



**UJAT**

UNIVERSIDAD JUÁREZ  
AUTÓNOMA DE TABASCO

“ESTUDIO EN LA DUDA. ACCIÓN EN LA FE”



# Universidad Juárez Autónoma de Tabasco

Tesis de Maestría

## Impacto de la distribución poblacional en el algoritmo basado en el forrajeo de bacterias

Que presenta

**Margarita Ivette Hernández Hernández**

Para obtener el grado de

**Maestro en Ciencias de la Computación**

Directora

**Dra. Betania Hernández Ocaña**

**Dr. José Adán Hernández Nolasco**

Cuerpo Académico

**Inteligencia Artificial**

Línea de Generación y Aplicación del Conocimiento

**Ciencia de Datos y Optimización**

Cunduacán, Tabasco, México

Julio 2022



**UJAT**

UNIVERSIDAD JUÁREZ  
AUTÓNOMA DE TABASCO

“ESTUDIO EN LA DUDA. ACCIÓN EN LA FE”



# Universidad Juárez Autónoma de Tabasco

Tesis de Maestría

## Impacto de la distribución poblacional en el algoritmo basado en el forrajeo de bacterias

Que presenta

**Margarita Ivette Hernández Hernández**

Para obtener el grado de

**Maestro en Ciencias de la Computación**

Directora

**Dra. Betania Hernández Ocaña**

**Dr. José Adán Hernández Nolasco**

Jurado:	<b>Dr. José Hernández Torruco</b>	Presidente
	<b>Dra. Juana Canul Reich</b>	Secretario
	<b>Dr. Oscar Alberto Chávez Bosquez</b>	Vocal
	<b>Dra. Betania Hernández Ocaña</b>	Vocal
	<b>Dr. José Adán Hernández Nolasco</b>	Vocal

Cuerpo Académico  
**Inteligencia Artificial**

Línea de Generación y Aplicación del Conocimiento  
**Ciencia de Datos y Optimización**

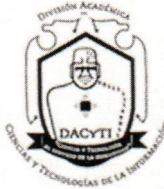
Cunduacán, Tabasco, México

Julio 2022



**UNIVERSIDAD JUÁREZ  
AUTÓNOMA DE TABASCO**

"ESTUDIO EN LA DUDA. ACCIÓN EN LA FE"



**DIVISIÓN ACADÉMICA DE  
CIENCIAS Y TECNOLOGÍAS  
DE LA INFORMACIÓN**



PRECURSOR DE LA REVOLUCIÓN MEXICANA

"2022, Año de Ricardo Flores Magón"

Cunduacán, Tabasco a 12 de mayo de 2022

Oficio No. 0527/DACYTI/CP/2022

Asunto: Autorización de impresión de Tesis

**C. Margarita Ivette Hernández Hernández**  
**Matricula: 192H13004**

En virtud de que cumple satisfactoriamente los requisitos establecidos en el Reglamento General de Estudios de Posgrado vigente en la Universidad, informo a Usted que se autoriza la impresión del trabajo recepcional **"IMPACTO DE LA DISTRIBUCIÓN POBLACIONAL EN EL ALGORITMO BASADO EN EL FORRAJE DE BACTERIAS"**, para presentar examen y obtener el Grado de Maestro en Ciencias de la Computación.

Sin otro particular, aprovecho la ocasión para enviarle un afectuoso saludo.

**Atentamente**

**MTE. Oscar Alberto González González**  
**Director**

UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO



DIVISIÓN ACADÉMICA DE  
CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

C.c.p. Dr. Eddy Arquímedes García Alcocer. - Encargado del Despacho de la Coordinación de Posgrado DACYTI  
Archivo.  
Consecutivo.

MTE/OAGG/EAGA

X

Carretera Cunduacán-Jalpa Km. 1, Colonia Esmeralda, C.P. 86690.  
Cunduacán, Tabasco, México.  
Tel: (993) 358 1500 ext. 6727; (914) 336 0616; Fax: (914) 336 0870  
E-mail: direccion.dacyti@ujat.mx

www.ujat.mx


## Carta de Autorización.

La que suscribe, autoriza por medio del presente escrito a la Universidad Juárez Autónoma de Tabasco para que utilice tanto física como digitalmente la Tesis "**Impacto de la distribución poblacional en el algoritmo basado en el forrajeo de bacterias**", de la cual soy autor y titular de los Derechos de Autor.

La finalidad del uso por parte de la Universidad Juárez Autónoma de Tabasco de la tesis antes mencionada, será única y exclusivamente para difusión y educación, sin fines de lucro; la autorización se hace de manera enunciativa más no limitativa para subirla a cualquier Red Académica con las que la Universidad tenga relación institucional.

Por lo antes mencionado, libero a la Universidad Juárez Autónoma de Tabasco de cualquier reclamación legal que pudiera ejercer respecto al uso y manipulación de la Tesis mencionada.

Se firma la presente autorización en la ciudad de Cunduacán, Tabasco a 18 de julio de 2022.

  
C. Margarita Ivette Hernández Hernández.



UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE CIENCIAS  
Y TECNOLOGÍAS DE LA INFORMACIÓN



F5: Liberación de dirección de tesis

Cunduacán, Tabasco., a 18 de julio de 2022.

**MTE. Oscar Alberto González González**

Director de la División Académica de Ciencias y Tecnologías de la Información

Presente

Por medio de la presente nos permitimos comunicarle que después de haber concluido la dirección de la Tesis: "**Impacto de la distribución poblacional en el algoritmo basado en el forrajeo de bacterias**", elaborada por la **C. Margarita Ivette Hernández Hernández** de la Maestría en Ciencias de la Computación, consideramos que puede continuar con los trámites para la obtención del grado.

Sin otro particular, aprovechamos la ocasión para enviarle un cordial saludo.

**Atentamente**

**Dra. Betania Hernández Ocaña**

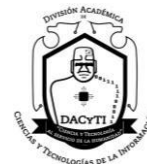
**Dr. José Adán Hernández Nolasco**

c.c.p. Dr. Eddy Arquímedes García Alcocer. Encargada del despacho de la Coordinación de Posgrado.  
Directores de Tesis.  
Estudiante



UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE CIENCIAS  
Y TECNOLOGÍAS DE LA INFORMACIÓN



F6: Respuesta de jurado

Cunduacán, Tabasco, a 18 de julio de 2022.

**MTE. Óscar Alberto González González**  
**Director de la División Académica de Ciencias y Tecnologías de la Información**  
**Presente**

En atención a los oficios girados por usted, en los que se nos designa como parte del jurado para efectuar la revisión de la tesis titulada ***“Impacto de la distribución poblacional en el algoritmo basado en el forrajeo de bacterias”***, realizada por el ***C. Margarita Ivette Hernández Hernández***, estudiante de la Maestría en Ciencias de la Computación, nos permitimos informarle que, en virtud de que ha atendido las observaciones realizadas, otorgamos nuestra aprobación para que continúe los trámites para la obtención del grado.

Sin otro particular, aprovechamos la ocasión para enviarle un cordial saludo.

Atentamente integrantes del jurado

Dra. Juana Canul Reich

Dr. José Hernández Torruco

Dr. Oscar Alberto Chavez Bosquez

c.c.p. Dr. Eddy Arquímedes García Alcocer. Encargado del despacho de la Coordinación de Posgrado. Estudiante.

# Agradecimientos

Quiero agradecer a mis padres Manuel y Margarita por haberme dado su apoyo incondicional durante todos estos años y por ser esa razón el más grande aliciente para el cumplimiento de mis objetivos que significan alegría y orgullo para ellos y para mí.

A mis hermanos Manuel y Miriam y a mis sobrinas Ximena y Vanessa que en el día a día con su presencia, respaldo y cariño me impulsan para salir adelante.

A mis amigos por ser ese punto de apoyo e impulsarme a no rendirme y seguir adelante.

A mis directores de tesis por su apoyo, comprensión y sobre todo paciencia al ayudarme con las cosas que se me dificultaban y a mis maestros por su apoyo siempre.

Pero sobre todo quiero agradecer a Dios ya que gracias a el estoy culminando este proyecto de tesis ya que ha sido mi guía y me ha dado la fortaleza necesaria ante todas las adversidades.

# Índice general

Tabla de contenido	I
Índice de Figuras	III
Índice de Tablas	IV
Resumen	1
<b>1. Generalidades</b>	<b>2</b>
1.1. Introducción . . . . .	2
1.2. Planteamiento del problema . . . . .	4
1.2.1. Definición del problema . . . . .	4
1.2.2. Delimitación de la investigación . . . . .	6
1.3. Pregunta de investigación e hipótesis . . . . .	7
1.4. Objetivo general . . . . .	7
1.5. Objetivos específicos . . . . .	7
1.6. Justificación . . . . .	8
1.7. Metodología utilizada . . . . .	8
<b>2. Marco teórico</b>	<b>11</b>
2.1. Conceptos y teorías fundamentales de la investigación . . . . .	11
2.1.1. Algoritmo de optimización basado en bacterias (BFOA) . . . . .	11
2.1.2. Algoritmo de Optimización Modificado Basado en el Forrajeo de Bacterias(MBFOA) . . . . .	11
2.1.3. Algoritmo de Optimización Modificado Basado en el Forrajeo de Bacterias de Doble Nado(TS-MBFOA) . . . . .	12
2.1.4. Distribución Uniforme . . . . .	13
2.1.5. Distribución Gaussiana . . . . .	13



2.1.6.	Distribución Exponencial . . . . .	14
2.1.7.	Bacterias E. Coli . . . . .	15
2.1.8.	Medidas de Estadísticas Básicas . . . . .	16
2.1.9.	Kruskal-Wallis Test . . . . .	17
2.2.	Literatura relacionada . . . . .	17
2.3.	Marco tecnológico . . . . .	25
<b>3.</b>	<b>Modelo de las distribuciones poblacionales y el reinicio del TS-MBFOA</b>	<b>26</b>
3.1.	Distribución Uniforme . . . . .	26
3.2.	Distribución Gaussiana . . . . .	26
3.3.	Distribución Exponencial . . . . .	27
3.4.	TS-MBFOA con reinicio de población . . . . .	28
<b>4.</b>	<b>Experimentos y Resultados</b>	<b>30</b>
4.1.	Problemas . . . . .	30
4.1.1.	Esfera . . . . .	30
4.1.2.	P1: Resorte de tensión/compresión . . . . .	31
4.1.3.	P2: Recipiente de presión . . . . .	32
4.1.4.	P3: Viga soldada . . . . .	32
4.1.5.	P4: Síntesis de procesos MINLP . . . . .	34
4.1.6.	P5: Diseño de una viga de concreto reforzada . . . . .	35
4.1.7.	P6: Un problema cuadrático restringido . . . . .	35
4.2.	Experimentos . . . . .	36
4.2.1.	Parámetros . . . . .	36
4.2.2.	Resultados . . . . .	37
4.2.3.	Gráficas de Convergencia . . . . .	39
4.2.4.	Discusión de resultados . . . . .	44
<b>5.</b>	<b>Contribuciones, conclusiones y trabajos futuros</b>	<b>46</b>
5.1.	Contribuciones . . . . .	46
5.2.	Conclusiones . . . . .	46
5.3.	Trabajos Futuros . . . . .	48
	<b>Bibliografía</b>	<b>49</b>

## Índice de figuras

4.1. Población de bacterias con distribución uniforme, gaussiana y exponencial en el problema de la esfera . . . . .	38
4.2. Gráfica de convergencia del P1: Resorte de tensión/compresión con distribución uniforme, gaussiana y exponencial . . . . .	39
4.3. Gráfica de convergencia del P2: Recipiente de presión con distribución uniforme, gaussiana y exponencial . . . . .	41
4.4. Gráfica de convergencia del P3: Viga soldada con distribución uniforme, gaussiana y exponencial . . . . .	42
4.5. Gráfica de convergencia del P4: Síntesis de procesos MINLP con distribución uniforme, gaussiana y exponencial . . . . .	42
4.6. Gráfica de convergencia del P5: Diseño de una viga de concreto reforzada con distribución uniforme, gaussiana y exponencial .	43
4.7. Gráfica de convergencia del P6: Un problema cuadrático restringido con distribución uniforme, gaussiana y exponencial . .	43

## Índice de tablas

4.1. Mejor resultado de TS-MBFOA con diferentes distribuciones .	38
4.2. Estadísticas básicas de las 30 ejecuciones independientes de TS-MBFOA con diferentes distribuciones . . . . .	40
4.3. Resultado de la prueba no paramétrica Kruskal-Wallis Test de las 30 mejores soluciones encontradas en TS-MBFOA con las tres distribuciones . . . . .	41

# Resumen

En la presente tesis se trata el algoritmo de optimización basado en el forrajeo de bacterias de doble nado modificado (TS-MBFOA, por sus siglas en inglés *Two-Swim Modified Bacterial Foraging Optimization Algorithm*), el cual es una evolución del algoritmo BFO original (optimización de forrajeo de bacterias) y se propone mejorar su rendimiento tanto en tiempo como en espacio computacional mediante el uso de una distribución poblacional distinta, ya que normalmente se usa la distribución uniforme.

