

Algoritmo basado en el forrajeo de bacterias para la optimización de un *smart grid*

Betania Hernández-Ocaña¹, Oscar Chávez-Bosquez¹, José Hernández-Torruco¹,
Edgar A. Portilla-Flores², Luis G. Montané Jiménez³,
Margarita I. Hernández-Hernández¹, Jorge J. Brito-Ríos¹

¹ Universidad Juárez Autónoma de Tabasco, DAIS, Tabasco, México

² Instituto Politécnico Nacional, CIDETEC, Ciudad de México, México

³ Universidad Veracruzana, FEI, Xalapa, México

{betania.hernandez,oscar.chavez,jose.hernandezt}@ujat.mx,
{magui_hernandez96,jor.95}@hotmail.com, aportilla@ipn.mx, lmontane@uv.mx

Resumen. En este artículo, un algoritmo basado en el forrajeo de bacterias E.Coli llamado TS-MBFOA es implementado para optimizar un modelo matemático que minimiza a un *Smart Grid*, el cual es una red de distribución eléctrica combinada con modernas tecnologías de información que busca reducir los costos y la contaminación al medio ambiente. En este *Smart Grid* se hacen uso de fuentes de energía renovables como la eólica y solar, además de una unidad de generación convencional a partir del combustible diésel. TS-MBFOA fue implementado y probado en un primer experimento para calibrar los parámetros propios del algoritmo. En un segundo experimento, el algoritmo previamente calibrado fue ejecutado 30 veces de manera independiente para observar su rendimiento. Los resultados encontrados muestran que TS-MBFOA es mejor en comparación con dos algoritmos evolutivos utilizados para resolver el mismo problema de optimización.

Palabras clave: optimización, forrajeo de bacterias, algoritmo de inteligencia colectiva, red de distribución eléctrica inteligente.

Bacterial Foraging-Based Algorithm for *Smart Grid* Optimization

Abstract. In this paper, an E. Coli bacterial foraging-based algorithm called TS-MBFOA is implemented to optimize a mathematical model that minimizes a *Smart Grid*, which is an electrical distribution network combined with modern information technologies which seeks to reduce costs and environmental pollution. In this *Smart Grid*, renewable energy sources such as wind and solar are used, as well as a conventional generation unit from diesel fuel. TS-MBFOA was implemented and tested in a first experiment to calibrate the algorithm parameters. In a second experiment, the previously calibrated algorithm was executed 30 times independently to observe its performance. The results found show that

TS-MBFOA is better compared to two evolutionary algorithms used to solve the same optimization problem.

Keywords: optimization, bacterial foraging, swarm intelligence algorithm, smart grid.

1. Introducción

Debido al gran consumo de la energía eléctrica causado por la alta densidad demográfica que ha alcanzado nuestro planeta, a la preocupación del cambio climático por los altos índices de contaminación a causa del uso de aparatos eléctricos que emiten dióxido de carbono y a las fuentes de generación de energía eléctrica que pueden ser renovables o no renovables, se requiere del control y administración del consumo de energía eléctrica. Entre algunas medidas que se han tomado se encuentran las *Smart Grids*. Éstas son una herramienta tecnológica que permite un mejor desempeño de la energía eléctrica de manera inteligente (autónoma). Además, integra en una red las acciones de los usuarios y consumidores que se encuentran conectados a ella de manera inteligente, con el fin de conseguir un suministro eléctrico, seguro y sostenible [7]. La implementación generalizada de esta tecnología podría incidir en la reducción de los precios para todos los consumidores y reducir las emisiones de carbono por medio del mejoramiento de la eficiencia, de la respuesta a la demanda y de la gestión de la carga de la red eléctrica [4].

México ha puesto en marcha mecanismos para disminuir las emisiones contaminantes derivadas de la generación y el uso de la energía. Desde el punto de vista de la producción, uno de los principales objetivos de la Reforma Energética es que nuestro país utilice combustibles más limpios y mejores tecnologías. En esta lógica se inscriben los esfuerzos para sustituir el diésel y el combustóleo por gas natural en la generación de electricidad, al igual que el ingreso de empresas que utilicen procesos menos contaminantes para la extracción y transformación del petróleo [16].

El Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC) del Instituto Politécnico Nacional (IPN) está trabajando en el desarrollo de modelos matemáticos para las *Smart Grids* y uno de ellos es propuesto como un problema de optimización que calcula las potencias óptimas de una red de dispositivos de generación de energía para suministrar a una carga durante un periodo de 24 horas. En esencia, cada hora se resuelve un problema de optimización de acuerdo a las condiciones de la red y restricciones de operación de la misma. Al final se tiene una gráfica de las potencias óptimas y la suma de las 24 funciones objetivo en la cual se propone qué solución es mejor utilizar.

