

Algoritmo de optimización para el cálculo de múltiples raíces de sistemas de ecuaciones no lineales¹

Artículo de Investigación Científica - Fecha de recepción: 26 de febrero de 2013 - Fecha de aceptación: 16 de mayo de 2013

Jorge Cruz Duarte

Ingeniero Electrónico. Estudiante de Maestría de Investigación en Ingeniería Electrónica. Universidad Industrial de Santander. Bucaramanga, Colombia. mrcrois@gmail.com

Iván Amaya Contreras

Ingeniero Mecatrónico. Candidato a Doctor en Ingeniería, área Ingeniería Electrónica. Universidad Industrial de Santander. Bucaramanga, Colombia. iamaya2@gmail.com

Carlos Correa Cely

Ingeniero Químico. Doctor en Polymer Science and Engineering. Profesor Titular. Universidad Industrial de Santander, Bucaramanga, Colombia. crcorrea@uis.edu.co

Para citar este artículo / to reference this article:

J. Cruz, I. Amaya and C. Correa, "Algoritmo de optimización para el cálculo de múltiples raíces de sistemas de ecuaciones no lineales". *INGE CUC*, vol. 9, no. 1, pp. 197-208, Jun, 2013.

RESUMEN

En este artículo se presenta una estrategia novedosa para la búsqueda de múltiples soluciones en sistemas de ecuaciones no lineales, mediante la metaheurística de optimización por enjambre de partículas, y su versión unificada. Inicialmente, se expone una sección de fundamentos, donde se muestran las bases de las técnicas numéricas utilizadas y también la transformación del problema de solución en uno de optimización. Se comprueba esta nueva estrategia con diferentes tipos de sistemas, variando sistemáticamente el tamaño de los mismos para observar el comportamiento de las metaheurísticas usadas. Luego de analizar los datos obtenidos, se encuentra que la estrategia implementada tiene un gran potencial de aplicación en problemas de ingeniería. Para los sistemas analizados en este escrito, se recomienda utilizar la versión unificada o la original, dependiendo de si el sistema es pequeño o grande (menor o mayor a cinco ecuaciones e incógnitas), respectivamente.

Palabras clave

Computación, ecuaciones no lineales, enjambre de partículas, enjambre de partículas unificado, estrategia de solución, metaheurísticas, métodos numéricos, optimización, sistemas de ecuaciones.

1 Artículo derivado del proyecto de investigación interno titulado: *Fortalecimiento de la línea de investigación en optimización*. Grupo de Investigación CEMOS, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Facultad de Ingenierías Fisicomecánicas. Universidad Industrial de Santander.

Optimization algorithm for Finding Multiple Roots of Systems of Nonlinear Equations

ABSTRACT

This article presents a novel strategy for the search of multiple roots in a system of nonlinear equations. This procedure is carried out through the metaheuristics of particle swarm optimization and its unified version. At the beginning, a section regarding fundamentals is shown, where the basis of the numeric strategies, as well as the transformation of the problem of solving nonlinear equations, into an optimization one, are commented. This novel strategy is verified through several types of systems, systematically varying their sizes, in order to verify the behavior of the previously mentioned algorithms. After analyzing the data, it was found that the strategy holds a great applicability in engineering, and for the cases analyzed it is recommended to use the unified or the original algorithms, depending on whether the target system is a small or a big one, respectively (i.e. less or more than five equations and unknowns).

Keywords

Computing, Nonlinear equations, Particle swarm, Unified particle swarm, Solution strategy, Metaheuristics, Numerical methods, Optimization, Systems of equations.