Los resultados pueden mejorar al modificar el tipo de distribución ya que eso afectaría la forma en que las bacterias son distribuidas dentro del espacio de búsqueda. Se propone usar las funciones de distribución de probabilidad Gaussiana (normal) y Exponencial, se realizaron dos pruebas donde primero se ejecutó con el problema de la esfera por ser el más simple y después con 6 problemas de optimización numérica tomadas de la literatura especializada. Los resultados fueron analizados usando estadística básica, pruebas no paramétricas y gráficos de convergencia, para así ser comparados con los resultados obtenidos con los otros tipos de distribución.

Después de realizadas las pruebas, se concluye que aunque el algoritmo TSMBFOA tiene resultados diferentes en cada problema de prueba con las tres distribuciones abordadas en este trabajo, se puede obtener un mejor rendimiento del TS-MBFOA usando la distribución exponencial.

# Capítulo 1

## Generalidades

### 1.1. Introducción

Las metaheurísticas son herramientas que permiten solucionar problemas de optimización con o sin restricciones, combinatorios o numéricos de una forma aproximada, es decir generan una o más soluciones factibles cercanas al óptimo [14]. Un Problema de Optimización Numérica con Restricciones (PONRs), se conoce como el problema general de programación no-lineal y se consigue definir como [14]:

Minimizar  $f(\vec{x})$  sujeta a:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \dots, m,$$

$$h_j(\vec{x}) = 0, \quad j = 1, \dots, p,$$

En el que  $\vec{x} \in R^n$  donde  $n \geq 1$ , es el vector de soluciones  $\vec{x} = [x_1, x_2, \dots, x_n]^T$ , en que cada  $x_i$ ,  $i = 1, \dots, n$  está limitada por el límite inferior y superior  $L_i \leq x_i \leq U_i$ ;  $m$  es la representación de restricciones de desigualdad y  $p$  es la representación de restricciones de igualdad (estas restricciones pueden ser lineales o no lineales). Si expresamos con  $F$  la región factible y con  $S$  a todo el espacio de búsqueda, entonces debe ser claro que  $F \subseteq S$  [14].

Los algoritmos inspirados en la naturaleza son metaheurísticas que inicialmente resolvían problemas de optimización sin restricciones y debido a la implementación de ellos en problemas del mundo real se fueron creando técnicas para analizar problemas de optimización con restricciones, tales como las Reglas de Deb, Función de penalización, decodificadores, entre otras, que

funcionan con manejadores de restricciones [14].

Los algoritmos bioinspirados se dividen en dos grandes grupos: los Algoritmos Evolutivos (AEs), estos están basados en la teoría de la evolución de Charles Darwin y la conservación del más apto; y los Algoritmos de Inteligencia Colectiva (IC) que se fundamentan en el comportamiento social y cooperativo de organismos simples e inteligentes como por ejemplo insectos y aves [13].

En el año 2002 Kevin Passino propone BFOA en el cual las bacterias *E. Coli* tratan de maximizar la energía que obtiene la bacteria por módulo de tiempo usada en el transcurso del forrajeo, en el que a su vez esquiva sustancias nocivas. Las bacterias se pueden comunicar entre ellas por medio de la segregación de sustancias [21].

A lo largo del tiempo el Algoritmo BFOA ha tenido modificaciones en busca de mejorar su rendimiento, una de esas mejoras es el Algoritmo BFOA modificado (MBFOA), el cual hace una disminución de los parámetros y ciclos del algoritmo y a su vez aplica un componente para la administración de las reglas de factibilidad de Deb [18]. Otra propuesta es el *Improved* MBFOA donde se efectúa un módulo de sesgo para introducir la población inicial que contienen dos operadores de nado, tamaños de paso dinámico y un buscador local [15]. El Algoritmo TS-MBFOA es otra de las mejoras que ha tenido el Algoritmo BFOA a lo largo del tiempo y consiste en intercalar dos nados (uno de exploración y otro de explotación) en el proceso quimiotáxico, el nado de explotación se le conoce como el nado original y es el que se hace primero con una capacidad de paso aleatorio para posteriormente realizar el nado de exploración que incluye el operador de mutación para así mejorar la capacidad de explotación del Algoritmo [13].

Todas estas mejoras del algoritmo utilizan la distribución uniforme para obtener sus resultados, por lo que se piensa que los resultados de los problemas del benchmark pueden mejorar debido al tipo de distribución poblacional que se utilice, ya que afectará en la forma en la que las bacterias serán distribuidas aleatoriamente dentro del espacio de búsqueda logrando así que el algoritmo encuentre soluciones más rápido.

En la presente tesis se implementarán nuevos tipos de distribuciones como la Gaussiana y Exponencial, ya que su distribución por defecto es la uniforme

y usar una diferente puede mejorar el resultado del algoritmo de manera óptima, se buscará resolver 6 PONR encontrados en el estado del arte, los cuales son:

- Resorte de tensión/compresión
- Viga soldada
- Recipiente de presión
- Diseño de una viga de concreto reforzada
- Síntesis de procesos MINLP
- Un problema cuadrático restringido

Además, del problema de la esfera como prueba inicial con el objetivo de visualizar de manera gráfica el comportamiento de las bacterias con las diferentes distribuciones.

## 1.2. Planteamiento del problema

### 1.2.1. Definición del problema

El Algoritmo basado en el forrajeo de bacterias es un Algoritmo de optimización numérica sin restricciones (BFOA) sugerido por K.M. Passino que simula el transcurso de las bacterias *E. Coli* y fue propuesto para resolver problemas de optimización sin restricciones, sin embargo, versiones posteriores mejoraron el rendimiento del algoritmo reduciendo sus procesos y sus parámetros utilizados. Estas mejoras hicieron capaz al algoritmo de resolver problemas mono-objetivo y multi-objetivo de optimización numérica con y sin restricciones.

El Algoritmo de Optimización Modificado Basado en el Forrajeo de Bacterias de Doble Nado (TS-MBFOA) se basa en hacer una doble exploración en cada ciclo. Es normalmente utilizado con una población aleatoria de distribución uniforme; en la búsqueda realizada en la literatura especializada no se encuentra ningún estudio del Algoritmo basado en el forrajeo de bacterias que use otro tipo de distribución para generar su población inicial, es decir, se

desconoce si el algoritmo puede mejorar su rendimiento, en tiempo y utilizar menos espacio de memoria de la computadora.

En TS-MBFOA se hace uso de un mecanismo de sesgo que genera una población inicial distribuida uniformemente, sobre todo el área de búsqueda de la complicación a resolver. En este mecanismo de sesgo 3 grupos de bacterias son generados, uno con sesgo a los límites inferiores, otro hacia los límites superiores y un grupo totalmente aleatorio dentro del espacio de búsqueda. En los problemas de optimización resueltos, TS-MBFOA con mecanismo de sesgo obtuvo un mejor rendimiento que sus versiones anteriores como MBFOA e IMBFOA. Sin embargo, la generación aleatoria de este mecanismo de sesgo fue totalmente uniforme. Se piensa que los resultados de los problemas del benchmark mejorarán debido al tipo de distribución poblacional que se utilice, ya que afectara en la forma en la que las bacterias serán distribuidas aleatoriamente dentro del espacio de búsqueda logrando así que el algoritmo encuentre las soluciones más rápido.

En este trabajo se buscará mejorar el rendimiento del algoritmo probando 3 diferentes tipos de distribución para generar la población inicial del algoritmo y reinicializar un cierto número de población de las bacterias cada cierta generación del algoritmo, con el objetivo de dar diversidad poblacional y proveer a los operadores de nado nuevos espacios de búsqueda y evitar los óptimos locales en los que pueda caer el algoritmo. Los dos tipos de distribución: gaussiana y exponencial fueron seleccionadas de la literatura especializada por su simplicidad de uso y porque no incrementan el número de parámetros del algoritmo a calibrar por el usuario final. Además, estas distribuciones permiten explorar y generar puntos (regiones) distintas del espacio de búsqueda.

Las nuevas propuestas serán probadas en problemas de optimización numérica con restricciones que se basan tanto en diseño ingenieril como químico conocidos en la literatura como: Resorte de tensión/compresión, viga soldada, recipiente de presión, diseño de una viga de concreto reforzada, Síntesis de procesos MINLP y un problema cuadrático restringido.

Los resultados serán analizados usando estadísticas básicas como: mejor, mediana, desviación estándar y peor resultado, pruebas no paramétricas como Kruskal-Wallis test y gráficos de convergencia y una métrica que permita conocer cuánto ha progresado una solución factible con el paso de las genera-



ciones, con el objetivo de medir la mejora continua del algoritmo a partir de una primera solución factible encontrada hasta el final de sus generaciones. Cabe mencionar que esta métrica será buscada en la literatura especializada o se pretende proponerla dando prioridad al no incremento de parámetros del algoritmo.

### 1.2.2. Delimitación de la investigación

#### Alcances

- La propuesta de las distribuciones poblacionales encontró en todos los experimentos soluciones factibles.
- Los resultados obtenidos se evaluaron usando medidas estadísticas para conocer la calidad de los resultados del algoritmo. Las medidas estadísticas utilizadas son:
  - Mejor
  - Media
  - Peor
  - Desviación estándar
- Los resultados se compararon con las otras distribuciones (Uniforme, Gaussiana, Exponencial).

#### Limitaciones

- Se estudió el rendimiento del algoritmo a través de los resultados obtenidos al resolver 6 problemas de pruebas:
  - Resorte de tensión/compresión.
  - Viga soldada.
  - Recipiente de presión.
  - Diseño de una viga de concreto reforzada.
  - Síntesis de procesos MINLP.
  - Un problema cuadrático restringido.

- Las pruebas fueron realizadas con 3 distribuciones poblacionales:
  - Distribución Uniforme.
  - Distribución Gaussiana.
  - Distribución Exponencial.

### 1.3. Pregunta de investigación e hipótesis

- ¿Qué tipo de distribución (uniforme, gaussiana o exponencial) usada para generar la población inicial aleatoria permite un mejor rendimiento del Algoritmo TS-MBFOA ante problemas de optimización numérica con restricciones?

La utilización de una distribución uniforme, gaussiana o exponencial para generar la población de bacterias del TS-MBFOA impacta en su rendimiento ante problemas de optimización numérica con restricciones.

### 1.4. Objetivo general

Analizar y comparar el rendimiento del Algoritmo TS-MBFOA utilizando diferentes tipos de distribución poblacional, tales como Gaussiana, Exponencial y Uniforme.

### 1.5. Objetivos específicos

- Analizar los tipos de distribución y el Algoritmo TS-MBFOA encontrados en el estado del arte y la literatura especializada.
- Codificar e implementar el Algoritmo TS-MBFOA con las 2 diferentes distribuciones poblacionales (Gaussiana, Exponencial).
- Resolver el problema de la esfera y 6 problemas de diseño ingenieril y químico encontrados en la literatura especializada (benchmark) con las 2 distribuciones mencionadas.

- Analizar los resultados obtenidos con el Algoritmo TS-MBFOA con las 2 diferentes distribuciones poblacionales en los 6 problemas del benchmark usando estadística básica, graficas de convergencia y pruebas no paramétricas como Kruskal-Wallis test. Además del uso de una métrica de progreso.

## 1.6. Justificación

El trabajo propuesto es una técnica novedosa para resolver problemas de optimización empleando el TS-MBFOA utilizando una distribución de población diferente como la Gaussiana y la Exponencial, ya que hasta el momento no se ha registrado ningún escrito de alguna investigación de este tipo. El impacto que se obtendrá al tener la resolución de este problema es encontrar un mejor resultado con menos costo computacional al usar una distribución de población diferente a la uniforme.

Aunque tiene diferentes áreas de aplicación, este proyecto permite resolver problemas del área ingenieril y se busca optimizar los resultados de algunos problemas del benchmark y no estancarse en óptimos locales sino encontrar el óptimo global.

El objetivo al realizar este experimento es analizar y comparar el rendimiento del Algoritmo TS-MBFOA al resolver problemas complejos de optimización numérica con restricciones usando diferentes distribuciones al generar la población de bacterias, se espera encontrar que distribución o distribuciones permiten un mejor rendimiento de este algoritmo.