Este tipo de problemas de optimización son complejos y aunque por programación matemática pueden resolverse, puede resultar complicado y caótico implementarla. Actualmente existen las metaheurísticas, herramientas que permiten resolver problemas de optimización con o sin restricciones, combinatorios o numéricos de manera aproximada, es decir, generan una o más soluciones

factibles cercanas al óptimo. Un problema de optimización es también conocido como un problema general de programación no-lineal y se puede definir como: Minimizar $f(\vec{x})$ sujeta a: $g_i(\vec{x}) \leq 0$, $i = 1, \dots, m$ y/o $h_j(\vec{x}) = 0$, $j = 1, \dots, p$. Donde $\vec{x} \in \mathbb{R}^n$ tal que $n \geq 1$, es el vector de soluciones $\vec{x} = [x_1, x_2, \dots, x_n]^T$, donde cada x_i , $i = 1, \dots, n$ está delimitada por el límite inferior y superior $L_i \leq x_i \leq U_i$; m es el número de restricciones de desigualdad y p es el número de restricciones de igualdad (en ambos casos, las restricciones podrían ser lineales o no lineales). Si denotamos con F a la región factible (donde se encuentran todas las soluciones que satisfacen al problema) y con S a todo el espacio de búsqueda, entonces $F \subseteq S$.

Los algoritmos bio-inspirados forman parte de las metaheurísticas y se dividen en dos grandes grupos: 1) los Algoritmos Evolutivos (AEs), cuyo funcionamiento se basa en emular el proceso de evolución natural y la supervivencia del más apto [6] y 2) los Algoritmos de Inteligencia Colectiva (AICs) que basan su funcionamiento en comportamientos sociales y cooperativos de organismos simples e inteligentes como insectos, aves y bacterias [8].

A partir de las ideas iniciales de Bremermann [3], en el año 2002 Passino [14] propone un AIC llamado Algoritmo de Optimización del Forrajeo de Bacterias (BFOA, por las siglas en inglés de *Bacterial Foraging Optimization Algorithm*), en el cual cada bacteria E.Coli trata de maximizar su energía obtenida por unidad de tiempo empleada en el proceso de forrajeo, donde también evade sustancias nocivas. Más aún, las bacterias se pueden comunicar entre sí mediante la segregación de sustancias. En BFOA se tienen cuatro procesos principales: (1) quimiotaxis (nado-giro), (2) agrupamiento, (3) reproducción y (4) eliminación-dispersión. Las bacterias son soluciones potenciales al problema y su ubicación representa los valores de las variables de decisión del problema. Las bacterias pueden moverse (generar nuevas soluciones) mediante el ciclo quimiotáxico; se genera además un movimiento mediante la atracción que ejercen soluciones en zonas prometedoras del espacio de búsqueda, se permite la reproducción de las mejores soluciones y finalmente se eliminan del cúmulo aquellas bacterias localizadas en zonas de baja calidad.

En lo referente a propuestas de solución de instancias del problema de optimización global con restricciones, en 2009 se propuso una versión simplificada de BFOA llamada Algoritmo de Optimización del Forrajeo de Bacterias Modificado (MBFOA, por sus siglas en inglés) [11]. En MBFOA se tiene un mecanismo para el manejo de las restricciones basado en Reglas de factibilidad, las cuales consisten en: a) Entre dos soluciones factibles, aquella con el mejor valor de la función objetivo es seleccionada. b) Entre una solución factible y otra no factible, la factible es seleccionada y c) Entre dos soluciones no factibles, aquella con la menor suma de violación de restricciones es seleccionada [5]. Por último, una disminución de parámetros respecto a los del BFOA original.

Además MBFOA se aplicó a un conjunto de problemas de diseño en ingeniería química y mecánica obteniendo resultados competitivos. Por otro lado, MBFOA se aplicó a la resolución de un problema de diseño mecánico bi-objetivo en presencia de restricciones [12]. Una propuesta más reciente basada en MBFOA

usa un operador similar al de mutación (usado en AEs) como operador de nado dentro del proceso quimiotáxico, además del uso de un nado aleatorio en este mismo proceso y un mecanismo de sesgo para la población inicial basado en el rango de las variables también fue anexado. Esta propuesta es llamada TS-MBFOA (por sus siglas en inglés) [9].

En este artículo se propone utilizar la metaheurística TS-MBFOA para minimizar la *Smart Grid* o Red de Energía Eléctrica Inteligente propuesta por el CIDETEC-IPN, puesto que en el estado del arte no se reportan trabajos similares usando este algoritmo, sin embargo, se han utilizado AEs para resolver instancias relacionadas con los *SmartGrids* [1]. Este documento está organizado de la siguiente manera: En la Sección 2 se presenta el modelado matemático del *Smart Grid*. En la Sección 3 se describe brevemente a TS-MBFOA. En la sección 4 se presentan los resultados obtenidos y la discusión de éstos. Por último en la Sección 5 se presentan las conclusiones y trabajos futuros.

2. *Smart grids*

Las redes de distribución eléctrica suelen alcanzar su máxima capacidad de operación y además deben adaptarse a ciertos cambios del medio ambiente que influyen en su funcionamiento. Esta problemática se soluciona mediante el uso de Generadores Distribuidos (GeDis) que permiten la explotación de Fuentes de Energía Renovable (FER); en este caso eólica y solar, y alternativa, que se refiere a los combustibles. A pesar de que los GeDis solucionan la problemática planteada, al introducir el uso de energías renovables surge el reto de su administración, que se vuelve complicado debido a la fuerte intermitencia que presentan dichas energías a causa de la climatología [2].

