Perfiles de comportamiento numérico de los métodos de búsqueda

*immune network algorithm* y *bacterial foraging optimization algorithm* en funciones *benchmark*

*Performance Profiles of the Algorithms Immune Network Algorithm and Bacterial Foraging Optimization Algorithm in Benchmark Functions*

**Resumen**

En este trabajo se reporta la aplicación del concepto de perfiles de comportamiento numérico en la comparación del desempeño numérico de los métodos *Immune Network Algorithm* y *Bacterial Foraging Optimization* en 18 funciones *benchmark* de optimización. Específicamente la robustez, eficiencia y el tiempo de ejecución de estos métodos se compararon en espacios de búsqueda con múltiples mínimos locales, *bowl-shaped*, *plate-shaped*, *valley-shaped*, *steep ridges* y otras conocidas funciones de optimización como *styblinski-tang* y *beale function*. Los resultados muestran que el método AiNet (Castro et al., 2002) es más robusto que el método BFOA (Passino, 2010) para los casos de estudio considerados en este trabajo. Sin embargo, existen diferencias en la eficiencia (número de funciones evaluadas y tiempo de convergencia) entre ambos métodos. Donde BFOA es el algoritmo con mejor desempeño en cuanto al número de funciones evaluadas.

© 2016 Universidad Nacional Autónoma de México, Facultad de Ingeniería. Este es un artículo Open Access bajo la licencia CC BY-NC-ND (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Introducción

Los métodos de búsqueda inspirados en procesos naturales se consideran confiables y adecuados para resolver diversos problemas de optimización que se caracterizan por tener funciones objetivo altamente no lineales y no convexas. De forma particular, en el área de ingenierías existe diversas funciones conocidas como benchmark con las que se pretende representar la complejidad de problemas reales. Estas funciones se emplean para el estudio y comparación del desempeño de métodos de optimización (Kämpf et al., 2010). En este contexto, varios estudios han demostrado que los métodos inspirados en procesos naturales, son robustos, fáciles de implementar y de aplicación general para la resolución de problemas en diversos contextos de la aplicación de meta heurísticas (Yang, 2010). Específicamente los métodos basados en optimización según la actividad celular emplean la estrategia de reproducción o clonación de aquellas que ofrecen mejores resultados en la búsqueda. Esta es una característica en común con el conocido algoritmo genético de Holland (1975).

En cuanto a las funciones benchmark, estas tienen diferentes complejidades y suelen ser multimodales, multidimensionales, y con un número n de diferentes mínimos locales. Cada uno de los espacios de búsqueda representa un problema particular para el algoritmo de optimización (Molga et al., 2005). Hasta ahora se han desarrollado diferentes algoritmos inspirados en la actividad celular, como por ejemplo los basados en el sistema inmune de los organismos, que se encargan de la detección y eliminación de potenciales amenazas externas, así como aquellos algoritmos inspirados en la actividad bacterial en búsqueda de nutrientes. También las conocidas redes neuronales son sistemas artificiales inspirados en las células del sistema nervioso, conocidas como neuronas, pero es principalmente explotado en el área de reconocimiento de patrones y aprendizaje (Brownlee, 2011). Con base en lo anterior, en este trabajo se reporta un comparativo de comportamiento numérico entre dos algoritmos inspirados en procesos naturales, particularmente la actividad celular. Estos algoritmos son:

a) Bacterial Foraging Optimization Algorithm (Passino, 2010) e
b) Immune Network Algorithm (Castro et al., 2002).

La comparación de estos métodos se realizó empleando el concepto de perfiles numéricos propuesto por (Dolan et al., 2002) y recientemente empleado por (Bonilla et al., 2011) con el objeto de establecer las diferencias relativas entre los algoritmos en términos de robustez (capacidad de alcanzar el óptimo global), eficiencia (esfuerzo numérico requerido durante la secuencia de optimización).

Trabajos previos y justificación


*Adaptaciones y mejoras:* Por ejemplo, en Dasgupta (2009a) se diseñó una versión de BFOA adaptativo con una estrategia para acelerar la convergencia al óptimo. Señalan que este algoritmo puede quedar oscilante cuando está muy cerca del objetivo. Proponen un operador que cambia dinámicamente los pasos de quimiotorax según los valores de *fitness* obtenidos. En Abraham (2008) se analiza la etapa de reproducción de BFOA como determinante en la convergencia a través de dos ecuaciones diferenciales y en un escenario de dos bacterias en un plano. En Dasgupta (2009b) se propone una versión BFOA donde la mejor bacteria se mantiene intacta (elitismo) mientras las otras, que son unas cuantas, se reinician. En Borovska (2011) se hace una paralelización de BFOA aplicado al problema de planeación de tareas.


