

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

M.C. Cabrera-Hernández, J.A. Ramírez-Montañez, M.A. Aceves-Fernández,  
J.C. Pedraza-Ortega, J.M. Ramos-Arrequin

Universidad Autónoma de Querétaro, Facultad de Ingeniería,  
Querétaro, México

maria.delcarmen.cabrera@uaq.mx, julio.ramirez27@gmail.com,  
marco.aceves@uaq.mx, caryoko@yahoo.com, jramos@mecamex.net

**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 for Improving 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.

## 1. Introducción

### 1.1. Material particulado

El material particulado (MP) o partículas atmosféricas son piezas pequeñas de materia suspendidas en la atmósfera 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.

### 1.2. 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].

## 2. Metodología y materiales

### 2.1. 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.

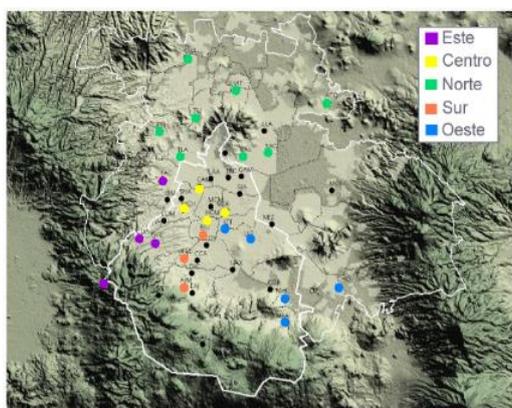


Fig. 1. Mapa de las estaciones de RAMA.

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

## 2.2. 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:

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

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:

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

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).

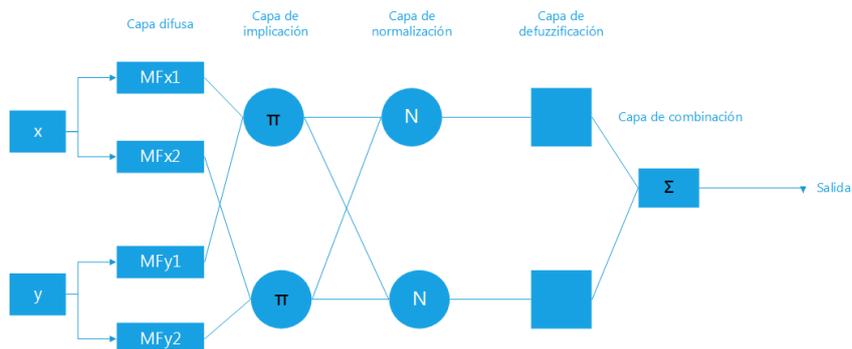


Fig. 2. Estructura ANFIS (adaptado de [25]).

### 2.3. 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 Fig. 3 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.

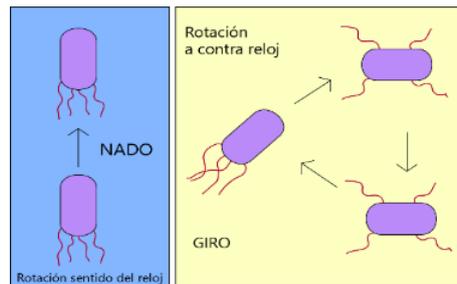


Fig. 3. Quimiotaxis de una bacteria (traducido de [17]).

Si suponemos que buscamos encontrar el mínimo de  $J(\theta)$  donde  $\theta \in \mathfrak{R}^p$  (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 el paso quimiotáctico,  $k$  el índice del paso de reproducción, y por último  $l$  es el índice es 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  $P(j, k, l) = \theta^i(j, k, l)$  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:

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

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.

## 2.4. Metodología propuesta

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

1. Adquisición de datos: Los datos puros para crear el modelo son obtenidos de una base de datos.
2. 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.
3. Entrada de datos: una vez que los datos han sido revisados y son válidos, entonces pueden ser alimentados al modelo.