INTRODUCCIÓN

El análisis de sistemas físicos modelados con ecuaciones es común en diferentes disciplinas del conocimiento. Existen variadas técnicas para solucionar cualquier tipo de conjunto de ecuaciones, tanto lineales como no lineales, que generalmente involucran un proceso de búsqueda de las raíces del sistema. En la práctica, es común utilizar métodos determinísticos, aunque con la actual evolución de la computación numérica y la creciente mejora en el procesamiento de datos, han aparecido otras alternativas, entre las que se incluyen los métodos estocásticos. En este artículo se muestran las mejoras realizadas para encontrar, no solo las raíces reales del sistema, sino también las complejas, usando las técnicas de optimización por enjambres PSO y UPSO [1]-[4]. Los resultados que aquí se muestran pueden llegar a tener alto impacto, especialmente a nivel científico, ya que pueden brindar una alternativa para solucionar modelos matemáticos más complejos, de una forma más eficiente (es decir, con menos recursos computacionales).

FUNDAMENTOS

En esta sección se presentan conceptos básicos alrededor de la investigación desarrollada, entre los que cabe destacar la estrategia de optimización implementada. Se define un conjunto de ecuaciones como en (1), donde cada una puede ser lineal o no. Dicho sistema, de dimensiones, posee al menos una solución, bien sea real o compleja o no tener ninguna. A través del *Teorema de Raíces Reales* [5] se puede tomar el sistema (1) y construir

la función objetivo (2), que corresponde a un problema de minimización, cuyos óptimos globales equivalen a la solución del sistema de ecuaciones original. También se pueden utilizar otras estrategias con un comportamiento similar, como se observa en [4].

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (1)$$

$$f_{\text{Obj}}(x_1, x_2, \dots, x_n) = \sum_{k=1}^n \|f_k(x_1, x_2, \dots, x_n)\|^2 \quad (2)$$

Esta transformación ha sido utilizada para la búsqueda de raíces reales [1]-[4], [6], [7]. No obstante, es posible definirla en el dominio complejo usando la magnitud de cada función evaluada como términos de la suma, $\|f_k(x_1, x_2, \dots, x_n)\|$, tal y como se aprecia en (2).

A lo largo de esta investigación se utilizaron las estrategias de *Optimización por Enjambre de Partículas* (PSO, de sus siglas en inglés) y de su versión unificada, *Optimización por Enjambre de Partículas Unificada* (UPSO, de sus siglas en inglés), ambas enfocadas a la minimización. El método base de optimización por enjambre de partículas fue propuesto por Eberhart y Kennedy en 1995 [8], y se define completamente con (3), que representa el cálculo de la nueva posición de una partícula, X , y con (4), que se refiere a la actualización de su velocidad, V . Estas expresiones son funcionales en sistemas de una o N dimensiones.

$$X_p^{t+1} = X_p^t + V_p^{t+1} \quad (3)$$

$$V_p^{t+1} = wV_p^t + C_1R_1(P_p - X_p^t) + C_2R_2(P_g - X_p^t) \quad (4)$$

La ecuación (4) indica que el nuevo valor de la velocidad de una partícula, p , depende de la información de ésta en el paso actual, t . Esta información se compone de: w , conocido como el factor inercial (éste limita la influencia de velocidades anteriores). C_1 y C_2 , conocidos como factores de confianza propia y del enjambre, respectivamente. R_1 y R_2 , definidos como números aleatorios entre cero y uno (uniformemente distribuidos). P_p , es la mejor posición encontrada por cada partícula y P_g , es la mejor posición encontrada en todo el enjambre hasta el paso actual. Una variante de PSO, bastante usada en la literatura [9], [10], es la propuesta por Parsopoulos y Vrahatis en 2005 [11], que se conoce como UPSO y que presenta dos grandes modificaciones al algoritmo base. La primera se observa en (5) y (6), y consiste en definir dos tipos de velocidades similares a la usada por PSO, donde G y L son llamadas velocidad global y local, referidas al movimiento de una partícula con respecto al enjambre total y a la vecindad a la cual esta pertenece, respectivamente. χ es el factor de constricción, el cual no permite a las partículas abandonar rápidamente el dominio de búsqueda definido. V es la velocidad total del paso anterior. R_1 , R_2 , R_3 y R_4 son números aleatorios entre cero y uno, similar a PSO. P_{g_p} es la mejor posición local encontrada en cada vecindad o subconjunto de partículas. Los otros parámetros mantienen el mismo sentido que en PSO.