## 1.7. Metodología utilizada

El enfoque a utilizar es cuantitativo, ya que se utilizarán medidas de rendimiento como: estadísticas básicas (media, mediana, desviación estándar, mejor y peor resultado), graficas de convergencia y pruebas no paramétricas Kruskal-Wallis Test que permite determinar si sucede una diferencia demostrativa entre los resultados obtenidos.

Las fuentes a consultar son: libros y artículos arbitrados donde se pueda encontrar información sobre los tipos de distribuciones en probabilidad, y su uso en diversos algoritmos. Para los problemas a resolver por el algoritmo y el Algoritmo TS-MBFOA se revisarán artículos arbitrados y tesis de licenciatura, maestría y doctorado.

Los pasos para realizar esta metodología son:

1. Revisión del estado del arte acerca de las distribuciones de probabilidad y el Algoritmo TS-MBFOA.
2. Revisión de los problemas a resolver encontrados en el estado del arte.
3. Comprensión del Algoritmo TS-MBFOA con las diferentes distribuciones seleccionadas (Gaussiana y Exponencial).
4. Instalación y configuración del entorno de programación.
5. Codificación y ejecución del Algoritmo TSMBFOA con las 3 distribuciones para resolver el problema de la esfera.
6. Representación mediante gráficas de la población inicial del problema de la esfera con las 3 distribuciones (Uniforme, Gaussiana y Exponencial).
7. Medición del rendimiento del TS-MBFOA con el problema de la esfera mediante estadística básica (mejor, media y desviación estándar) y representar la población inicial y final de cada versión del algoritmo.
8. Codificación y ejecución de los 3 algoritmos con distribución diferente de TS-MBFOA, en la solución de los 6 problemas seleccionados del Benchmark utilizando el mismo valor de parámetros.
9. Obtener tablas de estadística básica y gráficas de convergencia y la prueba de Kruskal-Wallis Test de las 30 ejecuciones del algoritmo.
10. Seleccionar de acuerdo al punto 9 la o las distribuciones que favorecen el rendimiento del TS-MBFOA.

11. Analizar si el reinicio o actualización de la población ayuda a obtener un mejor rendimiento de acuerdo a la estadística básica y prueba de Kruskal-Wallis Test.
12. Medir con un operador ya existente o propuesto, cuánto mejora el algoritmo desde una solución factible hasta el final de las generaciones.

Universidad Juárez Autónoma de Tabasco.  
México.

## Capítulo 2

### Marco teórico

#### 2.1. Conceptos y teorías fundamentales de la investigación

##### 2.1.1. Algoritmo de optimización basado en bacterias (BFOA)

El Algoritmo BFOA simula el procedimiento grupal de las bacteria o animales cuando buscan alimento o resguardo[21], el cual lo modela como un proceso de optimización. En la búsqueda de su alimento las bacterias o animales toman acciones de maximizar la energía que obtiene la bacteria por módulo de tiempo gastado en la búsqueda de alimentos, ante las restricciones presentadas por su propia fisiología (por ejemplo, las capacidades de intuición y cognitivas) y el ambiente (por ejemplo, el peso de la presa, los riesgos de predadores, las sustancias nocivas, las características físicas del área de búsqueda).

##### 2.1.2. Algoritmo de Optimización Modificado Basado en el Forrajeo de Bacterias(MBFOA)

MBFOA es un algoritmo presentado en [18] procedente de BFOA en el que se implementan diversos componentes que aprueban al algoritmo para resolver problemas de optimización mono-objetivo con restricciones. Los mecanismos implementados al BFOA son las siguientes:

1. Un solo lapso para incluir el paso quimiotáxico, la reproducción y eliminación-dispersión.
2. Tamaño de paso diferente para cada variable de diseño.
3. Mecanismos para operar las restricciones del problema fundado en las reglas de factibilidad de Deb [5] utilizadas en los procesos de nado y reproducción.
4. Notificación entre las bacterias y la bacteria sobresaliente de la población de una reproducción.

### 2.1.3. Algoritmo de Optimización Modificado Basado en el Forrajeo de Bacterias de Doble Nado(TS-MBFOA)

El Algoritmo TS-MBFOA intercala dos nados, el inicial es el nado de explotación con una capacidad de paso aleatorio y posteriormente realizar el nado de exploración que contiene un operador de mutación con el fin de optimizar el contenido de exploración y explotación del algoritmo [13].

En este proceso de quimiotáxis ambos nados se intercalan, en cada ciclo solo uno ya sea el nado de explotación o exploración es realizado. Empieza con el nado de original. No obstante, una bacteria no precisamente escalonará los nados, ya que si la nueva colocación de un nado proporcionado,  $\Theta^i (j+1, G)$  tiene una sobresaliente competitividad (basado en las reglas de factibilidad) que la colocación original  $\Theta^i (j, G)$ , otro nado con la semejante dirección se realizara en el ciclo consecutivo. De lo contrario, un giro diferente será calculado. El transcurso se interrumpe después de  $N_c$  pruebas [13].

El nado de exploración manipula la mutación entre baterías y se deduce con la siguiente ecuación:

$$\Theta^i (j+1, G) = \Theta^i (j, G) + (\beta-1) (\Theta_1^r (j, G) - \Theta_2^r (j, G))$$

en el que  $\Theta_1^r(j,G)$  y  $\Theta_2^r(j,G)$  son dos bacterias disparejas elegidas aleatoriamente de la población.  $\beta$  es un parámetro determinado por el usuario manipulado en el operador de agrupamiento el cual especifica la proximidad de la nueva colocación de una bacteria basándose en la colocación de la bacteria sobresaliente de la población, en este operador,  $\beta_1$  es un parámetro de control positivo para escalar los diversos vectores en  $(0,1)$ , es decir, escalas de la zona donde la bacteria puede moverse [13].

El nado de explotación se calcula con la siguiente ecuación:

$$\Theta^i(j+1, G) = \Theta^i(j, G) + C(i, G) \phi(i)$$

en el que  $\phi(i)$  se deduce con el operador de giro original de BFOA y  $C(i, G)$  es la capacidad de paso aleatorio de cada bacteria reestablecido con la siguiente ecuación:

$$C(i, G) = R^* \Theta(i)$$

#### 2.1.4. Distribución Uniforme

La distribución uniforme es continua y organiza una jerarquía de valores con igualdad de probabilidad. La distribución uniforme se describe mediante cotas inferior y superior. La distribución uniforme igualmente se conoce como la distribución rectangular [19].

La distribución Uniforme es el modelo continuo más simple y pertenece al caso de una variable aleatoria que toma valores percibidos entre dos extremos  $a$  y  $b$ , de forma que todos los intervalos tienen una probabilidad equivalente, es decir, la función de densidad debe obtener el valor similar para todos los tantos dentro del intervalo  $(a, b)$  [6]. Algunos de los ejemplos reales son el tiempo que tarda un autobús en ir a un destino a otro y viceversa, conociendo sus intervalos, otro de los ejemplos puede ser el peso de una fruta [10].

#### 2.1.5. Distribución Gaussiana

La distribución normal es la distribución de probabilidad más significativa del Cálculo de probabilidades y de la Estadística. Fue descubierta, como proximidad de la distribución binomial, por Abraham De Moivre (1667-1754) y



divulgada en 1733 en su texto *The Doctrine of Chances*; En 1809, Carl Friedrich Gauss (1777-1855) divulgó un libro sobre el movimiento de los cuerpos celestes en el que asumía errores normales, por esta razón la distribución igualmente es distinguida como distribución Gaussiana [8].

La importancia de la distribución normal queda afianzada por ser la distribución final de numerosas variables aleatorias, discretas y continuas, como se señala a través de los teoremas centrales del límite. Los efectos de estos teoremas involucran la representación de la distribución normal en todos los campos de las ciencias empíricas: biología, medicina, psicología, física, economía, etc. En particular, diversas medidas de datos continuos en medicina y en biología (talla, presión arterial, etc.) se acercan a la distribución normal [8]. Como ejemplo del uso de esta distribución en problemas reales podemos calcular cuántos días se espera alcanzar una temperatura máxima teniendo como referencia un histórico de las temperaturas máximas alcanzadas en ese territorio, con las cuales se calculará la media y la desviación típica, otra de las aplicaciones puede ser para aplicar un examen donde se tienen los datos de la media y la desviación típica de los resultados [25].

La distribución normal queda completamente determinada mediante dos parámetros: la media ( $\mu$ ) y la desviación estándar o desviación típica ( $\sigma$ ). Su función de densidad es proporcionada respecto a la media y la desviación estándar nos exterioriza el mayor o menor grado de comienzo de la curva que, por su fachada, se suele llamar campana de Gauss. Esta distribución se expresa por  $N(\mu, \sigma)$ .

Campo de variación:

$$-\infty < x < \infty$$

Parámetros:

$$\mu: \text{media, } -\infty < \mu < \infty$$

$$\sigma: \text{desviación estándar, } \sigma > 0$$

### 2.1.6. Distribución Exponencial

Esta legislación de distribución describe técnicas en los que importa saber el tiempo hasta que sucede determinado evento; en particular, se utiliza para modelar lapsos de supervivencia. Un ejemplo es el lapso que tarda una partícula radiactiva en desintegrarse [8].

Esta distribución se consigue identificar como la distribución del tiempo entre sucesos consecutivos formados por un proceso de Poisson; por ejemplo, el tiempo de curación que sucede entre dos heridas graves de una persona. Otros ejemplos de la vida real del uso de esta distribución es la comprobación del tiempo de vida de un cierto tipo de marcapasos y el tiempo que transcurre una persona esperando a ser atendida en una cafetería [9].

La media de la distribución de Poisson,  $\lambda$ , que simboliza la tasa de ocurrencia del evento por unidad de tiempo, es el parámetro de la distribución exponencial [8].

Campo de variación:

$$0 < x < \infty$$

Parámetros:

$\lambda$ : tasa,  $\lambda > 0$

### 2.1.7. Bacterias E. Coli

E. Coli, es un ejemplar común de bacteria con una densidad de  $1 \mu\text{m}$  y alrededor de un largo de  $2\mu\text{m}$ , y bajo circunstancias apropiadas puede reproducirse en 20 minutos. Una E. Coli va variando sus movimientos entre nadar o girar. Estas bacterias tienen flagelos que trabajan como hélices que ceden a la bacteria moverse, si nadan, las hélices van en sentido contrario de las manecillas del reloj, si giran, los flagelos van en sentido de las manecillas del reloj.

Operaciones quimiotácticas:

- Si está dentro de un medio neutro de nutrientes (alterna nados y giros), realiza la indagación de nutrientes.
- Si está nadando en un área de nutrientes (o fuera de sustancias nocivas) y nada en esa área (las concentraciones de nutrientes incrementan y las sustancias nocivas reducen), busca ambientes cada vez más favorables.

- Si está nadando en un área de nutrientes (o arriba de sustancias nocivas), entonces busca y evita ambientes desfavorables.

De esta forma, las bacterias logran escalar colinas de nutrientes y evadir sustancias nocivas al mismo tiempo.

Las E. Coli segregan quimio-atrayentes que distinguen a través de sus sensores para comunicarse entre la población de bacterias y como recibidores de proteínas (un pequeño cambio en la concentración de nutrientes puede causar un significativo cambio en el comportamiento). Los sensores hacen un promedio de las agrupaciones detectadas en un lapso de tiempo [1].

### 2.1.8. Medidas de Estadísticas Básicas

La Estadística es una disciplina científica que se usa en base a técnicas matemáticas para obtener, ordenar y analizar un grupo de datos de los que se consiguen conclusiones distinguidas en base a los datos observados. Su principal objetivo es optimizar la perspicacia de los hechos a partir de la investigación disponible [22].

Las conclusiones se obtendrán a través de las medidas de estadísticas básicas: Mejor, Media, Peor y Desviación Estándar, con las cuales se presentarán los resultados del análisis de los datos.

#### **Media**

La media nos da una idea de alrededor de qué valor se producen los valores aleatorios de una variable [3].

#### **Desviación Estándar**

La desviación estándar establece la raíz cuadrada positiva de la varianza [20], mientras que la varianza es la propagación de los valores en torno a la media [3].

## Mejor y Peor

El valor conseguido nos brinda la indagación en relación al intervalo entre los valores límites en los que se observaron los datos [20].

### 2.1.9. Kruskal-Wallis Test

La prueba de Kruskal-Wallis, propuesta por William Kruskal y W. Allen Wallis en 1952, es un método no paramétrico para probar si las muestras se originan en la misma distribución. La hipótesis nula de la prueba de Kruskal-Wallis es que los rangos medios de los grupos son los mismos [28].

La prueba de Kruskal-Wallis es una alternativa no paramétrica a la prueba ANOVA de un factor para medidas independientes. Se basa en el orden de rango de los datos en lugar de cálculos que involucran medias y varianzas, y le permite evaluar las diferencias entre tres o más muestras independientes (tratamientos)[24].