*Comparativas de desempeño:* en Jati (2012); Rout (2013); Mezura (2009); Baijal (2011) y Mezura (2008) se hace una comparativa del desempeño de BFOA con otros algoritmos de optimización y en el contexto de diferentes problemas.


En el área optimización es frecuente hacer comparativas de desempeño entre algoritmos en problemas de interés, una de las prácticas comunes es trabajar con promedios de repeticiones de forma que faciliten la comparativa, y en algunos casos se realizan pruebas estadísticas como la conocida prueba *t* para contrastar promedios. Sin embargo, este procedimiento no permite observar en términos globales el desempeño de un conjunto de algoritmos a la luz de un conjunto de problemas. Ante esto existen herramientas como el perfil de comportamiento numérico. Este trabajo introduce la aplicación de esta herramienta como comparativa para estos algoritmos y se realiza para tres diferentes métricas. Se trata de un aporte para documentar y contrastar la eficiencia y eficacia de los algoritmos presentados.

**Métodos inspirados en la naturaleza**

Desde el comienzo de la humanidad, el hombre fue capaz de obtener recursos de la naturaleza, como alimentos y refugio que encontraba a su paso. Con el tiempo, fue posible comprender mejor las bases de diferentes procesos naturales hasta lograr cultivar alimentos, domesticar animales y construir refugios según las necesidades. Gradualmente, el conocimiento sobre los procesos naturales se ha incrementado, hasta los días modernos en que es posible “manejar la vida” en diferentes niveles, por ejemplo, crear alimentos transgénicos, combatir enfermedades y cultivar cepas de organismos unicelulares (Castro et al., 2004).

Actualmente con los recursos de cómputo es posible observar la naturaleza como fuente de inspiración para
el desarrollo de técnicas de optimización y solución de problemas complejos en ingeniería y otros contextos. En el área de la computación inspirada en procesos naturales, existe la rama de computación inspirada en procesos biológicos que se refiere particularmente a procesos naturales de los organismos vivos. Entre los procesos naturales no biológicos existe por ejemplo el recocado de materiales que inspira el algoritmo de recocado simulado Kirkpatrick (1984).

En el área optimización existen modelos inspirados en procesos biológicos:

a) **Bacterial Foraging** (forrajeo de bacterias). Este modelo se inspira en la forma en que las colonias de bacterias son capaces de trasladarse hacía un punto en el que perciben mayor cantidad de nutrientes. Así mismo, son capaces de apartarse de toxinas y factores desfavorables para la colonia. Este es un modelo relacionado con el algoritmo de quimiotaxis bacterial de Muller (2002) así como otros basados en el modelo de enjambres. Las bacterias como *E.Coli* segregan sustancias que atraen y repelen a otras bacterias de la misma especie. La movilidad bacterial está determinada por el uso de flagelos que se mueven en dirección o contra-dirección de las manecillas del reloj. El movimiento de una bacteria se puede dar en forma de tumbos (tumble) o nado (swim), y esto se determina por su percepción del entorno, es decir, el gradiente de nutrientes/toxinas en conjunto con la interacción de otras bacterias (Brownlee, 2011). Este trabajo se centra en el algoritmo BFOA.

b) **Immune System** (sistema inmune). El área de sistemas inmunes artificiales (SIA) está relacionada con los métodos computacionales inspirados en el sistema inmuno-biológico de los organismos, principalmente el de los mamíferos, que permite la detección y eliminación de patógenos que significan un riesgo para el organismo en cuestión. Los patógenos pueden ser bacterias, virus, parásitos y polen. La identificación de patógenos se logra por diferenciación. El sistema inmune adquirido, también llamado adaptativo, es responsable de especializar la defensa para amenazas específicas. Este tipo de sistema de defensa está presente en organismos vertebrados, a diferencia del sistema inmune innato. Una característica del sistema inmune es la capacidad de retenir información de los patógenos que atacan al sistema, como estrategia para futuras apariciones del mismo patógeno. Las células conocidas como glóbulos blancos, son los principales actores en el sistema inmune, ya que están involucrados tanto en la identificación como en la eliminación de los patógenos (Brownlee, 2011). Los trabajos por incorporar a este modelo biológico en modelos algorítmicos enfocados a la optimización tienen orígenes en trabajos previos (Hoffmann, 1986 y Hofmeyr, 1999). Los SIA modernos están inspirados en algunas de las tres sub-áreas de estudio: clonal selection, negative selection o immune network. Estos modelos se emplean en problemas de optimización, clasificación y reconocimiento de patrones. Destaca el trabajo de Castro (2002) que es una introducción a inmunología con el grado de detalle para diseño de algoritmos. Este trabajo se centra en el algoritmo AiNet.