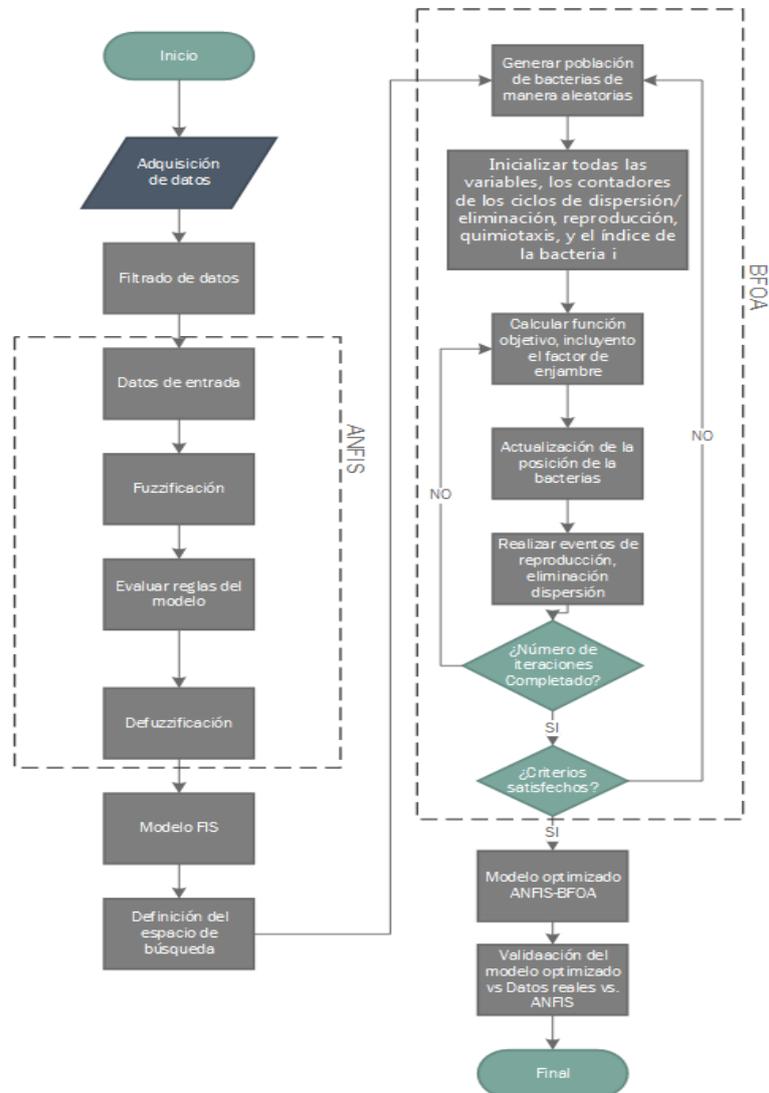


Fig. 4. Modelo optimizado.

4. Fuzzificación: Es el proceso de convertir en los datos de entrada en un valor lingüístico, esto depende de las funciones de membresía (ecuaciones 1 y 2).
5. Evaluación de las reglas: las reglas del modelo son evaluadas con respecto a las reglas difusas y los valores de las funciones de membresía (ecuaciones 3 y 4).
6. Defuzzificación: El método de defuzzificación utilizado fue el de promedio ponderado de todas las salidas de reglas (ecuación 5).
7. 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 (ecuación 6).

8. 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
9. 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.
10. 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.

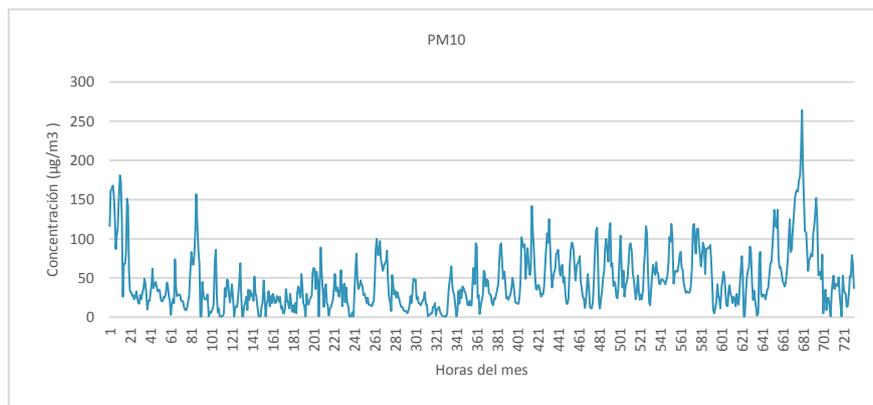
11. Cálculo de la función objetivo: Se debe de calcular el criterio que nos dirá que tan saludable está nuestra bacteria, en este algoritmo se agrega el cálculo del factor de enjambre  $J_{cc}$  con la ecuación 8 y agregarse al costo J de la bacteria en la posición actual. Actualizar posición de las 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 (ecuación 7) 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.
12. 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.

13. 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.

### 3. 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.

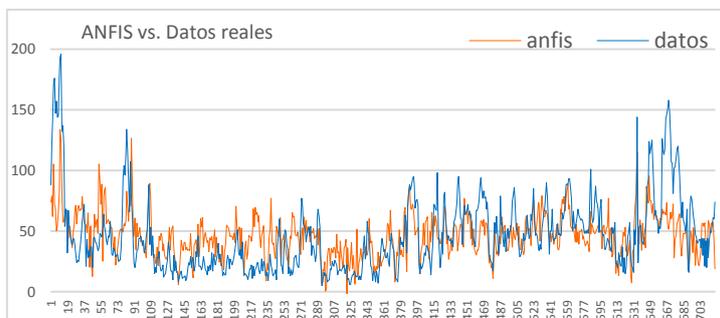


Fig. 6. Comparación datos reales vs ANFIS.