$$G_p^{t+1} = \chi \cdot [V_p^t + C_1R_1(P_p - X_p^t) + C_2R_2(P_g - X_p^t)] \quad (5)$$

$$L_p^{t+1} = \chi \cdot [V_p^t + C_1R_3(P_p - X_p^t) + C_2R_4(P_{g_p} - X_p^t)] \quad (6)$$

La segunda modificación aporta la característica de unificación, y consiste en calcular la velocidad total, V , a partir de la global y la local, como muestra (7). El factor de unificación, u , puede tomar valores entre cero y uno, para así balancear las dos contribuciones. La ecuación (8) es equivalente a la (3), propia de PSO.

$$V_p^{t+1} = (1 - u)L_p^{t+1} + uG_p^{t+1} \quad (7)$$

$$X_p^{t+1} = X_p^t + V_p^{t+1} \quad (8)$$

Usando de (3) a (8), se puede describir un algoritmo base, que pueda ser usado por ambos métodos, como se muestra a continuación.

Algoritmo base de PSO o UPSO

1. Asignar una posición inicial aleatoria entre $X_{mín}^0$ y $X_{máx}^0$ (es decir, el dominio de búsqueda), y un valor de cero para la velocidad inicial de cada partícula.
2. Evaluar f_{Obj} y encontrar P_p y P_g (y P_{g_p} adicionalmente para UPSO).
3. Actualizar la posición, X , y la velocidad, V , para cada partícula usando (3) y (4) en PSO o (5)-(8) en UPSO.
4. Evaluar la función objetivo f_{Obj} usando la posición actual para cada partícula.

5. Comparar, para cada partícula, el valor de la función objetivo evaluada con P_p . Si esta es menor, entonces actualizar P_p .
6. (Sólo para UPSO) Seleccionar la mejor posición local para cada vecindad y comparar ésta con cada P_{gp} . Si es menor, entonces actualizar P_{gp} .
7. Seleccionar la mejor posición global y comparar con P_g . Si ésta es menor, entonces actualizar P_g .
8. Evaluar P_g en el criterio de convergencia establecido. Si no lo cumple, retornar a 3.

Adicionalmente, se propone una Estrategia de Búsqueda de Raíces Reales o Complejas, la cual tiene como objetivo principal encontrar más de una raíz (solución) al sistema de ecuaciones, de la siguiente forma:

Estrategia de Búsqueda de Raíces Reales o Complejas con PSO o UPSO

1. Ejecutar el Algoritmo Base de PSO o UPSO para encontrar una raíz.
2. Comparar la solución encontrada con las raíces almacenadas en memoria.
 - 2.1. Si esta es diferente a todas, entonces almacenarla en memoria.
 - 3.2. Si esta es aproximadamente igual a otra raíz encontrada cumpliendo una precisión definida, conservar la mejor entre ambas.
4. Evaluar el criterio de saturación. Si no se ha cumplido, retornar a 1.

El criterio de saturación mencionado anteriormente se estableció como el número máximo de ejecuciones del *Algoritmo Base* sin obtener una nueva raíz, o sin mejorar una existente.

RESULTADOS

Para utilizar los métodos de optimización por enjambre de partículas, se definen los parámetros C_1 y C_2 iguales a 2, $w = 0,6$, $\chi = 0,6$ y $u = 0,5$. El criterio de convergencia se fijó en el orden de 1×10^{-10} y el tamaño del enjambre de 100 partículas. Estos valores se encontraron de forma empírica en el desarrollo de esta investigación, y fueron contrastados con algunas publicaciones que los describen [1]-[3], [11]-[13], encontrando similitud entre ellos. Para iniciar la búsqueda en el dominio complejo, se establece la distribución inicial entre $-10-10i$ y $10+10i$. Los algoritmos de estudio se construyeron como scripts bajo el entorno de cálculo numérico, MATLAB^{MR}, instalado en Microsoft[®] Windows^{MR} 7, Home Premium 64-bit, con un hardware Toshiba[®], Satellite^{MR} A665-S6092, Intel[®] Core^{MR} i7 Q740@1,73GHz, 6GB RAM.