**Algoritmo BFOA**

La optimización por enjambre de bacterias o bacterial foraging optimization algorithm (BFOA) representa una aproximación diferente a la búsqueda de valores óptimos en funciones no lineales, desarrollado por Passino (2010), se basa en el comportamiento quimiotáctico de la bacteria *Escherichia Coli* (*E. Coli*). Si bien, utilizar la quimiotaxis como modelo para optimización se propuso por primera vez en Bremermann (1974) y se ha utilizado en trabajos de Leviská (2006), el trabajo de Passino incluye algunas modificaciones como la reproducción y la dispersión de los agentes. La *E.Coli* es quizás el microorganismo más comprendido, ya que su comportamiento y estructura genética están bien estudiados.

Esta consta de una cápsula que incluye sus órganos y flagelos, que utiliza para su locomoción; posee capacidad de reproducirse por división y también es capaz de intercambiar información genética con sus congéneres. Además, es capaz de detectar nutrientes y evitar sustancias nocivas, efectuando un tipo de búsqueda aleatoria, basado en dos estados de locomoción: el desplazamiento o nado (swim) y el giro o tumbo (tumble). La decisión de permanecer en alguno de estos dos estados se basa en la concentración de nutrientes o sustancias nocivas en el medio. Este comportamiento se denomina quimiotaxis. A continuación se describen los pasos de la optimización con el algoritmo BFOA (Regis et al., 2011).

**Paso 1**: El ciclo de la quimiotaxis se describe en la figura 1. En este proceso el movimiento de *E.Coli* se simula. El movimiento se efectúa de dos formas: dando tumbos o a nado, una operación a la vez. Se calcula el valor de la función objetivo. La bacteria cambia su posición si el valor de la función objetivo modificada es peor que la anterior. Al completar la quimiotaxis la bacteria estará rondando un punto en el espacio de búsqueda.

---

*Ingeniería Investigación y Tecnología*, volumen XVII (número 4), octubre-diciembre 2016: 479-490 ISSN 1405-7743 FI-UNAM
Figura 1. Proceso de Quimiotaxis: Las células o bacterias realizan movimiento en forma de nado (swim) o tumbos (tumble) como estrategia para su movilización en busca de nutrientes o apartarse de condiciones desfavorables.

**Paso 2:** El proceso de reproducción. El valor de la función objetivo se calcula para cada una de las bacterias en la población ordenada. La peor mitad de la población se descarta y la mejor mitad se duplica. Para esta nueva generación de bacterias el ciclo de quimiotaxis se inicia y el proceso continúa por el número de pasos de reproducción.

**Paso 3:** En el ciclo de eliminación-dispersión, algunas bacterias se eliminan con baja probabilidad y se dispersan en un punto aleatorio del espacio de búsqueda. Este proceso mantiene constante el número de bacterias.

Los parámetros de entrada son el número de bacterias $S$, límite de pasos de quimiotaxis $N_a$, límite de pasos de nado $N_n$, límite de ciclos de reproducción $N_r$, número de bacterias a producir $S$, límite del ciclo de eliminación-dispersión $N_d$, tamaño de paso $C$, y probabilidad de eliminación-dispersión $P_d$. El costo de cada bacteria se optimiza por su interacción con otras bacterias. La función de interacción se calcula de acuerdo con la expresión $g()$.

$$g(cell_i) = \sum_{i=1}^{S} [-d_{attr} \cdot e^{-w_{attr} \cdot \sum_{i=1}^{m}(cell_{i} - other_{i})^2} ] + \sum_{i=1}^{S} [h_{repel} \cdot e^{-w_{repel} \cdot \sum_{i=1}^{m}(cell_{i} - other_{i})^2} ]$$

donde

- $cell_i$ = bacteria en cuestión
- $d_{attr}$ y $w_{attr}$ = coeficientes de atracción
- $h_{repel}$ y $w_{repel}$ = coeficientes de repulsión
- $S$ = número de bacterias en la población
- $P$ = número de dimensiones o variables a optimizar en cada bacteria

La representación “other” es “otra célula” interactuando con $cell_i$.