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

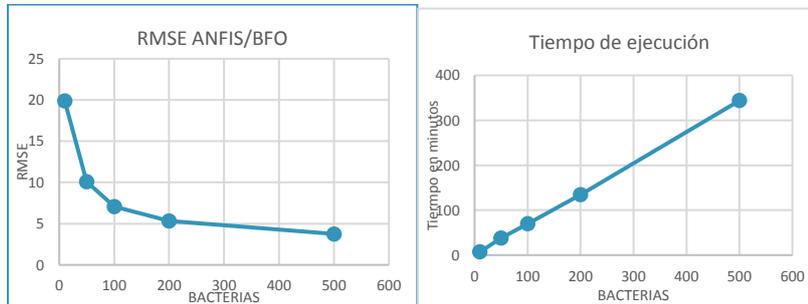


Fig. 7. Aumento de tiempo de ejecución.

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.

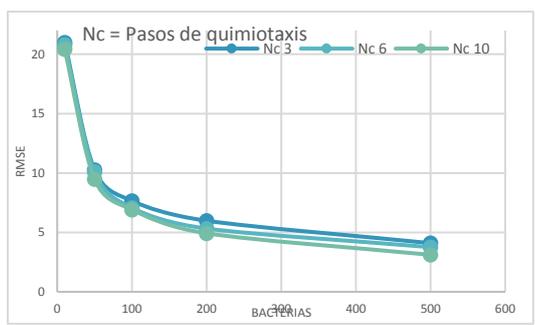


Fig. 8. Diferentes valores para Nc (pasos de quimiotaxis).

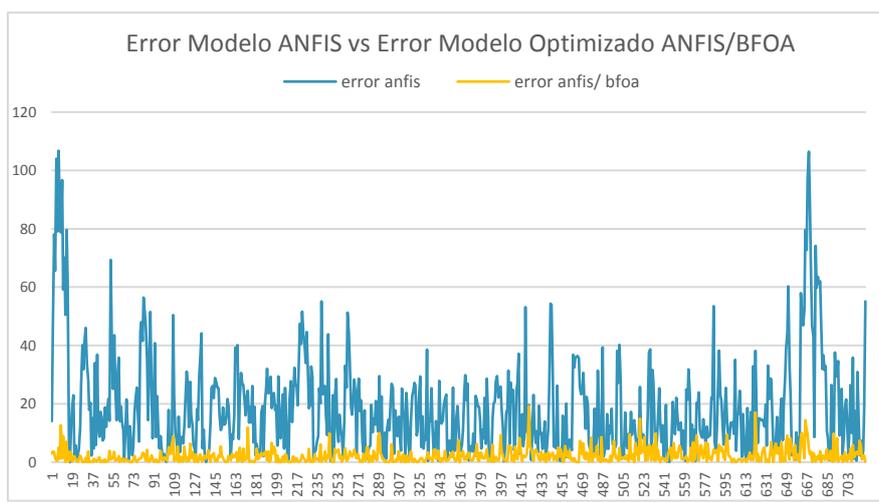


Fig. 9. Comparación de errores del Modelo ANFIS y Modelo optimizado BFOA/ANFIS.