Las microrredes (MRs) permiten coordinar los GeDis para proporcionar electricidad de forma exitosa, lo cual las convierte en soluciones fiables para el reto mencionado. Una MR suele estar conformada por Sistemas de Almacenamiento de Energía (SAE), sistemas de generación híbrida a partir de FER y unidades de generación convencional a partir de combustibles como el diésel. En términos monetarios, los SAE son muy costosos y deben operarse de forma segura, garantizando que tengan una larga vida útil. Adicionalmente, las unidades de generación convencional tienen un alto costo de operación debido a la logística necesaria para transportarlas y a los materiales con los que están construidas. Es por esto, que los GeDis deben gestionarse de forma óptima para el funcionamiento de las MRs.

En este caso de estudio, se debe optimizar la generación de energía en una MR que se encuentra en un lugar Remoto No Interconectable (RNI). En estos lugares, rurales en la mayoría de los casos, la generación fiable de energía se consigue a través de generación híbrida de potencia en donde se emplean FER, generación convencional con diésel y SAE. El reparto de potencias para suministrar energía a la carga de una Microrred Remota No Interconectable (MR-RNI), se calcula como una función de despacho económico usualmente empleada para generadores de gran escala de potencia [19], en donde se establece una función de costo a

minimizar sujeta a restricciones de capacidad de los generadores y de balance entre la carga generada y demandada.

2.1. Microrred Remota No Interconectable (MR-RNI)

La MR-RNI está conformada por un conjunto de cargas y un sistema híbrido de generación de potencia que a su vez se compone de un sistema de Generación Eólica (GE), un sistema de Generación Solar Fotovoltáico (GSF), un Generador de Diésel (GD) y un SAE. Las variables de control son las potencias de los generadores y el SAE, y funcionan como entradas del sistema de optimización. Las salidas de control, son el reparto de carga por cada generador y el costo de generación. A continuación, se describe el cálculo del costo para cada generador.

Costo de Generación Diésel. Los GD tienen una función de costo asociada a la potencia del generador. Dicha función se encuentra dada por la Ecuación 1, en donde i es la fuente de generación i , P_i y F_i son la salida de potencia eléctrica y el costo de operación de la fuente i , respectivamente, mientras que α , β y γ son los coeficientes de costo.

$$f_i(P_i) = \alpha_i + \beta_i P_i + \gamma_i P_i^2. \quad (1)$$

Para el GD, la función de costo queda como lo muestra la Ecuación 2:

$$f_1(P_1) = 14.88 + 0.3P_1 + 0.000435P_1^2. \quad (2)$$

Costo de Generación Solar Fotovoltáica. Para el generador GSF, la función de costo se define en la Ecuación 3:

$$f_3(P_3) = \alpha I^P P_3 + G^E P_3, \quad (3)$$

$$\alpha = \frac{r}{[1 - (1+r)^{-N}]}, \quad (4)$$

en donde P_3 es la potencia de salida de GSF, α es la tasa de retorno de inversión y se calcula con la Ecuación 4, r corresponde a la tasa de interés (se considera de 0.09 para el caso base), N se refiere a la vida útil, que ha sido establecida en 20 años, I^P indica el costo de inversión por unidad de potencia ($\$5000/kW$) y G^E es el costo de operación y mantenimiento por unidad de potencia generada ($\$1.6/kW$).

De acuerdo a la Ecuación 3, la función de costo para GSF queda establecida con la Ecuación 5:

$$f_3(P_3) = 545.016P_3. \quad (5)$$

Costo de Generación Eólica. Siguiendo el esquema de costos presentado en GSF, y tomando los mismos valores de I^P y G^E , la función de costo para la generación eólica se define con la Ecuación 6:

$$f_4(P_4) = 152.616P_4. \quad (6)$$

Costo de Sistema de Almacenamiento de Energía. Finalmente, para el SAE se considera un banco de baterías de $2kW$. El costo I^P por unidad de

almacenamiento instalada es de $\$1000/kW$, mientras que el G^E conserva el valor de $\$1.6/kW$ dando como resultado la siguiente función de costo definida en la Ecuación 7:

$$f_2(P_2) = 119P_2. \quad (7)$$

2.2. Declaración del problema de optimización

Para la optimización de la generación de energía se emplea el método de despacho económico, en donde se deben encontrar las potencias que serán suministradas por los generadores de la MR-RNI en cada hora del día, de tal forma que el costo sea lo menor posible. Para este caso de estudio, el vector de variables de diseño se define en la Ecuación 8:

$$\vec{p} = [P_1, P_2, P_3, P_4]^T. \quad (8)$$

Función Objetivo. La formulación clásica para el despacho económico es definida con la Ecuación 9:

$$\text{Min } F = \sum_{i=1}^{NG} f_i(P_i). \quad (9)$$

La función objetivo considerada en la gestión de la generación para la MR-RNI en la hora τ es definida con la Ecuación 10:

$$\text{Min } f(\vec{p}_\tau) = w_1 C_f f(P_1(\tau)) + w_2 f(P_2(\tau)) - w_3 f(P_3(\tau)) - w_4 f(P_4(\tau)), \quad (10)$$

en donde w_1 , w_2 , w_3 y w_4 son los pesos asignados a los generadores, los cuales suman 1 y han sido establecidos en 0.25, mientras que C_f es el costo del combustible diésel, que se asignó en USD $\$1$. Considerando que el costo de generación se debe calcular por cada hora del día, la función objetivo para el costo total durante un día completo queda definida con la Ecuación 11:

$$\text{Min } F = \sum_{\tau=1}^{24} f(\vec{p}_\tau) \quad \vec{p}_\tau \in \mathbb{R}^4. \quad (11)$$