**Algoritmo AiNet**

Recientemente el algoritmo de optimización basado en el sistema inmune (Castro y Timmis 2002) ha llamado la atención de los investigadores por su potente capacidad de manejo de información. El sistema inmune natural es complejo y con diferentes mecanismos de defensa contra organismos invasores, tiene características como: especificidad, reconocimiento de patrones, diversidad, tolerancia a fallas, memoria y aprendizaje, auto-organización, cooperación entre capas, entre otros.

El objetivo de una red inmune es el de preparar o construir un repertorio de detectores para un problema dado, donde las células con mejor desempeño suprimen a aquellas con baja afinidad en la red. Este objetivo se alcanza exponiendo a la población a información externa, a partir de la cual se generan respuestas en forma de clonaciones y dinámica iter-celular.

Las dos teorías que principalmente motivaron el desarrollo de este modelo de optimización son (Aragón et al., 2010): *Immune Network Theory* (IN) y *Clonal Selection Theory* (CS). El principio de CS es la teoría que se emplea para describir la respuesta adaptativa del sistema inmune hacia estímulos antígenicos. La premisa de este modelo establece que el sistema inmune reacciona únicamente cuando se invade por estímulos externos. Así que la respuesta del sistema es la producción de anticuerpos por las células B una vez que el antígeno ha entrado al sistema. Entonces en la presencia del antígeno, aquellos anticuerpos con mayor afinidad o capacidad de reconocer el antígeno son los privilegiados para proliferar, produciendo enormes cantidades de anticuerpos clones con diversidad.
Parámetros y recursos

Los algoritmos BFOA y AiNet se desarrollaron en lenguaje Ruby, en equipo de cómputo con procesador Intel core i7, 10Gb en memoria RAM y sistema operativo Windows 7. Las gráficas se generaron en el software Excel de Microsoft. En cuanto a los parámetros de los algoritmos, se adoptaron los considerados default, reportados por Brownlee (2011).

Determinación de los perfiles numéricos

La comparación del comportamiento numérico (robustez, eficiencia) de los métodos BFOA y AiNet se realizó al emplear el concepto de perfil de comportamiento numérico. El perfil de comportamiento para un método de optimización se define como la función de distribución acumulativa para una métrica de comportamiento o desempeño numérico (Dolan, 2002). Dicha métrica puede corresponder a algún factor de interés en la comparativa de desempeño, como el tiempo necesario para alcanzar la convergencia del método de optimización, la cantidad de funciones evaluadas durante la secuencia de cálculo o la capacidad del método para localizar al óptimo global de la función objetivo. En este trabajo, las siguientes métricas se consideran para la comparación de los dos métodos: la distancia relativa entre el óptimo localizado por el método de optimización y el óptimo global conocido: $d$. La cantidad de funciones evaluadas durante la secuencia de optimización (NFE) y el tiempo de ejecución (T) para la convergencia. La primera métrica se asoció con la robustez del método (es decir, la capacidad de la estrategia numérica para localizar al óptimo global de la función objetivo) mientras que la segunda y tercera corresponden a una medida de la eficiencia de los métodos estocásticos (Ali et al., 2005).

Para la evaluación de estas métricas, se asume que existen $n_1 = 2$ métodos de optimización y $n_p = 18$ problemas o casos de estudio. Cada uno de los casos de estudio se resolvió en 30 ocasiones con estimaciones iniciales aleatorias y diferentes secuencias de números aleatorios, considerando una tolerancia de 1.0E-06 en el valor de la función objetivo como criterio de convergencia para ambos métodos. Para cada problema y método de optimización, las métricas $t_{p,s}$ se calcularon empleando los resultados de 30 cálculos y las siguientes expresiones.

\[
\begin{align*}
t_{p,s}^d &= \frac{\hat{f}_{obj}}{\hat{f}_{obj}^{w}} - \frac{\hat{f}_{obj}}{\hat{f}_{obj}^{w}} = NFE \quad t_{p,s}^T = T
\end{align*}
\]

donde

\[
\begin{align*}
\hat{f}_{obj}^{w} &= \text{valor promedio de la función objetivo calculado por el método de optimización} \\
\hat{f}_{obj} &= \text{óptimo global de la función objetivo} \\
\hat{f}_{obj}^{w} &= \text{valor máximo para la función objetivo encontrado dentro la secuencia de cálculo} \\
NFE \text{ y } T &= \text{valor promedio del número de funciones evaluadas y el tiempo para alcanzar la convergencia del método de optimización}
\end{align*}
\]
### Tabla 1. Conjunto de funciones Benchmark