## 2.2. Literatura relacionada

En las revisiones del estado del arte se encontraron algunas propuestas de algoritmos bio-inspirados haciendo uso de distintas distribuciones como gaussiana, gamma y exponencial para mejorar su rendimiento con la prevalencia de diversidad en la población. lo que conlleva una mejor distribución de la población en el espacio de búsqueda, en estos artículos se han encontrado propuestas como [12, 7, 23, 2, 26, 4], que hacen uso de distribución gaussiana para mejorar la diversidad de la población en algoritmos evolutivos y las utilizan en un operador de mutación. En la propuesta de [29], utilizan una mutación gaussiana para mejorar el rendimiento del Algoritmo BFOA con el objetivo de aumentar su diversidad poblacional y evitar la convergencia prematura.

A continuación se muestra una lista de trabajos relacionados en donde se utiliza las metaheurísticas con alguna distribución de probabilidad de las que se exploran en este trabajo.

## Generadores aleatorios no gaussianos en el Algoritmo de búsqueda de bacterias para la optimización multiobjetivo.

De los autores [12], se busca analizar los efectos de los motores estocásticos no gaussianos en el rendimiento de la metaheurística cuando se implementa en un problema de ingeniería del mundo real como es la optimización de objetivos múltiples del sistema de moldes de arena unidos por resina. La influencia de los tres tipos de motores estocásticos (la distribución Weibull, la distribución Gamma, la distribución gaussiana) en la optimización de objetivos múltiples del sistema de moldes de arena unidos por resina se ejecutan mediante la implementación de la técnica BFA (Algoritmo de Forrajeo de Bacterias) [12].

La técnica BFA equipada con el gaussiano, las distribuciones de Weibull y Gamma se denominan G-BFA, W-BFA y  $\gamma$ -BFA respectivamente, mientras que el BFA junto con el generador caótico se llama Ch-BFA. Para la aproximación de la frontera de Pareto, se obtuvieron 53 soluciones para varios pesos para cada una de las variantes de BFA empleadas en este trabajo. Está claro que, para este problema, el componente caótico en el motor estocástico no mejora el rendimiento de la metaheurística. El W-BFA supera al  $\gamma$ -BFA, G-BFA y Ch-BFA en un 32.857 %, 8.538 % y 22.113 % respectivamente en términos de la mejor solución individual. Todos los puntos de solución obtenidos utilizando los enfoques fueron factibles y no se produjeron violaciones de restricciones [12].

El  $\gamma$ -BFA no funciona tan bien como el W-BFA o el G-BFA convencional, aunque también está equipado con un motor estocástico no gaussiano. Estos resultados muestran que, aunque tener un motor estocástico no gaussiano puede ser ventajoso, es posible que la elección de la distribución no gaussiana empleada en el método de solución dependa del tipo de panorama de la solución. Debido a esta dependencia, la efectividad del enfoque computacional con un tipo particular de motor estocástico no gaussiano variaría según las características y el tipo del problema.

**Un método híbrido basado en una red neuronal y un método mejorado de adaptación ambiental que utiliza la mutación gaussiana controlada con parámetros reales para el pronóstico de carga a corto plazo.**

De [23] se propone una metodología híbrida, a saber, ANN-IEAMCGM-R, para el pronóstico de carga a corto plazo. ANN (Redes Neuronales Artificiales) está integrado con un Algoritmo evolutivo mejorado (IEAMCGM-R) para encontrar pesos de red óptimos. Este Algoritmo evolutivo se compone de un método de adaptación ambiental mejorado con parámetros reales (IEAM-R) y el método propuesto de Mutación Gaussiana Controlada (CGM) para traer una mayor diversidad dentro de la población, lo que resulta en una mayor convergencia de soluciones.

Se utilizaron seis métricas de evaluación comúnmente utilizadas las cuales son: error absoluto medio (MAE), error porcentual absoluto medio (MAPE), Error máximo porcentual absoluto medio diario (Daily Peak MAPE), cambio direccional (DC), coeficiente de correlación de Pearson ( $r$ ) y índice de acuerdo (IA). Se establece el tamaño de la población como 100, el número máximo de iteraciones como 10,000 y el número de neuronas ocultas como 10, el número de neuronas de entrada es igual al número de parámetros de entrada tomados en el estudio de caso y una neurona de salida. Los resultados de pronóstico de ANN-IEAMCGM-R, ANN-IEAMGM-R, ANN-IEAM-R, ANN-Jaya, ANN-BFGS, LR y GRNN se realizan mediante la prueba estadística de Friedman. Se usa una prueba de comparación múltiple no paramétrica como lo es la prueba de Friedman y tiene como objetivo detectar diferencias significativas entre los resultados de dos o más modelos de pronóstico. La hipótesis nula para la prueba de Friedman se expresa como: todos los modelos de pronóstico generan los mismos errores de pronóstico y la hipótesis alternativa es la negación de la hipótesis nula [23].

ANN-IEAMCGM-R logra una mayor precisión de pronóstico. La mutación ofrece una gran diversidad que conduce a una mejor solución óptima que IEAM-R. La metodología híbrida propuesta supera a los otros modelos al generar un error de pronóstico mínimo al mantener su estabilidad en diferentes estaciones. Todas las métricas de error efectivas revelan que la metodología híbrida ANN-IEAMCGM-R es superior no solo a los modelos híbridos evolu-

tivos, sino también a otros modelos de predicción de última generación. Los resultados también muestran que ANN-IEAMCGM-R se desempeñó mejor que ANN-IEAMGM-R, lo que verifica que CGM brinda una mejor diversidad que GM y al agregar un valor gaussiano aleatorio a  $Pbest$  genera resultados superiores que agregar más de un valor de distribución gaussiana a candidatos elegidos al azar.

La propuesta de mutación gaussiana controlada aumenta la capacidad de exploración del algoritmo y también restringe para mantenerse atascado en los óptimos locales.

### **Diversidad poblacional de Algoritmos evolutivos no elitistas en la fase de exploración.**

De [2]. Trata de la discusión de la diversidad genética de las poblaciones codificadas procesadas por un Algoritmo Evolutivo (EA). La diversidad se expresa como una varianza o una matriz de covarianza de los individuos contenidos en la población, en casos de una o varias dimensiones, respectivamente. Las fórmulas se derivan para predecir la diversidad para la selección proporcional de aptitud física, torneo y truncamiento, con y sin cruce aritmético y con mutación gaussiana.

La validación experimental del caso multidimensional muestra que la precisión de la predicción es satisfactoria en un amplio espectro de configuraciones de parámetros de EA. Una conclusión de los experimentos es que la precisión de la predicción es alta en una amplia gama de configuraciones de parámetros. La precisión es relativamente sensible a la influencia del crossover aritmético en la precisión de la predicción para el tipo de selección y su parámetro, y para el tamaño de la población.

Se introdujeron fórmulas analíticas que describen la diversidad genética cuantificada como la matriz de covarianza de la población esperada. Analizaron el modelo de adecuación del ruido en el que los valores de la función objetivo son una variable aleatoria, cuya distribución no depende de la ubicación de los individuos. Se demostró que la matriz de covarianza pronosticada es linealmente proporcional a la matriz de covarianza utilizada para mutar el

número de padre efectivo y puede interpretarse como un tamaño de población ajustado para pesos de selección no uniformes [2].

### **Un Algoritmo evolutivo inspirado en la cuántica que utiliza la cuantización basada en la distribución gaussiana.**

De [26], amplía el alcance de su proyecto de investigación mediante la aplicación de principios de computación cuántica, en particular el principio de superposición cuántica que propone un nuevo enfoque de cuantificación. Como resultado se propone un Algoritmo de coevolución competitiva de inspiración cuántica (QCCEA). A diferencia de sus predecesores y los algoritmos tradicionales de inspiración cuántica, el QCCEA propuesto utiliza un enfoque novedoso para la cuantización utilizando la distribución gaussiana que altera el procedimiento de selección para identificar la aptitud óptima.

Se realizaron experimentos con 20 problemas de optimización numérica de referencia, así como un problema de laberinto combinatorio. Los resultados del experimento muestran que la técnica de cuantificación propuesta ha mejorado significativamente el rendimiento y la precisión de los algoritmos tradicionales no cuantificados para ambos tipos de problemas. Dentro de las 20 funciones 15 de estas funciones son de minimización, es decir, que 5 son de maximización, las funciones experimentadas son funciones multimodales. Las funciones son bien conocidas por su complejidad ya que el número de mínimos locales en ellas aumenta exponencialmente con la dimensión del problema, Las funciones son bien conocidas por su complejidad ya que el número de mínimos locales en ellas aumenta exponencialmente con la dimensión del problema. Los resultados obtenidos se evaluaron utilizando la prueba T para la verificación estadística para probar la validez de los experimentos [26].

Todas las funciones de referencia se definen en el mismo dominio  $[-100, 100]^D$ , con la dimensión del problema  $D = 10$ . El experimento se realiza durante 3000 generaciones y cada algoritmo se prueba de forma independiente en cada función 25 veces. Es evidente que QCCEA tiende a desempeñarse mejor que CCEA, independientemente de la forma, las propiedades y el número de óptimos locales de la prueba en funciones de referencia. En QCCEA, la solución candidata se representa a través de una combinación de un conjunto de



puntos de solución que se describen conjuntamente a través de distribuciones normales.

Los resultados obtenidos del experimento, QCCEA superó a CCEA en 16 de las 20 funciones de prueba. La velocidad de búsqueda de QCCEA también es notablemente más rápida, ya que, en comparación con CCEA, QCCEA identifica soluciones casi óptimas en un menor número de generaciones.

### **Una eficaz optimización de búsqueda de bacterias basada en la optimización sobre conjugación y nuevas estrategias de tamaño de paso.**

Según [4] la optimización de forrajeo de bacterias (BFO) es un ejemplo de un algoritmo metaheurístico activo que se ha aplicado ampliamente al mundo real. A pesar de la excelente funcionalidad informática, los algoritmos apenas pueden evitar la convergencia prematura inducida por atrapamiento fácil en óptimos locales. Para mejorar la función informática, de acuerdo con el Algoritmo de BFO, este artículo presenta un BFO mejorado algoritmo que emplea una nueva estrategia de evolución de longitud de paso. Además, el Algoritmo BFO mejorado adopta la estrategia de vuelo Lévy propuesto en LPBFO y la estrategia de conjugación propuesta en BFO-CC. Al combinar las tres estrategias asociadas, el Algoritmo de BFO de tamaño de paso de novedoso conjugado propuesto (CNS-BFO) logra un equilibrio excepcional entre explotación y exploración, mitigando efectivamente el problema de la prematuridad convergencia en Algoritmo BFO.

Las modificaciones de estructura de BFO también son propuestas para reducir el tiempo computacional, los investigadores han rediseñado la construcción del Algoritmo BFO, basado en rediseño de estructura (SRBFO) y IMBFOA. Recientemente, su estructura co-evolutiva rediseñada lograr un satisfactorio rendimiento y ahorra activamente los esfuerzos computacionales del Algoritmo BFO.

Para demostrar la eficacia del proyecto CNS-BFO propuesto, se compara con otros tipos de algoritmos heurísticos: GA y PSO. Además, como variante del Algoritmo BFO, se compara con BFO estándar y varias variantes populares

de BFO: BFO-LDC, BFO-NDC y BCO. Para hacer la comparación justa, algoritmos son asignados con las mismas posiciones de inicialización y los parámetros reparados se adoptan para los algoritmos de comparación.

Los resultados indican que el Algoritmo propuesto CNS-BFO tiene una mejora significativa en el rendimiento del Algoritmo BFO evitando la convergencia prematura y acelerando la convergencia. Como resultado, el Algoritmo CNS-BFO propuesto contribuirá a la perfección del Algoritmo de optimización de forrajeo bacteria e incluso otros algoritmos metaheurísticos [4].

## **Una optimización mejorada de la búsqueda de bacterias y su aplicación para entrenar la máquina de aprendizaje**

En [29] se menciona que el Algoritmo de optimización de bacterias (BFO) es un Algoritmo de inteligencia colectiva ampliamente utilizado en varios problemas de optimización. Sin embargo, BFO sufre de múltiples inconvenientes, incluida la velocidad de convergencia lenta, la incapacidad de saltar de los óptimos locales y la longitud de paso fija. Se propone un BFO mejorado con longitud de paso de quimiotaxis caótica, mutación gaussiana y búsqueda local caótica (CCGBFO) para superar la debilidad existente del BFO original.