Restricciones de diseño. Para que la gestión de la generación se realice de forma correcta, deben considerarse las siguientes restricciones:

Balance de potencia: La suma de las potencias suministradas por los generadores debe ser igual a la carga total del sistema, esto es definido en la Ecuación 12:

$$P_1 + P_2 + P_3 + P_4 = P_L. \quad (12)$$

Modelo del banco de baterías: El poder de salida del generador PV y la carga demandada en cierta hora t , determinan el poder de carga y descarga dentro y fuera del banco de baterías. t es un entero que representa la t -ésima hora.

Tabla 1. Parámetros del Sistema de Almacenamiento de Energía (SAE).

Parámetros	%
Eficiencia de carga (n_C)	85
Eficiencia de descarga (n_D)	100
Estado máximo de carga (SOC^{max})	95
Estado mínimo de carga (SOC^{min})	40

El Estado de Carga (SOC) del banco de baterías en cualquier hora, $SOC(t)$, depende del SOC en la hora previa $SOC(t - 1)$. Para el flujo de energía de la hora $t - 1$ a t , se debe considerar la Ecuación 13:

$$SOC_t = SOC(t - 1) - \alpha_D P_2(t) + \alpha_C P_3(t) + \alpha_C P_4(t), \quad (13)$$

en donde $\alpha_D = \frac{n_D}{B_{c^{max}}}$, $\alpha_C = \frac{n_C}{B_{c^{max}}}$, mientras que n_C y n_D son la eficiencia de carga y descarga de la batería, respectivamente. La Ecuación 14 se puede aplicar para la dinámica de la batería [17]:

$$SOC(t - 1) = SOC(0) - \alpha_D \sum_{\tau=1}^t P_2(\tau) + \alpha_C \sum_{\tau=1}^t P_3(\tau) + \alpha_C \sum_{\tau=1}^t P_4(\tau), \quad (14)$$

en la cual $SOC(0)$ es considerado como el SOC inicial de la batería. $\alpha_C \sum_{\tau=1}^t P_3(\tau) + \alpha_C \sum_{\tau=1}^t P_4(\tau)$ es el poder aceptado por la batería en la hora t y $\alpha_D \sum_{\tau=1}^t P_2(\tau)$ es el poder de descarga de la batería en el tiempo t .

La capacidad disponible de la batería no debe ser menor que la capacidad mínima permitida ni mayor que la capacidad máxima permitida [17], esto es definido por la Ecuación 15:

$$SOC^{min} \leq SOC(t) \leq SOC^{max}, \quad (15)$$

es decir, detalladamente con la Ecuación 16 :

$$SOC^{min} - SOC(t) \leq 0, \quad SOC(t) - SOC^{max} \leq 0. \quad (16)$$

Los parámetros del banco de baterías se muestran en la Tabla 1.

Finalmente, el problema de optimización queda descrito por las Ecuaciones 17 a 20.

$$Min F = \sum_{\tau=1}^{24} f(\vec{p}_\tau) \quad \vec{p}_\tau \in \mathbb{R}^4, \quad (17)$$

sujeito a:

$$g_1(\vec{p}_\tau) = P_1(\tau) + P_2(\tau) + P_3(\tau) + P_4(\tau) = P_L(\tau), \quad (18)$$

$$g_2(\vec{p}_\tau) = SOC^{min} - SOC(t) \leq 0, \quad (19)$$

$$g_3(\vec{p}_\tau) = SOC(t) - SOC^{max} \leq 0. \quad (20)$$

La carga $P_L(\tau)$ fue obtenida de [18] y los límites superior e inferior de cada variable se encuentran descrito por las Ecuaciones 21 a 24:

$$0 \leq P_1(\tau) \leq GD_{nominal}, \quad (21)$$

$$0 \leq P_2(\tau) \leq SOC(0) \times Bc_{max} - SOC^{min} \times Bc_{max}, \quad (22)$$

$$0 \leq P_3(\tau) \leq P_{pv}(\tau), \quad (23)$$

$$0 \leq P_4(\tau) \leq P_{wind}(\tau), \quad (24)$$

en donde $GD_{nominal}$ se refiere a la capacidad nominal del generador diésel, que para este caso es de 5000 VA (5 kVA). El límite superior para $P_2(\tau)$ depende de la capacidad máxima de la batería (Bc_{max}) que es de 2000 VA (2 kVA) y el estado inicial de la misma. Los datos de $P_{pv}(\tau)$ y $P_{wind}(\tau)$ donde P_{pv} es la potencia de salida fotovoltaica y P_{wind} es la potencia de salida eólica y fueron tomados de [13].