#### 4. 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

1. Kelly, F.J., Fussell, J.C.: Size, source and chemical composition as determinants of toxicity attributable to ambient particulate matter. *Atmospheric environment*, 60, pp. 504–526 (2012)
2. Brook, R.D., Rajagopalan, S., Pope III, C.A., Brook, J.R., Bhatnagar, A., Diez-Roux, A.V., Peters, A.: Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the American Heart Association. *Circulation* 121(21), 2331–2378 (2010)
3. Tellez-Rojo, M.M., Romieu, I., Ruiz-Velasco, S., Lezana, M.A., Hernandez-Avila, M.M.: Daily respiratory mortality and PM10 pollution in Mexico City: importance of considering place of death. *European Respiratory Journal*, 16(3), 391–396 (2000)
4. Cortina-Januchs, M.G., Quintanilla-Dominguez, J., Vega-Corona, A., Andina, D.: Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. *Atmospheric Pollution Research*, 6(4), 626–634 (2015)
5. Moustris, K.P., Ziomas, I.C., Paliatsos, A.G.: 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 (2010)
6. Hooyberghs, J., Mensink, C., Dumont, G., Fierens, F., Brasseur, O.: A neural network forecast for daily average PM10 concentrations in Belgium. *Atmospheric Environment*, 39(18), 3279–3289 (2005)
7. Alkasasbeh, M., Sheta, A.F., Faris, H., Turabieh, H.: 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 (2013)
8. Zarandi, M.F., Faraji, M.R., Karbasian, M.: Interval type-2 fuzzy expert system for prediction of carbon monoxide concentration in mega-cities. *Applied soft computing*, 12(1), 291–301 (2012)
9. Aceves-Fernandez, M.A., Pedraza-Ortega, J.C., Sotomayor-Olmedo, A., Ramos-Arreguín, J.M., Vargas-Soto, J.E., Tovar-Arriaga, S.: Analysis of key features of non-linear behavior using recurrence quantification. Case study: urban airborne pollution at Mexico City. *Environmental Modeling & Assessment*, 19(2), 139–152 (2014)
10. Aceves-Fernandez, M.A., Estrada, A.L., Pedraza-Ortega, J.C., Gorrostieta-Hurtado, E., Tovar-Arriaga, S.: Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models. *Int. J. Environ. Sci. Toxicol.* vol. 3(1), (2015)
11. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: *Robots and biological systems: towards a new bionics?* Springer, Berlin, Heidelberg, w, pp. 703–712 (1993)
12. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics*, 26(1), 29–41 (1996)
13. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, vol. 200. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005)
14. Hosseini, H.S.: Problem solving by intelligent water drops. In: *2007 IEEE congress on evolutionary computation*, pp. 3226–3231 (2007)

15. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, Berlin, Heidelberg, pp. 65–74 (2010)
16. Hedayatzadeh, R., Salmassi, F.A., Keshtgari, M., Akbari, R., Ziarati, K.: Termite colony optimization: A novel approach for optimizing continuous problems. In: 2010 18th Iranian Conference on Electrical Engineering. IEEE, pp. 553–558 (2010)
17. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems magazine*, 22(3), 52–67 (2002)
18. Tripathy, M., Mishra, S.: Bacteria foraging-based solution to optimize both real power loss and voltage stability limit. *IEEE Transactions on Power Systems*, 22(1), 240–248 (2007)
19. Das, S., Biswas, A., Dasgupta, S., Abraham, A.: Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: *Foundations of Computational Intelligence vol. 3*, Springer, Berlin, Heidelberg, pp. 23–55 (2009)
20. Ali, E.S., Abd-Elazim, S.M.: BFOA based design of PID controller for two area Load Frequency Control with nonlinearities. *International Journal of Electrical Power & Energy Systems*, 51, pp. 224–231 (2013)
21. Wu, C., Zhang, N., Jiang, J., Yang, J., Liang, Y.: Improved bacterial foraging algorithms and their applications to job shop scheduling problems. In: *International Conference on Adaptive and Natural Computing Algorithms*. Springer, Berlin, Heidelberg, pp. 562–569 (2007)
22. Mahapatra, G., Banerjee, S.: A study of bacterial foraging optimization algorithm and its applications to solve simultaneous equations. *International Journal of Computer Applications*, 72(5), 1–6 (2013)
23. Chen, Y.P., Li, Y., Wang, G., Zheng, Y.F., Xu, Q., Fan, J.H., Cui, X.T.: A novel bacterial foraging optimization algorithm for feature selection. *Expert Systems with Applications* 83, pp. 1–17 (2017)
24. Narendhar, S., Amudha, T.: A hybrid bacterial foraging algorithm for solving job shop scheduling problems. *arXiv preprint arXiv:1211.4971* (2012)
25. Abd-Elazim, S.M., Ali, E.S.: A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design. *International Journal of Electrical Power & Energy Systems*, 46, pp. 334–341 (2013)
26. Azizipanah-Abarghooee, R.: A new hybrid bacterial foraging and simplified swarm optimization algorithm for practical optimal dynamic load dispatch. *International Journal of Electrical Power & Energy Systems* 49, pp. 414–429 (2013)
27. Sistema de Monitoreo Atmosférico (SIMAT). El monitoreo de la calidad del aire, <http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnmI=%27>. Último acceso: 2019/05/01
28. Jang, J.S.: ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665–685 (1993)
29. Abdulshahed, A.M., Longstaff, A.P., Fletcher, S.: The application of ANFIS prediction models for thermal error compensation on CNC machine tools. *Applied Soft Computing* 27, pp.158–168 (2015)
30. Hansen, C.H., Endres, R.G., Wingreen, N.S.: Chemotaxis in *Escherichia coli*: a molecular model for robust precise adaptation. *PLoS computational biology*, 4(1), e1 (2008)
31. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), 679–688 (2006)