Primero, se utiliza una operación de longitud de paso de quimiotaxis caótica para producir la longitud de paso de quimiotaxis adaptativa, por lo tanto, la velocidad de convergencia y la precisión del algoritmo se pueden mejorar a través de la longitud del paso de quimiotaxis variable adaptativa. Luego, combinando la posición óptima en la bacteria actual con la operación de mutación gaussiana para aprovechar al máximo la información de la posición óptima, es decir, la información de la posición óptima en la población actual se utiliza por completo, aumentando así la diversidad de la población y evitando los óptimos locales en favor de la búsqueda global. Finalmente, se introduce una búsqueda local caótica con una operación de estrategia de reducción en el paso de quimiotaxis para garantizar que el algoritmo propuesto pueda explorar un gran espacio de búsqueda en la etapa inicial, evitando así caer en los óptimos locales, y que el rango de búsqueda pueda disminuir en la etapa posterior, lo que hace que el algoritmo propuesto converja en una solución óptima.

La distribución gaussiana (es decir, distribución normal) juega un papel decisivo en la estadística matemática y su teoría de la probabilidad. La mutación gaussiana se conoce como la adición de un vector aleatorio *obeying* y se aplica de la siguiente manera:

Paso 1: aplique la mutación gaussiana a la posición óptima *gbest* en la población bacteriana actual para generar la mutación posición *gbestGao*. La expresión se muestra en la ecuación siguiente, donde *Gauss*(0, 1) es la distribución normal estándar:

$$gbestGao = gbest * (1 + Gauss(0, 1)) \quad (2.1)$$

Paso 2: si la función de condición física de la posición mutada *gbestGao* es mejor que *gbest* (representa la función óptima de condición física en la población bacteriana actual), se reemplaza *gbest* con la aptitud de *gbestGao*, y *gbest* se actualiza con *gbestGao*. Aumenta la perturbación aleatoria de la distribución gaussiana término *gbest* \* *Gauss* (0 , 1) sobre la base de *gbest* , que completa uso de la alteración de la información de la posición óptima *gbest* en la población bacteriana actual.

Por lo tanto, la ecuación no solo puede permitir que las bacterias salten del óptimo local y converjan al global óptima, pero también mejora la velocidad de convergencia.

El rendimiento del algoritmo propuesto CCGBFO fue evaluado en 23 funciones de referencia clásicas, que son, divididas en tres grupos: unimodal, multimodal o de dimensión fija multimodal. Se comparó con 7 Algoritmos de optimización metaheurísticos conocidos incluyendo PSO, DE, Algoritmo de murciélago (BA), FA, ABC, Algoritmo de polinización de flores(FPA) y lobo gris Optimizador (GWO). Además, CCGBFO también se comparó con el BFO original y tres meta- desarrollados recientemente: MFO, Algoritmo de libélula (DA), y Algoritmo de optimización de saltamontes (GOA). Los parámetros de los algoritmos mencionados anteriormente se establecieron de acuerdo con el valor de parámetros de sus documentos originales [29].

Dados los múltiples inconvenientes sufridos por BFO, incluida la lentitud velocidad de convergencia, baja precisión de búsqueda e incapacidad para saltar

de óptima local, los resultados experimentales muestran que el CCGBFO propuesto supera significativamente al BFO original en términos de velocidad de convergencia y precisión de la solución indicando que la combinación de estas tres estrategias puede lograr un mejor equilibrio relación entre exploración y explotación. Además, CCGBFO se utilizó para resolver el problema de selección de parámetros de KELM y así obtener resultados de predicción muy precisos en dos problemas del mundo real.

### 2.3. Marco tecnológico

TS-MBFOA fue adaptado y ejecutado para resolver los 6 problemas del benchmark en un equipo de cómputo con Memoria RAM de 8.0GB, procesador de 2.4GHZ y un sistema operativo Windows de 64bits. El algoritmo fue programado en lenguaje Java en el software de desarrollo NetBeans IDE 8.2. que es de licencia gratuita.

## Capítulo 3

# Modelo de las distribuciones poblacionales y el reinicio del TS-MBFOA

### 3.1. Distribución Uniforme

La distribución uniforme es una distribución continua que forma un rango de valores con igual probabilidad y se define mediante cotas inferior y superior  $[a, b]$  [19]. Si  $x \sim u(a, b)$ , su función de distribución viene dada por:

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x - a}{b - a} & a \leq x < b \\ 1 & x \geq b \end{cases} \quad (3.1)$$

Aplicando el método de inversión de la función de distribución se consigue el siguiente esquema:

1. Generar un número aleatorio  $u$
2. Tomar  $x = a + u(b - a)$

### 3.2. Distribución Gaussiana

La distribución normal permanece íntegramente definida a través de dos parámetros: la media ( $\mu$ ) y la desviación estándar o desviación típica ( $\sigma$ ). Su

función de densidad es proporcionada respecto a la media y la desviación estándar nos revela el mayor o menor grado de apertura de la curva que, por su aspecto, se suele llamar campana de Gauss [8]. Esta distribución se expresa por  $N(\mu, \sigma)$ .

La distribución  $N(\mu, \sigma)$  se puede relacionar con la distribución  $N(0, 1)$ , mediante el sucesivo proceso al que se denomina tipificación o estandarización:

$$X \sim N(\mu, \sigma) \Rightarrow Z = \frac{X - \mu}{\sigma} \quad (3.2)$$

A la distribución  $N(0, 1)$  se le denomina Normal Estándar.

En diversos lenguajes de programación como Java, existen funciones programadas que permiten generar un número aleatorio con distribución normal con media y desviación estándar de cero. El siguiente esquema permite generar un número aleatorio con distribución normal con media y desviación estándar definida por el usuario final:

1. Generar un número aleatorio  $u$  con distribución  $N(0, 1)$
2. Definir los valores de  $\sigma$  y  $\mu$  de un conjunto de valores
3. Tomar  $x = u * \sigma + \mu$

### 3.3. Distribución Exponencial

Esta distribución se puede caracterizar como la distribución del período entre eventos consecutivos formados por un proceso de Poisson; por ejemplo, el tiempo que tarda en sanar dos heridas graves sufridas por una persona [8]. La media de la distribución de Poisson,  $\lambda$ , que simboliza la tasa de ocurrencia del evento por unidad de tiempo, es el parámetro de la distribución exponencial, y su contraria es el valor medio de la distribución.

Para generar un valor de una variable aleatoria  $X$  con distribución exponencial y parámetro  $\lambda$  a partir de un valor de una variable aleatoria  $u = u(0, 1)$  se utiliza el método de la transformada inversa para obtener la siguiente ecuación:

$$X = -\frac{1}{\lambda} \ln(1 - u) \quad (3.3)$$

Utilizando el hecho de que si  $u = u(0, 1)$  entonces  $1 - u = u(0, 1)$ . El esquema para generar un número aleatorio con distribución exponencial es:

1. Generar un número aleatorio  $u$
2. Tomar  $x = -\frac{1}{\lambda} \ln(u)$

### 3.4. TS-MBFOA con reinicio de población

El algoritmo TS-MBFOA es adaptado para reiniciar la mitad de la población, en este caso las peores bacterias, con una población aleatoria con distribución gaussiana o exponencial. El reinicio es único durante las generaciones del algoritmo y se lleva a cabo cuando la mitad de las bacterias de la población es factible, es decir, no se violan restricciones. Esto para garantizar que la nueva población de bacterias esté cerca o dentro de la región factible del problema y evitar que se generen bacterias fuera del espacio de búsqueda determinado por el rango de las variables del problema a resolver. Inicialmente TS-MBFOA la población de bacterias es totalmente generada con distribución uniforme.

De igual manera, se incluye un validador y corrector de números aleatorios con el objetivo de generar números dentro del espacio de búsqueda del problema. Después de generar cada número aleatorio con distribuciones gaussiana o exponencial. Para ello, cada número es validado verificando que se encuentre dentro del rango mínimo y máximo de cada variable de diseño del problema a resolver. Cuando un número aleatorio viola los rangos permitidos, un nuevo número aleatorio es generado con el mismo tipo de distribución usada.

Para el caso de la distribución uniforme, una bacteria es generada con la Ecuación 3.4:

$$\theta^i(j, G) = L_i x_i + u * (U_i x_i - L_i x_i) \quad (3.4)$$

donde  $L_i x_i$  es el límite inferior de la variable  $x_i$ ,  $U_i x_i$  es el límite superior de la variable  $x_i$  y  $u$  es un número aleatorio entre  $(0,1)$ .

Una bacteria con distribución gaussiana o normal es generada con la Ecuación 3.5

$$\theta^i(j, G) = u * \sigma(x_i) + \mu(x_i) \quad (3.5)$$

donde  $u$  es un número aleatorio con distribución  $N(0, 1)$ , en lenguaje Java corresponde a la función `nextGaussian()`.  $\mu$  es la media de la mitad de la población de bacterias factibles para cada variable  $x_i$  y  $\sigma$  es la desviación estándar de la mitad de la población de bacterias factibles para cada variable  $x_i$ .

Una bacteria con distribución exponencial es generada con la Ecuación 3.6

$$\theta^i(j, G) = -\frac{1}{\lambda} \ln(u) \quad (3.6)$$

donde  $u$  es un número aleatorio entre  $[0,1]$  y  $\lambda$  es la media de la mitad de la población de bacterias factibles para cada variable  $x_i$ .

La media y desviación estándar utilizada en estas distribuciones hacen uso de la información obtenida por las bacterias hasta la generación donde la mitad de la población de bacterias es factible.



# Capítulo 4

## Experimentos y Resultados

En esta sección se describen los experimentos realizados y los resultados obtenidos del Algoritmo de optimización de doble nado modificado basado en el forrajeo de bacterias (TS-MBFOA) con el fin de analizar el rendimiento del mismo utilizando dos distribuciones de población diferentes a la uniforme (la distribución gaussiana y distribución exponencial) usando una comparativa de los mejores resultados obtenidos en diferentes problemas de prueba. Este capítulo se divide en dos secciones: en la primera describe los seis problemas de optimización a resolver; en la segunda sección se detalla la calibración de parámetros, las ejecuciones realizadas y los resultados obtenidos.

### 4.1. Problemas

#### 4.1.1. Esfera

El problema de la esfera busca en este proyecto graficar su distribución en el área de búsqueda para observar el comportamiento de las funciones de distribución de probabilidad (Gaussiana, Exponencial y Uniforme). Cuenta con dos variables: altura  $x_1$  y radio  $x_2$  que es un problema sin restricciones y se representa como:

$$x_1^2 + x_2^2 \tag{4.1}$$

Donde:

$$\begin{aligned} -5 &\leq x_1 \leq 5 \\ -5 &\leq x_2 \leq 5 \end{aligned} \quad (4.2)$$

#### 4.1.2. P1: Resorte de tensión/compresión

Busca minimizar el peso de un resorte sujeto a restricciones de desviación mínima, tensión de corte, frecuencia de oleada, límites sobre el diámetro exterior, esto sobre variables de diseño. El diseño de la función para ser procesada en el Algoritmo TS-MBFOA cuenta con las siguientes variables de diseño [17]:

- Diámetro del cable  $d(x_1)$
- Diámetro del rollo  $D(x_2)$
- el número de rollos involucrados  $N(x_3)$

Formalmente, el problema puede expresarse de la siguiente manera: Minimizar:

$$(x_3 + 2)x_2x_1^2 \quad (4.3)$$

Sujeto a:

$$\begin{aligned} g_1(X) &= 1 - ((x_2^3x_3)/(71785x_1^4)) \leq 0 \\ g_2(X) &= ((4x_2^2 - x_1x_2)/12566(x_2x_1^3 - x_1^4)) + (1/5108x_1^2) - 1 \leq 0 \\ g_3(X) &= 1 - (140.45x_1/x_2^2x_3) \leq 0 \\ g_4(X) &= (x_2 + x_1/1.5) - 1 \leq 0 \end{aligned} \quad (4.4)$$

Donde:

$$\begin{aligned} 0.05 &\leq x_1 \leq 2 \\ 0.25 &\leq x_2 \leq 1.3 \\ 2 &\leq x_3 \leq 15 \end{aligned} \quad (4.5)$$

### 4.1.3. P2: Recipiente de presión

Se busca minimizar el costo total de un recipiente de presión, incluyendo el costo del material, forma y soldadura. Existen cuatro variables de diseño:

- $T_s$  (grosor de la tapa ( $x_1$ ))
- $T_h$  (grosor de la cabeza ( $x_2$ ))
- R (radio interior ( $x_3$ ))
- L (longitud de la sección cilíndrica del recipiente)
- no incluyendo la cabeza ( $x_4$ )

