CO, SO<sub>2</sub>, and O<sub>3</sub> using artificial neural networks in Athens, Greece. *Water, Air, & Soil Pollution*, 209(1-4), 29-43.
- Hooyberghs, J., Mensink, C., Dumont, G., Fierens, F., & Brasseur, O. (2005). A neural network forecast for daily average PM<sub>10</sub> concentrations in Belgium. *Atmospheric Environment*, 39(18), 3279-3289.
- Alkasassbeh, M., Sheta, A. F., Faris, H., & Turabieh, H. (2013). Prediction of PM<sub>10</sub> and TSP air pollution parameters using artificial neural network autoregressive, external input models: a case study in salt, Jordan. *Middle-East Journal of Scientific Research*, 14(7), 999-1009.
- Zarandi, M. F., Faraji, M. R., & Karbasian, M. (2012). Interval type-2 fuzzy expert system for prediction of carbon monoxide concentration in mega-cities. *Applied soft computing*, 12(1), 291-301.
- Aceves-Fernandez, M. A., Pedraza-Ortega, J. C., Sotomayor-Olmedo, A., Ramos-Arreguín, J. M., Vargas-Soto, J. E., & Tovar-Arriaga, S. (2014). Analysis of key features of non-linear behaviour using recurrence quantification. Case study: urban airborne pollution at Mexico City. *Environmental Modeling & Assessment*, 19(2), 139-152.
- Aceves-Fernandez MA, Estrada AL, Pedraza-Ortega JC, Gorrostieta-Hurtado E, Tovar-Arriaga S (2015). Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models. *Int. J. Environ. Sci. Toxic.* Vol. 3(1):xxx-xxx
- Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and biological systems: towards a new bionics?* (pp. 703-712). Springer, Berlin, Heidelberg.
- Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics*, 26(1), 29-41.
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Hosseini, H. S. (2007, September). Problem solving by intelligent water drops. In 2007 IEEE congress on evolutionary computation (pp. 3226-3231). IEEE.
- Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Springer, Berlin, Heidelberg.
- Hedayatzadeh, R., Salmassi, F. A., Keshtgari, M., Akbari, R., & Ziarati, K. (2010, May). Termite colony optimization: A novel approach for optimizing continuous problems. In 2010 18th Iranian Conference on Electrical Engineering (pp. 553-558). IEEE.
- Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems magazine*, 22(3), 52-67.
- Tripathy, M., & Mishra, S. (2007). Bacteria foraging-based solution to optimize both real power loss and voltage stability limit. *IEEE Transactions on Power Systems*, 22(1), 240-248.

- Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In *Foundations of Computational Intelligence Volume 3* (pp. 23-55). Springer, Berlin, Heidelberg.
- Ali, E. S., & Abd-Elazim, S. M. (2013). BFOA based design of PID controller for two area Load Frequency Control with nonlinearities. *International Journal of Electrical Power & Energy Systems*, 51, 224-231.
- Wu, C., Zhang, N., Jiang, J., Yang, J., & Liang, Y. (2007, April). Improved bacterial foraging algorithms and their applications to job shop scheduling problems. In *International Conference on Adaptive and Natural Computing Algorithms* (pp. 562-569). Springer, Berlin, Heidelberg.
- Mahapatra, G., & Banerjee, S. (2013). A study of bacterial foraging optimization algorithm and its applications to solve simultaneous equations. *International Journal of Computer Applications*, 72(5), 1-6.
- Chen, Y. P., Li, Y., Wang, G., Zheng, Y. F., Xu, Q., Fan, J. H., & Cui, X. T. (2017). A novel bacterial foraging optimization algorithm for feature selection. *Expert Systems with Applications*, 83, 1-17.
- Narendhar, S., & Amudha, T. (2012). A hybrid bacterial foraging algorithm for solving job shop scheduling problems. *arXiv preprint arXiv:1211.4971*.
- Abd-Elazim, S. M., & Ali, E. S. (2013). A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design. *International Journal of Electrical Power & Energy Systems*, 46, 334-341.
- Azizipanah-Abarghoee, R. (2013). A new hybrid bacterial foraging and simplified swarm optimization algorithm for practical optimal dynamic load dispatch. *International Journal of Electrical Power & Energy Systems*, 49, 414-429.
- Sistema de Monitoreo Atmosférico (SIMAT). El monitoreo de la calidad del aire. Recuperado de <http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnmI=%27>
- Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.
- Abdulshahed, A. M., Longstaff, A. P., & Fletcher, S. (2015). The application of ANFIS prediction models for thermal error compensation on CNC machine tools. *Applied Soft Computing*, 27, 158-168.
- Hansen, C. H., Endres, R. G., & Wingreen, N. S. (2008). Chemotaxis in *Escherichia coli*: a molecular model for robust precise adaptation. *PLoS computational biology*, 4(1), e1.
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), 679-688.