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Función</th>
<th>var</th>
<th>Espacio de Búsqueda</th>
<th>Mínimo Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross in Tray</td>
<td>$f(x) = -0.0001 \left( \sin x_1 \sin x_2 \sqrt{\frac{x_1^2 + x_2^2}{2}} + 1 \right)^{0.1}$</td>
<td>$x_i \in [-10, 10]$</td>
<td>-2.06261</td>
<td></td>
</tr>
<tr>
<td>Drop Wave</td>
<td>$f(x) = -\frac{1 + \cos(12 \sqrt{x_1^2 + x_2^2} / 0.5(x_1^2 + x_2^2) + 2)}{1 + 10^(-</td>
<td>x_1</td>
<td>+</td>
<td>x_2</td>
</tr>
<tr>
<td>Holder Table</td>
<td>$f(x) = -\sqrt{x_1 x_2} \cos(x_1) \cos(x_2) \exp\left(\frac{x_1^2 + x_2^2}{2}\right)$</td>
<td>$x_i \in [-10, 10]$</td>
<td>-19.2085</td>
<td></td>
</tr>
<tr>
<td>EggHolder</td>
<td>$f(x) = -\left(x_1 + 47 \right) \sin \left(\sqrt{x_1^2 + \frac{x_2^2}{2}} + \frac{47}{2}\right) - \sin \left(\sqrt{x_1^2 - (x_2 + 47)^2}/2\right)$</td>
<td>$x_i \in [-512, 512]$</td>
<td>-959.6407</td>
<td></td>
</tr>
<tr>
<td>Rastrigin</td>
<td>$f(x) = 10d + \frac{1}{2} \sum_{i=1}^{d} x_i^2 - 10 \cos x_i \cos \left(\frac{x_i}{\sqrt{d}}\right)$</td>
<td>$x_i \in [-5.12, 5.12]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Shubert</td>
<td>$f(x) = \left(\sum_{i=1}^{d} \cos(i+1)x_i + 1\right) \left(\sum_{i=1}^{d} \cos(i+1)x_i + 1\right)$</td>
<td>$x_i \in [-10, 10]$</td>
<td>-186.7309</td>
<td></td>
</tr>
<tr>
<td>Schwefel</td>
<td>$f(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin \sqrt{</td>
<td>x_i</td>
<td>}$</td>
<td>$x_i \in [-500, 500]$</td>
</tr>
<tr>
<td>Schaffer2</td>
<td>$f(x) = 0.5 \left( \sin^2 \left( x_1 - x_2 \right) - 0.5 \frac{x_1^2 + x_2^2}{1 + 0.001(x_1^2 + x_2^2)} \right)$</td>
<td>$a = 20$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Ackley</td>
<td>$f(x) = \frac{1}{1 + e^{-a x^T b}} - e^{c x^T a} + a + e$</td>
<td>$b = 0.2$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Booth</td>
<td>$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$</td>
<td>$x_i \in [-10, 10]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Matyas</td>
<td>$f(x) = 0.2(x_1^2 + x_2^2) - 0.4x_1x_2$</td>
<td>$x_i \in [-10, 10]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Zakharov</td>
<td>$f(x) = \sum_{i=1}^{d} \left( 100(x_i^* - x_i)^2 + (x_i^* - 1)^2 \right)$</td>
<td>$x_i \in [-5, 10]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sphere</td>
<td>$f(x) = \sum_{i=1}^{d} x_i^2$</td>
<td>$x_i \in [-5.12, 5.12]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Rosenbrok</td>
<td>$f(x) = \sum_{i=1}^{d} \left[ 100(x_i - x_i^<em>)^2 + (1 - x_i^</em>)^2 \right]$</td>
<td>$x_i \in [-5, 10]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Michaelwicz</td>
<td>$f(x) = -\sum_{i=1}^{d} \sin(x_i) \sin\left(\frac{m</td>
<td>x_i</td>
<td>^m}{\delta} \right)$</td>
<td>$m = 10$</td>
</tr>
<tr>
<td>Easom</td>
<td>$f(x) = -\cos(x_1)\cos(x_2) e^{-\left( x_1^2 + x_2^2 \right)}$</td>
<td>$x_i \in [-100, 100]$</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Beale</td>
<td>$f(x) = (x_1 - 4.5)^2 + 10 \left[ 1 - \cos \left( \frac{\sqrt{x_1^2 + x_2^2}}{\sqrt{2}} \right) \right]$</td>
<td>$x_i \in [-4.5, 4.5]$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Styblinski-Tang</td>
<td>$f(x) = \frac{1}{m} \sum_{i=1}^{m} \left( x_i^2 - 10 \sin^2 \left( \frac{x_i}{\sqrt{2}} \right) \right)$</td>
<td>$x_i \in [-5, 5]$</td>
<td>$-39,16599$</td>
<td></td>
</tr>
</tbody>
</table>
Tabla 2. Parámetros de los algoritmos BFOA y AiNet