La primera función objetivo se construye a partir de un polinomio de quinto grado, como muestra (9).

Los coeficientes se seleccionaron arbitrariamente, usando los seis primeros números primos, incluyendo el número uno, todos con signo positivo para garantizar diversidad en las soluciones.

$$f_1 = x^5 + 2x^4 + 3x^3 + 5x^2 + 7x + 11 = 0 \quad (9)$$

Se ejecutan cuatro repeticiones de la *Estrategia de Búsqueda de Raíces Reales y Complejas* usando PSO y UPSO con la función objetivo construida a partir de (9). Una forma de probar la cercanía de las raíces encontradas consiste en usar el paquete básico de solución de polinomios, *roots*, presente en MATLAB^{MR}.

ANÁLISIS DE RESULTADOS

En la Tabla I se observa que ambos métodos hallan cuatro raíces complejas, y

una real, como soluciones al sistema. La Tabla II muestra la información correspondiente a cada repetición realizada tanto para PSO como para UPSO. Es fácil apreciar que el primero, para este caso, es computacionalmente costoso comparado con el segundo. Lo anterior se sustenta en que evaluó, aproximadamente, 2,8 veces más la función objetivo. Consecuentemente, el tiempo de convergencia de UPSO es menor que el requerido por el método original (en promedio un 51% y 81% en el peor y mejor de los casos, respectivamente).

TABLA I. RAÍCES OBTENIDAS PARA EL SISTEMA 1D (9).

PSO	UPSO	Software numérico
-1,7653-1,7e-7i	-1,7653+4,5e-7i	-1,7653+0,0i
0,71939-1,3895i	0,71939-1,3895i	0,71939-1,3895i
0,71939+1,3895i	0,71939+1,3895i	0,71939+1,3895i
-0,83673-1,3583i	-0,83673-1,3583i	-0,83673-1,3583i
-0,83673+1,3583i	-0,83673+1,3583i	-0,83673+1,3583i

TABLA II. DETALLES DE LAS REPETICIONES REALIZADAS PARA SOLUCIONAR (9)

Rep.	Método	Núm. de eval.	Tiempo [s]	Núm. Ejec.
1	PSO	492700	11,32	55
	UPSO	233400	5,54	45
2	PSO	407200	9,35	47
	UPSO	153800	3,52	30
3	PSO	407200	9,35	47
	UPSO	153800	3,52	30
4	PSO	498600	11,82	56
	UPSO	93000	2,19	18

A manera de demostración, se prueba el algoritmo con un sistema de ecuaciones lineal en dos dimensiones. Este tipo de sistemas son comunes en el análisis de circuitos lineales y pueden presentar co-

eficientes complejos, como es el caso de (10), cuando la topología es analizada en el dominio de la frecuencia, evitando la aparición de cualquier tipo de ecuación diferencial. Al ser el sistema de natura-

leza lineal, la forma común de solución se encuentra fácilmente en la literatura.

$$\begin{aligned} f_1 &= (1 - i) \cdot x_1 - 2i \cdot x_2 - 1 = 0 \\ f_2 &= 2i \cdot x_2 + (1 - 2i) \cdot x_1 = 0 \end{aligned} \tag{10}$$

La Tabla III presenta las raíces encontradas, tanto con las metaheurísticas como con el procedimiento algebraico tradicional. En este caso, PSO y UPSO requirieron de 253100 y 183700 evaluaciones de la función objetivo, y de 2,19 segundos y 1,91 segundos, respectivamente. Como se observa, existe una única solución y las estrategias implementadas la encuentran satisfactoriamente.