$T_s$  y  $T_h$  son enteros múltiples de 0.0625 pulgadas, los cuales son capas disponibles que hacen rodar platos de acero y R y L son continuos [17]. Formalmente, el problema puede expresarse de la siguiente manera: Minimizar:

$$0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (4.6)$$

Sujeto a:

$$\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \leq 0 \\ g_2(X) &= -x_2 + 0.00954x_3 \leq 0 \\ g_3(X) &= \pi x_3^2 x_4 - 4/3 \pi x_3^3 + 1296000 \leq 0 \\ g_4(X) &= x_4 - 240 \leq 0 \end{aligned} \quad (4.7)$$

Donde:

$$\begin{aligned} 1 &\leq x_1 \leq 99 \\ 1 &\leq x_2 \leq 99 \\ 10 &\leq x_3 \leq 200 \\ 10 &\leq x_4 \leq 200 \end{aligned} \quad (4.8)$$

### 4.1.4. P3: Viga soldada

El objetivo de este problema es encontrar el costo mínimo de fabricación de la viga soldada sujeta a restricciones como la tensión de corte ( $\tau$ ), doblar la tensión de la viga ( $\sigma$ ), abrochar la carga sobre la barra ( $P_c$ ), desviación final

de la viga ( $\delta$ ) y restricciones de equipo [17]. Formalmente, el problema puede expresarse de la siguiente manera: Minimizar:

$$1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (4.9)$$

Sujeto a:

$$\begin{aligned} g_1(X) &= \tau(X) - \tau_{max} \leq 0 \\ g_2(X) &= \sigma(X) - \sigma_{max} \leq 0 \\ g_3(X) &= x_1 - x_4 \leq 0 \\ g_4(X) &= 0.125 - x_1 \leq 0 \\ g_5(X) &= \delta(X) - \delta_{max} \leq 0 \\ g_6(X) &= P - P_c \leq 0 \\ g_7(X) &= 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \end{aligned} \quad (4.10)$$

Donde:

$$\begin{aligned} 0.1 &\leq x_1 \leq 2 \\ 0.1 &\leq x_2 \leq 10 \\ 0.1 &\leq x_3 \leq 10 \\ 0.1 &\leq x_4 \leq 2 \end{aligned} \quad (4.11)$$

$$\begin{aligned}\tau(X) &= \sqrt{(\tau)^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2} \\ \tau_1 &= \frac{P}{\sqrt{2}x_1x_2} \\ \tau_2 &= \frac{MR}{J} \\ M &= P\left(L + \frac{x_2}{2}\right) \\ J(X) &= 2\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\end{aligned}\tag{4.12}$$

$$\begin{aligned}R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\ \sigma(X) &= \frac{6PL}{x_4x_3^2} \\ \delta(X) &= \frac{6PL^3}{Ex_3^3x_4} \\ P_c(X) &= \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)\end{aligned}$$

$$\begin{aligned}G &= 12x10^6psi, E = 30x10^6psi, P = 6000lb, L = 14in, \\ \tau_{max} &= 13600, \sigma_{max} = 30000, \delta_{max} = 0.25\end{aligned}\tag{4.13}$$

#### 4.1.5. P4: Síntesis de procesos MINLP

Minimizar [11]:

$$(y_1-1)^2+(y_2-2)^2+(y_3-1)^2-\log(y_4+1)+(x_1-1)^2+(x_2-2)^2+(x_3-3)^2\tag{4.14}$$

Sujeto a:

$$\begin{aligned}
 g_1(X) &= y_1 + y_2 + y_3 + x_1 + x_2 + x_3 \leq 5 \\
 g_2(X) &= y_3^2 + x_1^2 + x_2^2 + x_3^2 \leq 5.5 \\
 g_3(X) &= y_1 + x_1 \leq 1.2 \\
 g_4(X) &= y_2 + x_2 \leq 1.8 \\
 g_5(X) &= y_3 + x_3 \leq 2.5 \\
 g_6(X) &= y_4 + x_1 \leq 1.2 \\
 g_7(X) &= y_2^2 + x_2^2 \leq 1.64 \\
 g_8(X) &= y_3^2 + x_3^2 \leq 4.25 \\
 g_9(X) &= y_2^2 + x_3^2 \leq 4.64
 \end{aligned} \tag{4.15}$$

Donde:

$$\begin{aligned}
 0 \leq x_i &\leq (1.2, 1.8, 2.5) \quad i = 1, 2, 3 \\
 y_i &= 0, 1 \quad i = 1, 2, 3, 4.
 \end{aligned} \tag{4.16}$$

#### 4.1.6. P5: Diseño de una viga de concreto reforzada

Minimizar [16]:

$$29.4x_1 + 18x_2 \tag{4.17}$$

Sujeto a:

$$g_1(X) = x_1 - 0.2458x_1^2/x_2 \leq 6 \tag{4.18}$$

Donde:

$$\begin{aligned}
 0 \leq x_1 &\leq 115.8 \\
 0.00001 \leq x_2 &\leq 30
 \end{aligned} \tag{4.19}$$

#### 4.1.7. P6: Un problema cuadrático restringido

Minimizar [27]:

$$x_1^4 - 14x_1^2 + 24x_1 - x_2^2 \tag{4.20}$$

Sujeto a:

$$\begin{aligned} g_1(X) &= -x_1 + x_2 - 8 \leq 0 \\ g_2(X) &= x_2 - x_1^2 - 2x_1 + 2 \leq 0 \end{aligned} \quad (4.21)$$

Donde:

$$\begin{aligned} -8 &\leq x_1 \leq 10 \\ 0 &\leq x_2 \leq 10 \end{aligned} \quad (4.22)$$

## 4.2. Experimentos

### 4.2.1. Parámetros

Los parámetros utilizados para resolver cada uno de los problemas ya mencionados son:

- bacterias ( $S_b$ ) = 50
- tamaño de paso ( $R$ ) = 0.0005
- factor de escalamiento ( $\beta$ ) = 1.8
- ciclo quimiotáxico ( $N_c$ ) = 12
- bacterias a reproducir ( $S_r$ ) = 1
- frecuencia de reproducción ( $RepCycle$ ) = 100
- número de evaluaciones = 15,000

Los valores para cada parámetro fueron calibrados en un conjunto de 50 ejecuciones independientes con cinco combinaciones de valores diferentes.

En cada problema se realizaron 30 ejecuciones independientes para cada una de las distribuciones analizadas y se obtuvieron los resultados de las estadísticas básicas (mejor, media, desviación estándar, peor) los resultados obtenidos se comparan entre sí para analizar cuál de las distribuciones permite un mejor rendimiento del algoritmo sobre cada uno de los problemas de prueba.

### 4.2.2. Resultados

Como primer experimento se probó el Algoritmo TS-MBFOA con el problema conocido como la esfera 4.1, esto se realizó con el objetivo de observar de manera gráfica la distribución de la población de bacterias con las diferentes distribuciones, cabe mencionar que el Algoritmo TS-MBFOA inicia con una población con distribución uniforme.

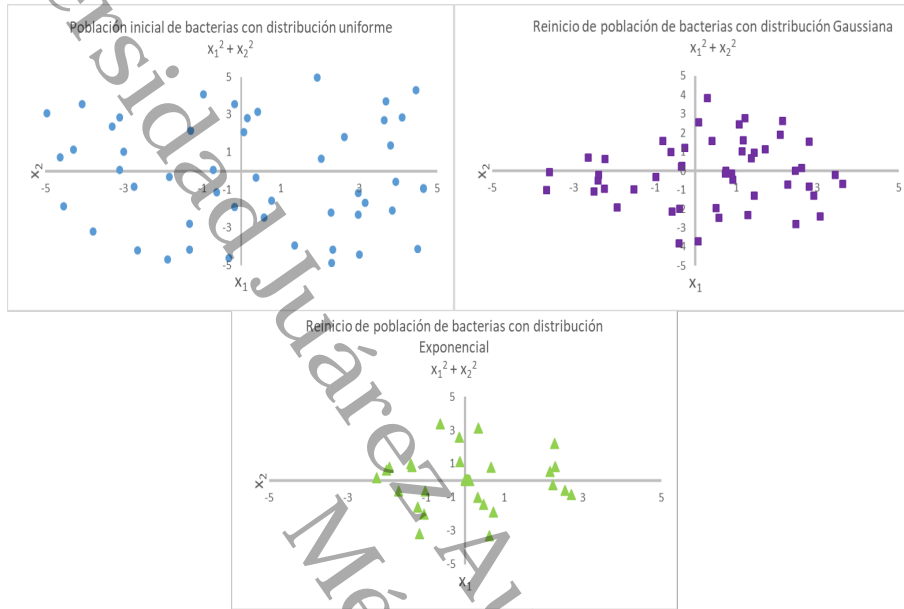
En el problema de la esfera es posible graficar sus variables debido a que solo cuenta con dos dimensiones. Este problema no tiene restricciones de igualdad o desigualdad, por lo tanto, el reinicio de la población con distribución gaussiana y exponencial es llevada a cabo en la generación 10 del algoritmo para efectos de visualizar como quedan dispersas las bacterias en el espacio de búsqueda.

Como se puede observar en la Figura 4.1, la población de bacterias con distribución exponencial se encuentra más agrupada hacia el origen, cabe mencionar que el valor óptimo de la función de la esfera se encuentra en dicho lugar, cuando  $x_1$  y  $x_2$  toman el valor de cero. La población de bacterias con distribución uniforme es la que abarca más lugares en el espacio de búsqueda, lo cual permite una mayor diversidad en la población, sin embargo, esto puede ralentizar al algoritmo a ubicar las zonas prometedoras donde se encuentre el óptimo global del problema. Finalmente, la población de bacterias con distribución gaussiana hace el efecto de campana, generando pocas bacterias en los límites del espacio de búsqueda y concentra a muchas en la parte central de los límites de ambas variables de diseño.

En la Tabla 4.1 se presenta a la mejor bacteria de la población al finalizar la ejecución del Algoritmo TS-MBFOA con diferentes distribuciones, donde el Algoritmo TS-MBFOA con distribución gaussiana obtuvo el mejor resultado.

Como segundo experimento se utilizó el Algoritmo TS-MBFOA para resolver los seis problemas en 30 ejecuciones independientes con cada una de las distribuciones. En la Tabla 4.2, se presentan los resultados estadísticos de las mejores soluciones encontrados en los problemas de resorte de tensión/compresión, el recipiente de presión, la viga soldada, síntesis de procesos MINLP, diseño de una viga de concreto reforzada y un problema cuadrático restringido respectivamente. Donde los mejores resultados se encuentran en





**Figura 4.1.** Población de bacterias con distribución uniforme, gaussiana y exponencial en el problema de la esfera

**Tabla 4.1.** Mejor resultado de TS-MBFOA con diferentes distribuciones

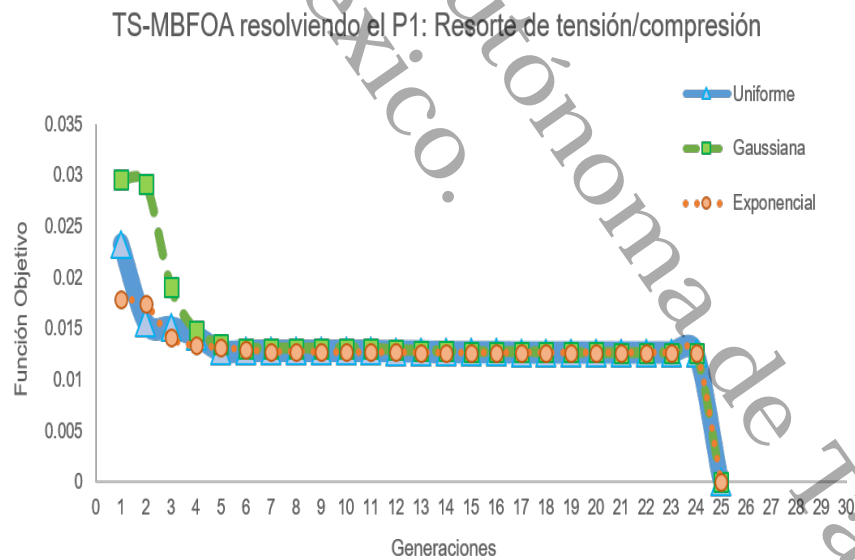
TS-MBFOA con distribución	$x_1$	$x_2$	$f(\vec{x}) = x_1^2 + x_2^2$
Uniforme	1.81824E-12	-3.86054E-12	1.82097E-23
Gaussiano	<b>4.02235E-15</b>	<b>-1.85888E-15</b>	<b>1.96347E-29</b>
Exponencial	3.80961E-14	4.19444E-14	3.21064E-27

letras negritas.

Posteriormente en la Tabla 4.3 se presentan los resultados de la prueba no paramétrica de Kruskal-Wallis Test la cual fue aplicada con un 95 % de certeza sobre el conjunto de las 30 mejores soluciones encontradas por cada versión del Algoritmo TS-MBFOA a cada uno de los problemas, no obstante, el problema numero 6 obtiene los mismos resultados, por lo cual el resultado es 0. En los otros 5 problemas los resultados indican que no hay una diferencia significativa al comparar las tres versiones del Algoritmo TS-MBFOA.