<table>
<thead>
<tr>
<th>Parámetro</th>
<th>Valor</th>
<th>Parámetro</th>
<th>Valor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algoritmo BFOA</td>
<td></td>
<td>Algoritmo AiNet</td>
<td></td>
</tr>
<tr>
<td>Población</td>
<td>50</td>
<td>Límite de Generaciones</td>
<td>150</td>
</tr>
<tr>
<td>Step size</td>
<td>0.1</td>
<td>Población</td>
<td>20</td>
</tr>
<tr>
<td>Ciclos de eliminación-dispersión (Ned)</td>
<td>1</td>
<td>Número de Clones</td>
<td>10</td>
</tr>
<tr>
<td>Ciclos de reproducción (Nre)</td>
<td>4</td>
<td>Beta</td>
<td>100</td>
</tr>
<tr>
<td>Ciclos de quimiotaxis (Nc)</td>
<td>70</td>
<td>Números aleatorios</td>
<td>2</td>
</tr>
<tr>
<td>Longitud de nado (Ns)</td>
<td>4</td>
<td>Umbral de afinidad</td>
<td>0.05 del espacio de búsqueda</td>
</tr>
</tbody>
</table>

Es importante señalar que los valores de \( \hat{f}_{obj}, NFE \) y \( \hat{T} \) se determinaron empleando los 30 experimentos numéricos realizados para el caso de estudio. Conforme a lo establecido por Ali et al. (2005) y Bonilla et al. (2011), en la literatura se suelen usar valores promedio para las métricas de comportamiento con el objeto de describir el desempeño de los métodos. Con base en lo anterior, este estudio también emplea dicho enfoque. Por otro lado, para las tres métricas, la tasa de comportamiento numérico \( r_{ps} \) se define como

\[
r_{ps} = \frac{t_{ps}}{\min(t_{ps}, s \in S)}
\]

Donde S corresponde al conjunto de métodos de optimización analizados. Se puede observar que el valor de dicha tasa es igual a 1 para el método que presenta el mejor comportamiento en un problema específico, ya que para ambas métricas es deseable obtener el valor mínimo posible (Dolan, 2002 y Ali et al., 2005). Finalmente, la tasa de probabilidad acumulativa \( \rho_s(\tau) \) para el método de optimización \( s \) y la métrica en cuestión se define como

\[
\rho_s(\tau) = \frac{1}{n_p} \text{cantidad } \{ p \in P : r_{ps} \leq \tau \}
\]

Donde \( \tau \) es un factor que se define en \((1, \infty)\). La gráfica del perfil de comportamiento, es decir, la gráfica de \( \rho_s(\tau) \), versus \( \tau \), compara el desempeño relativo entre los métodos de optimización para el grupo de problemas considerados (Dolan, 2002). Hasta el momento los perfiles de comportamiento se han utilizado por Montaz et al. (2005) empleando funciones objetivo clásicas del área de optimización, y por Bonilla et al. (2011) empleando funciones del área de la ingeniería química. No obstante, dicho concepto no se ha empleado en la comparativa de los métodos descritos en este estudio.

**Resultados y discusión**

El perfil de comportamiento para la métrica \( t_{ps, t}^{eff} \), que se asocia a la capacidad del método de optimización para acercarse al óptimo global en los problemas considerados, se muestra en la figura 2. Como se puede observar, el método AiNet presenta un mejor comportamiento para esta métrica, en contraste con el método BFOA dentro del rango analizado para \( t \). También estos resultados indican que en 81% de los casos de estudio el método AiNet proporciona mejor solución (\( \tau = 1 \)), mientras que BFOA solamente lo consigue en 19% de los casos.

Para el caso de la eficiencia, en la métrica \( NFE \) es indudable que el método BFOA supera al método AiNet (figura 3) para todos los casos de estudio.

Para el caso de la eficiencia, en la métrica del tiempo empleado por el método de optimización \( T \), se puede observar en la figura 4 que el método AiNet tuvo un mejor desempeño en contraste con BFOA.