3. TS-MBFOA

El TS-MBFOA es un algoritmo derivado de MBFOA propuesto para resolver PONR [9], en el cual una bacteria i representa una solución potencial y se denota como $\theta^i(j, G)$, donde j es el ciclo quimiotáxico y G es el ciclo generacional. Una generación consta de un proceso quimiotáxico, agrupamiento, reproducción y eliminación-dispersión.

En el proceso de quimiotáxis dos nados se intercalan, en cada ciclo solo un nado de explotación o exploración es realizado. El proceso comienza con el nado de explotación (nado clásico). Sin embargo, una bacteria no necesariamente intercalará exploración y explotación en los nados, ya que si la nueva posición de un nado dado, $\theta^i(j+1, G)$ tiene una mejor aptitud (basado en las reglas de factibilidad) que la posición original $\theta^i(j, G)$, otro nado en la misma dirección se llevará a cabo en el siguiente ciclo. De lo contrario, un nuevo giro será calculado. El proceso se detiene después de N_c intentos.

El nado de exploración usa la mutación entre bacterias y es calculado con la Ecuación 25:

$$\theta^i(j+1, G) = \theta^i(j, G) + (\beta)(\theta_1^r(j, G) - \theta_2^r(j, G)), \quad (25)$$

donde $\theta_1^r(j, G)$ y $\theta_2^r(j, G)$ son dos bacterias diferentes seleccionadas aleatoriamente de la población. β es un parámetro definido por el usuario utilizado en el operador de agrupamiento el cual define la cercanía de la nueva posición de una bacteria con respecto a la posición de la mejor bacteria de la población, en este operador, β es un parámetro de control positivo para escalar los diferentes vectores en (0,1], es decir, escalas de la zona donde una bacteria puede moverse.

El nado de explotación es calculado con el Ecuación 26:

$$\theta^i(j+1, G) = \theta^i(j, G) + C(i, G)\phi(i), \quad (26)$$

donde $\phi(i)$ se calcula con el operador de giro original de BFOA definido en la Ecuación 27:

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}}, \quad (27)$$

$\Delta(i)^T$ es un vector aleatorio generado con elementos dentro de un intervalo $[-1, 1]$.

$C(i, G)$ es el tamaño de paso aleatorio de cada bacteria actualizado con la Ecuación 28:

$$C(i, G) = R * \Theta(i), \quad (28)$$

donde $\Theta(i)$ es un vector generado de forma aleatoria de tamaño n con elementos dentro del rango de cada variable de decisión: $[U_k, L_k]$, $k = 1, \dots, n$, y R es un parámetro definido por el usuario para escalar el tamaño de paso, este valor debe ser cercana a cero (por ejemplo $5.00e-04$). La inicial $C(i, 0)$ se genera utilizando $\theta(i)$. Este tamaño de paso aleatorio permite que las bacterias se puedan mover en diferentes direcciones dentro del espacio de búsqueda y evita la convergencia prematura como se sugiere en [10].

En el ciclo medio del proceso quimiotáxico es aplicado el operador de agrupamiento con la Ecuación 29, donde β es un parámetro positivo definido por el usuario entre (0,1):

$$\theta^i(j+1, G) = \theta^i(j, G) + \beta(\theta^B(G) - \theta^i(j, G)), \quad (29)$$

donde $\theta^i(j+1, G)$ es la nueva posición de la bacteria i , $\theta^B(G)$ es la actual posición de la mejor bacteria generacional y β es un parámetro llamado factor de escalamiento, el cual regula qué tan cerca estará la bacteria i de la mejor bacteria θ^B . Sin embargo, si una solución viola el límite de las variables de decisión, una nueva solución de x_i es generada aleatoriamente entre los límites inferior y superior $L_i \leq x_i \leq U_i$ de las variables de decisión.

En la reproducción se ordenan las bacterias con base en la técnica de manejo de restricciones, eliminando a las peores bacterias $S_b - S_r$ y duplicando a las mejores cada cierto número de ciclos, definido por el usuario con el parámetro *RepCycle*.

En la eliminación-dispersión se elimina a la peor bacteria de la población $\theta^w(j, G)$ (basado en las reglas de factibilidad) y se genera una nueva aleatoriamente.

Aunque en su propuesta original de TS-MBFOA se utiliza un mecanismo de sesgo para generar la población inicial aleatoria y un buscador local, en este artículo no se hace uso de dicho mecanismo para consumir menos costo computacional. El pseudocódigo de TS-MBFOA es presentado en el Algoritmo 1.