### 4.2.3. Gráficas de Convergencia

Las gráficas de convergencia que se presentan en las Figuras 4.2, 4.3 4.4, 4.5, 4.6, 4.7, son generadas con los datos correspondientes a la ejecución número 15, la cual es la mediana de las 30 ejecuciones independientes realizadas.



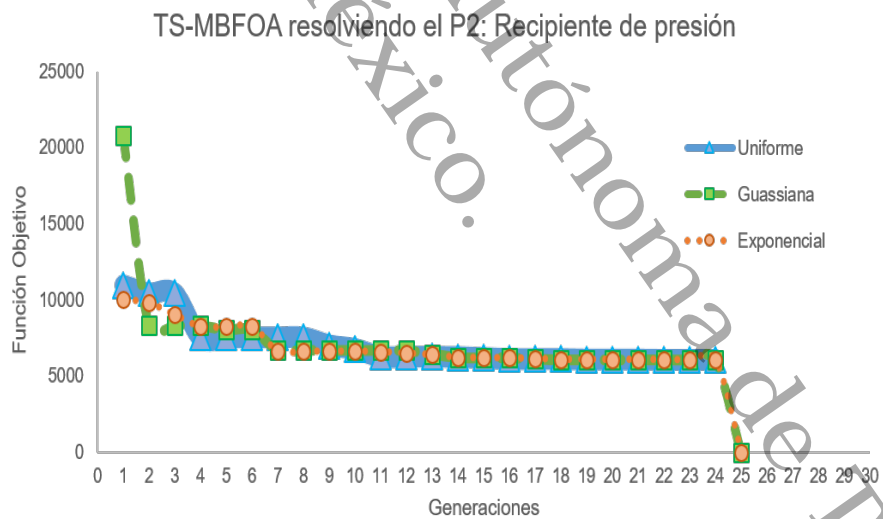
**Figura 4.2.** Gráfica de convergencia del P1: Resorte de tensión/compresión con distribución uniforme, gaussiana y exponencial

**Tabla 4.2.** Estadísticas básicas de las 30 ejecuciones independientes de TS-MBFOA con diferentes distribuciones

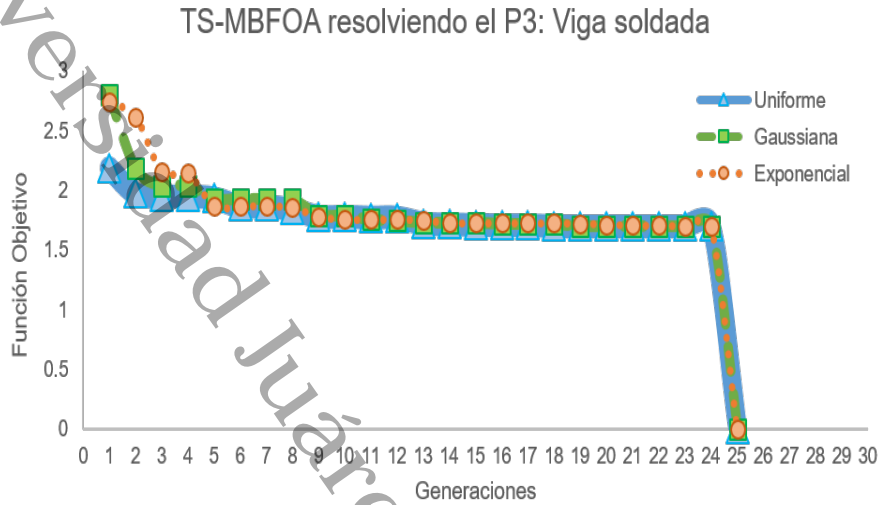
Criterios	TS-MBFOA uniforme	TS-MBFOA gaussiana	TS-MBFOA exponencial
P1: Resorte de tensión/compresión			
Mejor	<b>0.012665672</b>	0.012665852	0.012665788
Media	0.012670502	0.012670288	<b>0.012669101</b>
Peor	0.012682834	0.012692327	<b>0.012680927</b>
Desv.Est.	4.25232E-06	5.1684E-06	<b>3.8474E-06</b>
P2: Recipiente de presión			
Mejor	6060.45664	<b>6059.91253</b>	6060.1399
Media	<b>6066.38096</b>	6068.32289	6070.39165
Peor	<b>6083.76512</b>	6085.95015	6125.74094
Desv.Est.	<b>6.26708704</b>	8.20355469	13.2921831
P3: Viga soldada			
Mejor	1.69688124	1.6966019	<b>1.69606302</b>
Media	1.69967792	<b>1.69945779</b>	1.70012672
Peor	1.7093649	<b>1.70664196</b>	1.71531475
Desv.Est.	0.0024805	<b>0.00242742</b>	0.00417448
P4: Síntesis de procesos MINLP			
Mejor	4.38087167	<b>4.37341746</b>	4.38082053
Media	4.39756072	<b>4.39496775</b>	4.39708882
Peor	4.42905279	4.42229274	<b>4.41677386</b>
Desv.Est.	0.01236893	0.01037455	<b>0.00992847</b>
P5: Diseño de una viga de concreto reforzada			
Mejor	376.291932	376.291932	376.291932
Media	376.291933	376.291933	376.291933
Peor	376.291936	<b>376.291935</b>	<b>376.291935</b>
Desv.Est.	1.0997E-06	5.9623E-07	<b>4.6927E-07</b>
P6: Un problema cuadrático restringido			
Mejor	-118.70486	-118.70486	-118.70486
Media	-118.70486	-118.70486	-118.70486
Peor	-118.70486	-118.70486	-118.70486
Desv.Est.	<b>2.9559E-09</b>	3.4806E-09	4.1032E-09

**Tabla 4.3.** Resultado de la prueba no paramétrica Kruskal-Wallis Test de las 30 mejores soluciones encontradas en TS-MBFOA con las tres distribuciones

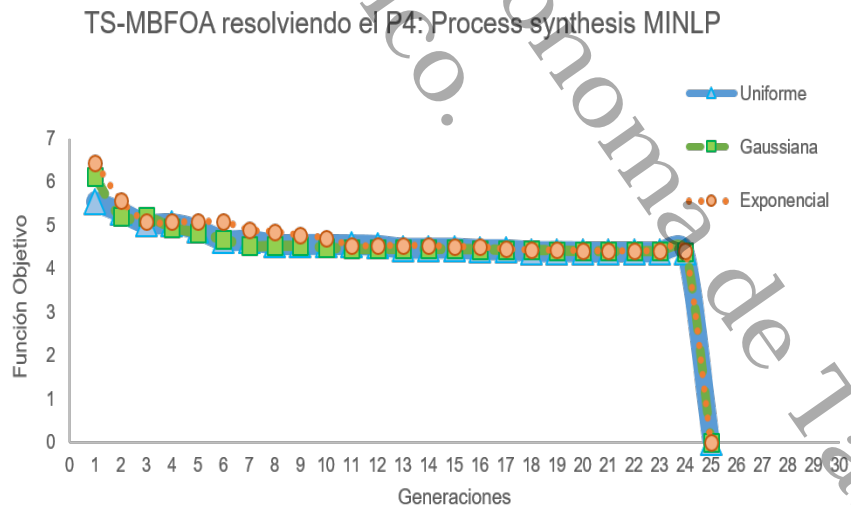
<b>Problema</b>	P1: Resorte de tensión/compresión	P2: Recipiente de presión	P3: Viga soldada
<b>P- value</b>	0.29457	0.74873	0.8352
<b>Problema</b>	P4: Síntesis de procesos MINLP	P5: Diseño de una viga de concreto reforzada	P6: Un problema cuadrático restringido
<b>P- value</b>	0.7954	0.39008	0



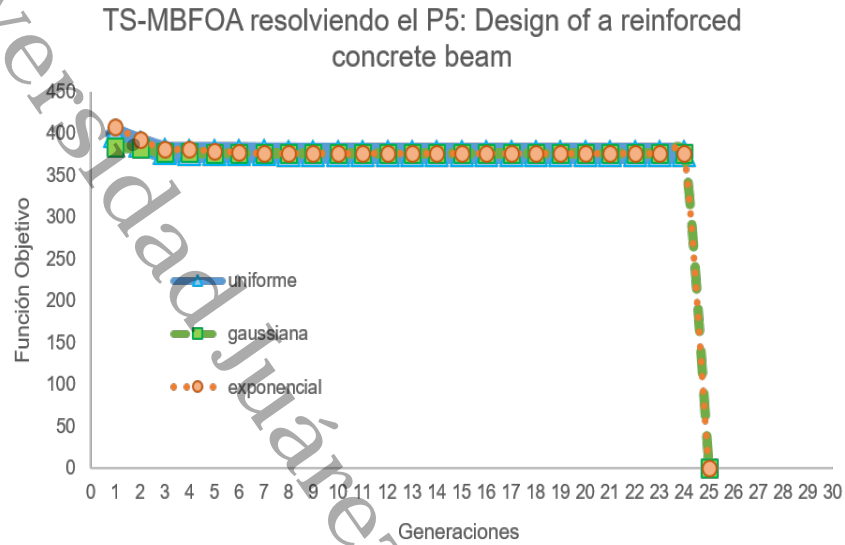
**Figura 4.3.** Gráfica de convergencia del P2: Recipiente de presión con distribución uniforme, gaussiana y exponencial



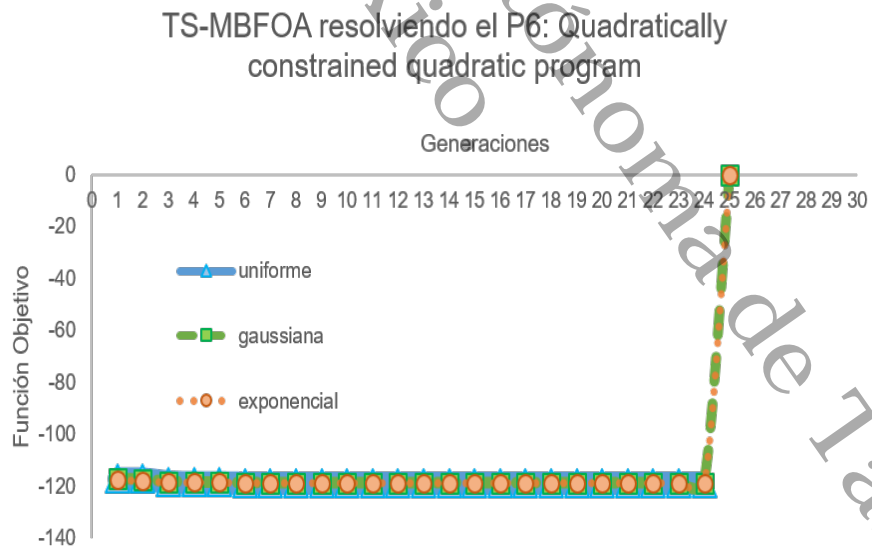
**Figura 4.4.** Gráfica de convergencia del P3: Viga soldada con distribución uniforme, gaussiana y exponencial



**Figura 4.5.** Gráfica de convergencia del P4: Síntesis de procesos MINLP con distribución uniforme, gaussiana y exponencial



**Figura 4.6.** Gráfica de convergencia del P5: Diseño de una viga de concreto reforzada con distribución uniforme, gaussiana y exponencial



**Figura 4.7.** Gráfica de convergencia del P6: Un problema cuadrático restringido con distribución uniforme, gaussiana y exponencial

En la Gráfica 4.2 se puede observar que el Algoritmo TS-MBFOA con distribución exponencial converge de manera más rápida antes de la generación número 5, mientras que los algoritmos con distribución uniforme y gaussiana convergen después de la generación número 5, en la Gráfica 4.3 podemos observar que el algoritmo con distribución gaussiana y exponencial convergen antes de la generación 10, en cambio el algoritmo con distribución uniforme converge después de la generación 10.

En la Gráfica del problema 3 4.4 podemos observar que las 3 distribuciones convergen al mismo tiempo en la generación 10, mientras que, en la Gráfica 4.5 podemos observar que el algoritmo con distribución gaussiana converge antes de la generación 10 y es más estable, mientras que el algoritmo con distribución uniforme converge una generación después y el algoritmo con distribución exponencial converge después de la generación 10 y en las Gráficas 4.6, 4.7 el Algoritmo TS-MBFOA con las 3 distribuciones convergen al mismo tiempo antes de la generación 5.