Con el objeto de proporcionar más elementos para el comparativo de estos métodos de optimización, en la tabla 3 se muestran los valores de \( f_{obj}, NFE \) y \( T \) para las dos estrategias de optimización en todos los casos de estudio considerados.
Los parámetros de atracción y repulsión en el caso del algoritmo BFOA son determinantes y para este estudio todos los experimentos se realizaron con valores propuestos en la literatura, pero pueden ajustarse con base en el juicio (Dasgupta et al., 2009). En este sentido, para el caso de la métrica que corresponde al número de funciones evaluadas $NFE$, destaca el resultado del algoritmo AiNet para la función de optimización Easom, donde considerablemente el algoritmo realiza un esfuerzo computacional mayor que BFOA. De acuerdo con la literatura, el parámetro de umbral de similitud puede ajustarse para problemas específicos (Brownlee, 2011). En cuanto al tiempo empleado por el método de optimización, la métrica $T$ describe dicha característica para los casos de estudio. Destaca el algoritmo AiNet haciendo uso de menos tiempo en todos los experimentos excepto el correspondiente al de la función Easom, que como es de esperarse y en congruencia con la métrica $NFE$ presenta mayor tiempo de ejecución.

**Tabla 3. Promedios de las métricas a comparar para los algoritmos BFOA y AiNet**

<table>
<thead>
<tr>
<th>Núm</th>
<th>Óptimo</th>
<th>Benchmark</th>
<th>BFOA</th>
<th>AINE</th>
<th>BFOA</th>
<th>AINE</th>
<th>BFOA</th>
<th>AINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>Ackley</td>
<td>9.3442</td>
<td>4.3250</td>
<td>28616.8333</td>
<td>150853.1000</td>
<td>3764.5153</td>
<td>1829.0026</td>
</tr>
<tr>
<td>2</td>
<td>-1.8013</td>
<td>Michalewicz</td>
<td>-1.8012</td>
<td>-1.7999</td>
<td>29372.3667</td>
<td>75678.2000</td>
<td>3873.6549</td>
<td>1253.0574</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>Rastrigin</td>
<td>0.7107</td>
<td>0.2212</td>
<td>28588.8000</td>
<td>129441.1330</td>
<td>3762.2486</td>
<td>2043.7369</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>Rosenbrock</td>
<td>0.0007</td>
<td>0.0762</td>
<td>32261.7667</td>
<td>136271.8330</td>
<td>4275.7113</td>
<td>2134.2732</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>Schwefel</td>
<td>201.1314</td>
<td>79.9905</td>
<td>40398.7333</td>
<td>149165.7000</td>
<td>2474.1749</td>
<td>1192.5015</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>Sphere</td>
<td>0.0000</td>
<td>0.0000</td>
<td>38576.7667</td>
<td>76409.0333</td>
<td>4499.3907</td>
<td>1091.4825</td>
</tr>
<tr>
<td>7</td>
<td>-78.33198</td>
<td>Styblinski</td>
<td>-78.3323</td>
<td>-78.2564</td>
<td>32229.9000</td>
<td>108285.3670</td>
<td>1959.0027</td>
<td>1093.2016</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>Zakharov</td>
<td>0.0001</td>
<td>0.0033</td>
<td>38901.1333</td>
<td>100223.0000</td>
<td>5398.3088</td>
<td>1015.1348</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>Beale</td>
<td>0.0000</td>
<td>0.0026</td>
<td>31692.7000</td>
<td>10971.2000</td>
<td>5485.3804</td>
<td>1063.6682</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>Booth</td>
<td>0.0001</td>
<td>0.2168</td>
<td>37141.1333</td>
<td>128546.3000</td>
<td>4843.2103</td>
<td>1405.6686</td>
</tr>
<tr>
<td>11</td>
<td>-2.06361</td>
<td>Cross In Tray</td>
<td>-2.0626</td>
<td>-2.0527</td>
<td>43420.5667</td>
<td>144084.0670</td>
<td>5238.7663</td>
<td>1565.3355</td>
</tr>
<tr>
<td>12</td>
<td>-1</td>
<td>Drop Wave</td>
<td>-0.8708</td>
<td>-0.9426</td>
<td>33297.1667</td>
<td>150218.1330</td>
<td>4514.5249</td>
<td>1609.6689</td>
</tr>
<tr>
<td>13</td>
<td>-1</td>
<td>Easom</td>
<td>-0.0333</td>
<td>-0.0232</td>
<td>56335.3333</td>
<td>400736.3700</td>
<td>6101.3156</td>
<td>15623.0552</td>
</tr>
<tr>
<td>14</td>
<td>-959.6407</td>
<td>Egg Holder</td>
<td>-719.0541</td>
<td>-838.0015</td>
<td>40293.5667</td>
<td>151782.8670</td>
<td>5974.6418</td>
<td>1639.7738</td>
</tr>
<tr>
<td>15</td>
<td>-19.2085</td>
<td>Holder Table</td>
<td>-19.2081</td>
<td>-19.1766</td>
<td>35989.3333</td>
<td>114001.6000</td>
<td>4941.5826</td>
<td>1607.4575</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>Matyas</td>
<td>0.0000</td>
<td>0.0031</td>
<td>40177.0000</td>
<td>112702.1330</td>
<td>5371.4072</td>
<td>1679.3944</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>Schaffer 2</td>
<td>0.0023</td>
<td>0.0002</td>
<td>26183.6000</td>
<td>146886.2330</td>
<td>3376.0931</td>
<td>2373.0716</td>
</tr>
<tr>
<td>18</td>
<td>-186.7309</td>
<td>Shubert</td>
<td>-178.8117</td>
<td>-186.5650</td>
<td>28487.7667</td>
<td>146344.9670</td>
<td>3935.1918</td>
<td>2535.8996</td>
</tr>
</tbody>
</table>
Conclusiones