4. Resultados

TS-BFOA fue adaptado y ejecutado para resolver el problema del *Smart Grid* en tres equipos de cómputo con las siguientes características: una PC con Memoria RAM de 4.0GB, procesador de 2.3Ghz y un sistema Operativo

Algoritmo 1: Pseudocódigo de TS-MBFOA.

```

1 Crear una población inicial de bacterias aleatorias  $\theta^i(j, 0) \forall i, i = 1, \dots, S_b$ 
2 Evaluar  $f(\theta^i(j, 0)) \forall i, i = 1, \dots, S_b$ 
3 for  $G=1$  to  $GMAX$  do
4   for  $i=1$  to  $S_b$  do
5     for  $j=1$  to  $N_c$  do
6       En el proceso quimiotáxico intercalar los nados propuestos con las Ecs.
          25 y 26. Aplicar el operador de agrupamiento con la Ec. 29 usando  $\beta$  para la
          bacteria  $\theta^i(j, G)$ 
7     end
8   end
9   if  $G \bmod RepCycle == 0$  then
10    Realizar el proceso de reproducción ordenando la población de acuerdo a las reglas
        de factibilidad y eliminar a  $S_r$  peores bacterias y duplicar el resto de bacterias
         $S_b - S_r$ .
11  end
12  Realizar el proceso de eliminación-dispersión eliminando a la peor bacteria  $\theta^w(j, G)$  de
        la población actual considerando la técnica de manejo de restricciones.
13  Actualizar el vector de tamaño de paso usando la Ec. 28
14 end

```

Windows de 64bits; y dos PC con Memoria RAM de 8.0GB, procesador de 2.4GHZ e igual sistema operativo. El algoritmo fue programado en lenguaje M en el software de desarrollo Matlab R2014a.

Dos experimentos fueron realizados, el primero consistió en calibrar los parámetros del algoritmo y en un segundo experimento se obtuvo un conjunto de soluciones para observar el rendimiento del algoritmo ante tal PONR y comparar su mejor resultado contra el obtenido por el algoritmo DE/rand/1/bin (una versión clásica del algoritmo Evolución Diferencial que emplean en CIDETEC [15]) y contra otro algoritmo evolutivo llamado C-LSHADE [20].

Primer experimento. Para la calibración de los parámetros del algoritmo se realizaron 87 ejecuciones al algoritmo con diversa combinación de parámetros y un número de generaciones de 15,000. Los rangos probados para cada parámetro fueron: S_b entre [10, 200], N_c entre [5,100], S_r entre [1, $S_b/2$], $RepCycle$ entre [10, 200] y R, B entre [0,1]. El mejor resultado obtenido de todas las ejecuciones independientes fue el valor -564959.112 .

Durante las ejecuciones del algoritmo en este primer experimento se observó que a mayor número de bacterias y ciclos quimiotáxicos, el tiempo de ejecución del algoritmo aumentaba de segundos a minutos; con la mejor combinación de parámetros encontrada el algoritmo tarda 14 minutos, en promedio, en resolver el problema con resultados factibles. Otro aspecto encontrado fue que al aumentar el número de bacterias a reproducir ($RepCycle$) la calidad de los resultados del algoritmo disminuía en comparación al usar valores entre 40 y 60 como máximo.

Además, los valores cercanos a cero para el tamaño de paso y factor de escalamiento, R y B , respectivamente, favorecían el rendimiento del algoritmo al generar soluciones de mayor calidad de acuerdo a la función objetivo.

Tabla 2. Detalles de la mejor solución encontrada por TS-MBFOA.

Hora	P1(Diésel)	P2(SAE)	P3(Solar)	P4(Eólica)	Carga	F. Objetivo
00:00	2496.3651	1.6701	1.0000	0.9649	2500	745.2825
01:00	1575.6793	423.8042	0.5166	500	2500	-6147.3127
02:00	2318.9864	15.5409	1.0000	514.4727	2850	-18540.63612
03:00	1620.3690	729.4394	0.1916	600	2950	-806.8983
04:00	1726.7810	122.2190	1	1000	2850	-34196.74276
05:00	2260.0548	44.1808	0.9957	194.7687	2500	-5523.7952
06:00	1592.7925	505.2331	0.5558	51.4187	2150	13392.21051
07:00	1962.2575	20.7425	266	1	2250	-35095.00219
08:00	2190.8451	38.9401	70	0.2148	2300	-7697.4964
09:00	1752.4745	240.1307	327.1400	0.2548	2320	-36970.82458
10:00	1916.7460	195.5353	237.5564	0.1623	2350	-26010.00763
11:00	2203.3221	487.8177	126.0000	132.8602	2950	-7027.6669
12:00	1179.4551	480.6729	589.8720	7.61391E-05	2250	-65828.93738
13:00	665.9984	1094.7179	535.9387	23.3450	2320	-41244.7373
14:00	0.0081	1079.7260	700	570.2659	2350	-85010.15548
15:00	0.0038	1079.7280	560	710.2682	2350	-71276.1857
16:00	71.2158	1092.9549	405.9995	879.8298	2450	-56363.05636
17:00	755.8825	1099.9999	63.0000	1231.1175	3150	-22708.51646
18:00	915.8824	1100	0.9084	1293.2093	3310	-16576.23946
19:00	2399.4340	850.2600	0.3059	1000	4250	-12090.66619
20:00	3562.3782	187.5911	0.0307	500.0000	4250	-11849.35186
21:00	2449.0000	2.67E-14	1	550	3000	-20281.31989
22:00	2422.4054	63.8122	0.1824	463.6000	2950	-14991.08009
23:00	1474.9268	1100.0000	0.7396	74.3335	2650	30139.01468