#### 4.2.4. Discusión de resultados

El Algoritmo TS-MBFOA cambia su comportamiento dependiendo de la distribución que se utilice, ya que al utilizar la distribución uniforme las bacterias se distribuyen de manera dispersa en todo el espacio de búsqueda y esto puede ocasionar que tarde un poco más en encontrar el resultado óptimo, en cambio con la distribución gaussiana las bacterias se distribuyen en forma de campana haciendo que se concentren más hacia el centro y queden pocas bacterias en los límites y por su parte la distribución exponencial concentra más las bacterias hacia el centro del espacio de búsqueda como podemos observar en la Gráfica 4.1.

Si realizamos una comparación con los trabajos similares encontrados en la literatura especializada podemos observar que al igual que en esta investigación encuentran mejorías. En el [23] su mejora es en cuanto a la capacidad de exploración del algoritmo y que no se atasque en los óptimos locales. En la investigación realizada en [4] obtienen un mejor rendimiento evitando la convergencia prematura. En [26] se generan soluciones casi optimas en un menor número de generaciones. No obstante en la investigación [12] obtienen como conclusión que la efectividad de el algoritmo depende de las caracterís-

ticas y el tipo de problema que se esté resolviendo. Todas estas mejoras las encuentran probando una distribución diferente a la uniforme que como se ha mencionado anteriormente es la distribución por defecto.

En general, el Algoritmo TS-MBFOA encuentra valores similares con cualquiera de las 3 distribuciones (uniforme, gaussiana y exponencial) ya que en la prueba no paramétrica de Kruskal-Wallis no se encontró significancia en los datos. Pero en la Tabla de las estadísticas básicas 4.2 se puede apreciar que para el Problema 1 se obtuvo una mejor solución con la distribución uniforme, pero igual se obtiene una mejor media, peor y desviación estándar con la distribución exponencial. En el Problema 2 el mejor resultado se obtuvo con la distribución gaussiana pero su mejor media, peor y desviación estándar se obtuvo con la distribución uniforme. Mientras que el Problema 3 obtiene el mejor resultado con la distribución exponencial y la mejor media, peor y desviación estándar la tiene con la distribución gaussiana. Para el Problema 4 se obtiene un mejor resultado y mejor media con la distribución gaussiana, mientras que el mejor peor resultado y mejor desviación estándar las tiene con la distribución exponencial. En el Problema 5 los resultados son iguales pero obtiene un mejor peor resultado con la distribución gaussiana y exponencial y la mejor desviación estándar la obtiene en la distribución exponencial. Por último, el Problema 6 donde los resultados estadísticos nos dan el mismo valor, pero tiene una mejor desviación estándar la distribución uniforme.



## Capítulo 5

# Contribuciones, conclusiones y trabajos futuros

### 5.1. Contribuciones

El Algoritmo basado en el forrajeo de bacterias E. Coli TS-MBFOA fue probado con distintas distribuciones. La versión original del algoritmo crea su población de bacterias con distribución uniforme, para esto se propuso reiniciar la mitad de la población de bacterias (las peores) con una población de bacterias aleatorias con distribuciones gaussiana y exponencial con el objetivo de observar el rendimiento del algoritmo en los PONRs. Este reinicio de la población se lleva a cabo en una sola ocasión cuando la mitad de población de bacterias es factible debido a que se hace uso de la media y desviación estándar de esta mitad de bacterias para poder generar las nuevas bacterias con distribución gaussiana o exponencial. En la lectura del estado del arte se encontraron reportes de algoritmos usados con distribuciones diferentes, así como la gaussiana, pero el rendimiento con el Algoritmo TS-MBFOA es algo que todavía no se había estudiado.

### 5.2. Conclusiones

Se realizaron 7 pruebas del algoritmo en distintos problemas de benchmark. En el primero se ejecutó el problema de la esfera para observar de manera gráfica como se distribuyen las bacterias en el espacio de búsqueda, donde pu-

dimos observar que con la distribución uniforme las bacterias se distribuyen de forma dispersa en todo el espacio de búsqueda, la distribución gaussiana se concentra más hacia el centro del lado positivo dejando pocas bacterias en los límites, a diferencia de la distribución exponencial donde las bacterias se concentran más en el centro del espacio de búsqueda. En los otros 6 experimentos se probó el Algoritmo TS-MBFOA en los problemas de optimización numérica con restricciones: Resorte de tensión/compresión, Recipiente de presión, Viga soldada, síntesis de procesos MINLP, diseño de una viga de concreto reforzada y un problema cuadrático restringido, en los cuales se realizaron 30 ejecuciones independientes en las tres versiones del Algoritmo con distribución uniforme, gaussiana y exponencial. Donde los resultados en el Problema 1, Problema 4 y Problema 5 son mejores con el Algoritmo con distribución exponencial, mientras que en el Problema 3 los resultados son mejores con el Algoritmo que tiene distribución gaussiana y en el Problema 2 y 6 los resultados con la distribución uniforme son mejores. Sin embargo, la prueba no paramétrica de Kruskal-Wallis test con el 95% de confianza determinó que no hay diferencia significativa entre los resultados de las tres versiones del algoritmo.

Cada uno de los objetivos propuestos en esta investigación fueron alcanzados satisfactoriamente ya que se analizaron los diferentes tipos de distribuciones para realizar una selección de las distribuciones a utilizar con ayuda de un experto en el tema. Con base en el estado del arte encontrado, se procedió a codificar el Algoritmo TS-MBFOA e implementar las distribuciones poblacionales previamente seleccionadas (Gaussiana y Exponencial) para así realizar las pruebas empezando con el Problema de la esfera para observar como se comportaba cada uno de las distribuciones incluyendo la distribución uniforme que es la distribución por defecto en el algoritmo y después de haber realizado esta prueba se empezaron a resolver 6 problemas de diseño ingenieril y químico encontrados en la literatura especializada. Al final se analizaron los resultados obtenidos con las 3 distribuciones mencionadas y se compararon mediante las estadísticas básicas, graficas de convergencia y la prueba de Kruskal-Wallis donde se obtuvo que el rendimiento de las 3 distribuciones es similar, pero tiene pequeñas mejoras con la distribución exponencial.

En base a estos resultados podemos concluir que, aunque el Algoritmo TS-MBFOA sí tiene resultados diferentes en cada problema con las tres distri-

buciones no tiene un impacto ni positivo ni negativo al generar la población con diferente distribución ya que no hay diferencia en los resultados, pero que ante los problemas de optimización numérica con restricciones abordados en este trabajo se puede obtener un mejor rendimiento del TS-MBFOA con la distribución exponencial.

### 5.3. Trabajos Futuros

Como trabajo futuro se espera probar en más problemas de optimización numérica con restricciones las poblaciones con distintas distribuciones, reinicializar la población de una manera auto adaptativa y realizar un análisis más completo sobre el costo computacional del Algoritmo TS-MBFOA usando las distintas distribuciones.

## Bibliografía

- [1] Rosa Laura Aquino de la Cruz and Juan David Gaitán Capetillo. Tesis de licenciatura: Normalización para el espacio de búsqueda de problemas de optimización numéricos. universidad juárez autónoma de tabasco, Agosto 2017.
- [2] Jaroslaw Arabas and Karol Opara. Population diversity of non-elitist evolutionary algorithms in the exploration phase. *IEEE*, 2019.
- [3] Guillermo Ayala. Media y varianza. In *Estadística Básica*. Universidad de Valencia, 2019.
- [4] Ming Chen, Xiaojun Qiu, and Hong Wang. An effective bacterial foraging optimization based on conjugation and novel step-size strategies. *EasyChair preprint*, (1370), 2019.
- [5] Kalyanmoy Deb, S Agrawal, A Pratap, and Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, 2000.
- [6] Universidad Autónoma del Estado de Hidalgo. Probabilidad y estadística: apuntes digitales. 2012.
- [7] Leandro dos Santos Coelho, Helon Vicente Hultmann Ayala, and Viviana Cocco Mariani. A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization. *Elsevier AMC*, 234:452–459, 2014.
- [8] Epidat. Ayuda de distribuciones de probabilidad. In *Distribuciones de Probabilidad*. Sergas (Servizo Galego de Saúde), 2014.

- [9] Montero Espinosa. Distribución exponencial, 2021.
- [10] Montero Espinosa. Distribución uniforme, 2021.
- [11] A. Floudas C. A and A.R.Ciric. Global optimum search for nonconvex nlp and minlp problems. *Computers chem. Engng*, 13:1117–1132, 1989.
- [12] T. Ganesan, P. Vasant, and I. Elamvazuthi. Non-gaussian random generators in bacteria foraging algorithm for multiobjective optimization. *Ind Eng Manage*, 4(182), 2015.
- [13] Margarita I. Hernández Hernández and Jorge de Jesús Brito Ríos. Algoritmo basado en el forrajeo de bacterias normalizado aplicado a smart grids, Febrero 2019.
- [14] Betania Hernández-Ocaña, Efrén Mezura-Montes, and Pilar Pozos-Parra. A review of the bacterial foraging algorithm in constrained numerical optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC'2013)*, pages 2695–2702. IEEE, 2013.
- [15] Betania Hernández-Ocaña, M.D.P. Pozos-Parra, and Efrén Mezura-Montes. Applied mathematics and information sciences. In *Improved modified bacterial foraging optimization algorithm to solve constrained numerical optimization problems*, page 607–622. AMIS, 2016.
- [16] L. J. Liebman, Ls.Lasdon and A. Waren. Modeling and optimization with gino the scientific press. *Palo Alto, CA*, 1986.
- [17] E. Mezura-Montes and C. Coello. Constrained optimization via multi-objective evolutionary algorithms, in j. knowles, d. corne and k. deb, eds, ‘multiobjective problem solving from nature. *Springer*, page 53–75, 2008.
- [18] Efrén Mezura-Montes and Betania Hernández-Ocaña. Modified bacterial foraging optimization for engineering design. In Cihan H. Dagli and et al., editors, *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE'2009)*, volume 19 of *Intelligent Engineering Systems Through Artificial Neural Networks*, pages 357–364, St. Louis, MO, USA, November 2009. ASME Press.
- [19] Minitab.com. Distribución uniforme, 2019.

- [20] Cecilia Salazar P. and Santiago del Castillo G. Medidas de dispersión de datos no agrupados. In *FUNDAMENTOS BÁSICOS DE ESTADÍSTICA*, pages 67, 68, 69. PRIMERA EDICIÓN, 2018.
- [21] Kevin M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, 2002.
- [22] Paula Nicole Roldán. Estadística. economipedia.com, Julio 2017.
- [23] Priyanka Singh, Pragya Dwivedi, and Vibhor Kant. A hybrid method based on neural network and improved environmental adaptation method using controlled gaussian mutation with real parameter for short-term load forecasting. *Elsevier Energy*, 174:460–477, 2019.
- [24] Social Science Statistics. Kruskal-wallis test calculator, 2021.
- [25] Superprof. Ejercicios para el calculo de la distribución normal y la probabilidad, Febrero 2020.
- [26] Sreenivas Sremath Tirumala. A quantum-inspired evolutionary algorithm using gaussian distribution-based quantization. *Springer*, 2017.
- [27] V.Manousiouthakis and D.Sourlas. A global optimization approach to rationally constrained rational programming. *Chem. Engng. Commun*, 115:127–147, 1992.
- [28] Yinglin Xia. Chapter eleven - correlation and association analyses in microbiome study integrating multiomics in health and disease. In Jun Sun, editor, *The Microbiome in Health and Disease*, volume 171 of *Progress in Molecular Biology and Translational Science*, pages 309–491. Academic Press, 2020.
- [29] Huiling Chen and Qian Zhang, Jie Luo, Yueting Xu, and Xiaoqin Zhang. An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. *Elsevier*, 86:1568–494, 2019.

Universidad Juárez  
Móxico.  
Benito Juárez  
Toluca  
Culiacán  
Minomoma de Tabasco.

# AneXo



LA SOCIEDAD MEXICANA DE INTELIGENCIA ARTIFICIAL  
Y LA UNIVERSIDAD DE SONORA

OTORGAN ESTE CERTIFICADO A:

Margarita I. Hernández-Hernández, Betania Hernández-Ocaña, José Adán Hernández Nolasco, Oscar A. Chávez-Bosquez y José Hernández-Torruco

por la presentación del artículo titulado:

**Rendimiento del algoritmo basado en el forrajeo de bacterias con distribución uniforme, gaussiana y exponencial**

**en el XIII Congreso Mexicano de Inteligencia Artificial, COMIA 2021**

Hermosillo, Sonora, México, del 18 al 21 de mayo de 2021

Dr. Félix Castro Espinoza  
Presidente SMIA

Dr. Oscar Herrera Alcántara  
Presidente del Comité de Programa

Dr. Noé A. Castro Sánchez  
Presidente del Comité de Programa

Dra. María Elena Robles Baldenegro  
Comité Local COMIA