Este trabajo describe la aplicación del método BFOA y AiNet en comparativa, empleando el modelo de perfil de comportamiento numérico (Dolan, 2002) para un conjunto de conocidas funciones benchmark. Los resultados obtenidos indican que el método AiNet (Castro, 2002) es más robusto que el método BFOA (Passino, 2010) para los casos de estudio considerados en este trabajo. Sin embargo, existen diferencias en la eficiencia (número de funciones evaluadas y tiempo de convergencia) entre ambos métodos. Donde BFOA es el algoritmo con mejor desempeño en cuanto al número de funciones evaluadas.

Con base en los resultados obtenidos, para futuras aplicaciones de los algoritmos aquí descritos se hacen los siguientes contrastes:

El algoritmo AiNet tuvo mejor desempeño en términos globales en comparación con BFOA, pero debe señalarse el caso de la función benchmark Easom, en la que AiNet tuvo particularmente dificultades. Esto aparentemente se debe a que la naturaleza de su estrategia basada en clonación puede ser ineficiente en la región plana del espacio de búsqueda, donde el algoritmo “parea perderse”, se recomienda observar la gráfica del espacio de búsqueda Easom. Esto debe tomarse en cuenta para futuras aplicaciones.

En el caso de la función benchmark Schwefel, los dos algoritmos tuvieron resultados alejados del óptimo esperado, donde BFOA fue el peor. Al observar la gráfica del espacio de búsqueda destacan los múltiples mínimos locales. Tal parece que ambos algoritmos tienen dificultades en un escenario de tal complejidad. Se recomienda hacer una sintonización no basada en los default para el parámetro de umbral de afinidad respecto a AiNet y los parámetros de atracción/repulsión (d_attr, w_attr, h_rep, w_rep) respecto a BFOA.

Referencias


Ali E.S.E. Hybrid BFOA-PSO approach for SSSC damping controller design, Control, Decision and Information Technologies (CoDIT), International Conference, 2013, IEEE, pp. 464-469.


Perfiles de comportamiento numérico de los métodos de búsqueda immune network algorithm y bacterial foraging optimization algorithm en funciones benchmark

Wei P. y George M.C. A fast opt-AInet approach to qualitative model learning with modified mutation operator, Proceedings of UKCI 2011, pp. 43-54.

Este artículo se cita:
**Citación estilo Chicago**

**Citación estilo ISO 690**

Semblanzas de los autores

**Ernesto Ríos-Willars.** Es participante del programa de doctorado en Biotecnología de la Universidad Autónoma de Coahuila. Su trabajo se refiere al área de optimización y metaheurísticas en bioinformática.

**Ernesto Liñán-García.** Doctorado en ciencias computacionales, maestro en sistemas computacionales e ingeniero en sistemas electrónicos. Su línea de investigación es sobre la aplicación de metaheurísticas aplicadas a problemas NP-Hard, especialmente plegamiento de proteínas, alineamiento de secuencias ADN y ruteo de vehículos.

**Rafael Batres.** Profesor investigador en la Escuela de Ingeniería y Ciencias del Tecnológico de Monterrey. Las áreas de investigación incluyen: métodos computacionales para el diseño de productos, síntesis de procesos, cadenas de suministro, ingeniería de ontologías, razonamiento basado en casos, optimización meta-heurística, reutilización de modelos matemáticos, minería de datos, modelado basado en agentes y simulación dinámica.

**Yolanda Garza-García.** Doctora en ciencias biológicas por la Universidad Autónoma de Nuevo León. Es investigadora y jefe del departamento de Biotecnología de la Universidad Autónoma de Coahuila.