Segundo experimento. Para observar el rendimiento del algoritmo ante el problema de optimización del *Smart Grid* se optó por utilizar la mejor combinación de parámetros proporcionada por el primer experimento, la cual fue: $S_b = 10$, $N_c = 8$, $S_r = 5$, $R = 0.015$, $B = 0.040$, $RepCycle = 60$ y $GMAX = 15000$. Se realizaron 30 ejecuciones independientes con un número de 15,000 generaciones. La mejor solución obtuvo un valor de -551960.121 como la sumatoria de las 24 funciones objetivo de la ejecución. Tal valor es negativo debido a que representa un ahorro monetario al operar con FERs, en lugar de usar únicamente el Generador Diésel (GD) y el SAE. Entre más energía suministren las FER y menos el GD y el SAE, el ahorro es mayor. Los valores de las variables de esta solución son presentados en la Tabla 2 con los cuales se realizó la gráfica de la Figura 1, en donde se observa que aunque el GD (P1) tenga asignación de carga durante todo el día, de las horas 13 a la 19 la energía generada proviene en gran parte de las FER, lo cual permite un ahorro monetario significativo en el suministro.

Al comparar el mejor resultado del TS-MBFOA contra el algoritmo C-LSHADE y DE/rand/1/bin, se puede observar en la Tabla 3 que el algoritmo basado en el forrajeo de bacterias obtiene mejor resultado para el problema de minimización del MR-RNI, sin embargo se requiere de mayor número de evaluaciones que las

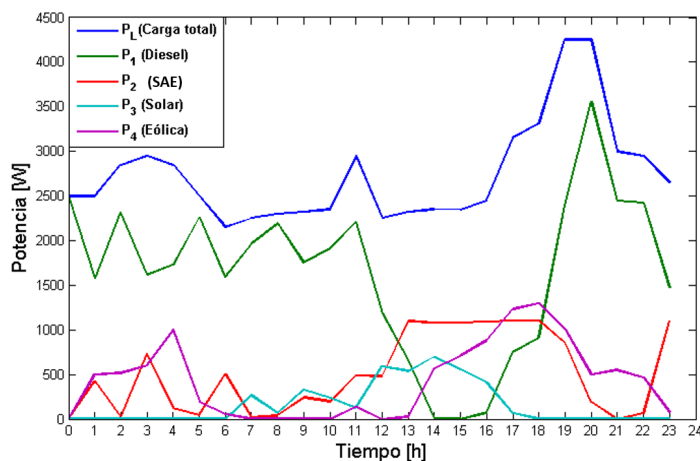


Fig. 1. Visualización del suministro de energía durante las 24 horas del día para el funcionamiento de la MR-RNI, obtenido por el TS-MBFOA

otras dos versiones de algoritmos evolutivos. Cabe mencionar que la evaluaciones son el número de veces que una solución es evaluada en la función objetivo. Para el caso de TS-MBFOA el número de evaluaciones es calculado por $S_b \times N_c \times GMAX$.

Tabla 3. Detalles de la mejor solución encontrada por TS-MBFOA.

Estadística	C-LSHADE	DE/rand/1/bin	TS-MBFOA
Mejor	-5.33E+05	-5.32E+05	-5.52E+05
Peor	-5.32E+05	-5.32E+05	-3.45E+05
Mediana	-5.32E+05	-5.32E+05	-4.98E+05
Promedio	-5.32E+05	-5.32E+05	-4.81E+05
Desv.Est.	8.73E+01	3.30E+00	4.86E+04
Evaluaciones	2.88E+05	2.88E+05	1.20E+06

5. Conclusiones

En este trabajo un algoritmo basado en el forrajeo de las bacterias E.Coli llamado TS-MBFOA fue implementado para resolver un Problema de Optimización Numérico con Restricciones que busca minimizar un *Smart Grid*, el cual es una red inteligente de energía que hace uso de generadores distribuidos que permiten la explotación de fuentes de energía renovables como la eólica y solar así

como de combustibles, por ejemplo el diésel. El modelado matemático del *Smart Grid* fue propuesto por el Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC) del Instituto Politécnico Nacional (IPN). En dicho modelo se calculan las potencias óptimas de una red de dispositivos de generación de energía para suministrar a una carga durante un periodo de 24 horas. En esencia, cada hora se resuelve un problema de optimización de acuerdo a las condiciones de la red y restricciones de operación de la misma. Al final se tiene una gráfica de las potencias óptimas y la suma de las 24 funciones objetivo en la cual se propone qué solución es mejor utilizar.

Dos experimentos fueron diseñados. En el primero, 87 ejecuciones independientes fueron realizadas con diferentes valores a los parámetros del algoritmo con el objetivo de obtener la mejor combinación de valores que permitiera el mejor rendimiento del algoritmo ante tal problema. Como resultado de este primer experimento se obtuvo que el mejor rendimiento del algoritmo fue encontrado con una población de 10 bacterias, 8 ciclos quimiotáxicos, 5 bacterias a reproducir cada 60 generaciones con un tamaño de paso de 0.015 y un factor de escalamiento de 0.040. Además se observó que entre mayor era el número de bacterias y ciclos quimiotáxicos más tiempo de ejecución requería el algoritmo.

En un segundo experimento, el algoritmo fue ejecutado 30 veces de manera independiente usando para los parámetros los valores obtenidos en el experimento previo. Los resultados obtenidos fueron comparados contra dos algoritmos evolutivos dando como mejor algoritmo la propuesta TS-MBFOA según las estadísticas presentadas. Como lectura a la mejor solución encontrada por el algoritmo basado en el forrajeo de bacterias, se hizo uso de su gráfica la cual indicó que aunque el generador de diésel (P_1) tenga asignación de carga durante todo el día, de las horas 13 a la 19 la energía generada proviene en gran parte de las fuentes de energía renovables como la solar (P_3) y eólica (P_4), lo cual permite un ahorro monetario significativo en el suministro.

Como trabajo futuro se buscará que TS-MBFOA disminuya el número de evaluaciones para encontrar soluciones altamente competitivas contra otros algoritmos del estado del arte, además de hacer uso de una población normalizada con el fin de disminuir el tiempo de ejecución de dicho algoritmo.

Referencias

1. Bharathi, C., Rekha, D., Vijayakumar, V.: Genetic algorithm based demand side management for smart grid. *Wireless Personal Communications* 93(2), 481–502 (Mar 2017)
2. Bordons, C., García-Torres, F., Valverde, L.: Gestión Óptima de la energía en microrredes con generación renovable. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 12(2), 117–132 (2015)
3. Bremermann, H.: Chemotaxis and optimization. *J. Franklin Inst* 297, 397–404 (1974)
4. Díaz Andrade, C., Hernández, J.: Smart grid: Las tics y la modernización de las redes de energía eléctrica – estado del arte. *Revista S&T* 9(18), 53–81 (2011)

5. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering* 186(2-4), 311–338 (2000)
6. Eiben, A., Smith, J.: *Introduction to Evolutionary Computing*. Natural Computing Series, Springer-Verlag (2003)
7. *Energía y sociedad: Smart grids* (2017/03/21), <http://www.energiaysociedad.es/manenergia/4-5-smart-grids/>
8. Engelbrecht, A.: *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons (2005)
9. Hernández-Ocaña, B., Pozos-Parra, M., Mezura-Montes, E., Portilla-Flores, E., Vega-Alvarado, E., nez, M.C.Y.: Two-swim operators in the modified bacterial foraging algorithm for the optimal synthesis of four-bar mechanisms. *Computational Intelligence and Neuroscience* 2016, 1–18 (2016)
10. Kasaiezadeh, A., Khajepour, A., Waslander, S.: Spiral bacterial foraging optimization method: Algorithm, evaluation and convergence analysis. *Engineering Optimization* 46(4), 439–464 (2014)
11. Mezura-Montes, E., Hernández-Ocaña, B.: Modified bacterial foraging optimization for engineering design. In: *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE'2009)*, vol. 19, pp. 357–364. in Cihan H. Dagli et al. (editors), ASME Press Series, *Intelligent Engineering Systems Through Artificial Neural Networks* (2009)
12. Mezura-Montes, E., Portilla-Flores, E., Hernández-Ocaña, B.: Optimization of a mechanical design problem with the modified bacterial foraging algorithm. In: *Proceedings of the XVII Argentine Congress on Computer Sciences*. La Plata, Argentina (October 2011)
13. Mikati, M., Santos, M., Armenta, C.: Modelado y simulación de un sistema conjunto de energía solar y eólica para analizar su dependencia de la red eléctrica. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 9(3), 267–281 (2012)
14. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine* 22(3), 52–67 (2002)
15. Portilla-Flores, E., Mezura-Montes, E., Alvarez-Gallegos, J., Coello-Coello, C., Cruz-Villar, A., Villarreal-Cervantes, M.: Parametric reconfiguration improvement in non-iterative concurrent mechatronic design using an evolutionary-based approach. *Engineering Applications of Artificial Intelligence* 5(24), 757–771 (2011)
16. Secretaría de Energía Oaxaca, México: Evaluación rápida del uso de la energía. *SENER* 1(1) (2016)
17. Tazvinga, H., Xia, X., Zhang, J.: Minimum cost solution of photovoltaic–diesel–battery hybrid power systems for remote consumers. *Solar Energy* 96, 292–299 (2013)
18. Tazvinga, H., Zhu, B., Xia, X.: Energy dispatch strategy for photovoltaic–wind–diesel–battery hybrid power system. *Solar Energy* 108, 412–420 (2016)
19. Wood, A., Wollenberg, B.: *Power generation, operation, and control*. John Wiley & Sons, NJ (2012)
20. Zapata, M.Z.: Control de parámetros del algoritmo evolución diferencial con variantes combinadas para la solución de problemas de optimización en mecatrónica (Agosto 2017)