Posteriormente se realiza un análisis similar, también para un sistema de dos dimensiones pero con múltiples solucio-

nes, y que se muestra en (11). Debido al número de raíces, se opta por utilizar una representación gráfica de aquí en adelante.

$$\begin{aligned} f_1 &= x_1^2 - 7x_1x_2 + x_2^2 + 3 = 0 \\ f_2 &= 6x_2 - 3x_1x_2^2 - x_1^2 - 2 = 0 \end{aligned} \tag{11}$$

Al resolver el sistema (11) se encuentran las raíces mostradas en el plano complejo de la Fig. 1, donde cada solución corresponde a un marcador diferente, y donde el método utilizado para encontrarla se diferencia por su tonalidad. El orden en el que aparecen los marcadores corresponde a la secuencia con que se encontraron las raíces, de la siguiente forma: círculo, asterisco, cuadrado, rombo. Este formato se preserva en todas las gráficas siguientes.

TABLA III. RAÍCES OBTENIDAS PARA EL SISTEMA 2D LINEAL (10)

	Álgebra Lineal	PSO	UPSO
x1	0,029412+0,38235i	0,029411+0,38235i	0,029407+0,38235i
x2	0,17647+0,29412i	0,17647+0,29412i	0,17647+0,29412i

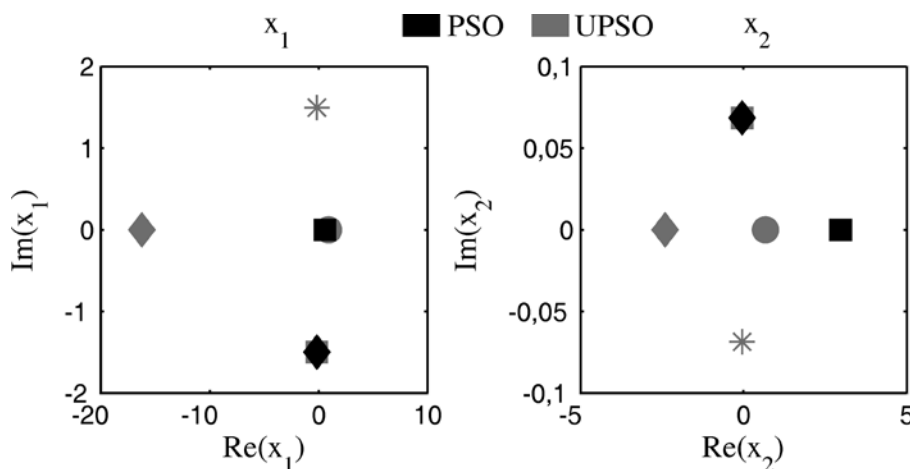


Fig. 1. Raíces obtenidas para el sistema 2D (11)

Es fácil apreciar que tres de las cuatro raíces encontradas con ambos métodos son similares (Fig. 1). En este caso, PSO se ejecutó 94 veces, consumiendo un total de 1227900 evaluaciones de la f_{Obj} en 14,53 segundos; mientras que UPSO se ejecutó 61 veces, para un total de 339800 evaluaciones, ejecutados en un tiempo de 4,31 segundos.

Siguiendo con la dinámica de experimentación, se migra a un sistema 3D no lineal. La función objetivo para esta prueba se construye con las expresiones mostradas en (12), y en la Fig. 2 se muestran las cuatro raíces obtenidas por cada metaheurística.

$$\begin{aligned}
 f_1 &= x_1 + x_2 + x_3^2 + 1 = 0 \\
 f_2 &= 0,3x_1 - x_2 + 0,1x_3 + 0,4 = 0 \quad (12) \\
 f_3 &= x_1^2 - 0,2x_1 + x_2^2 - 1,25x_3 - 0,6 = 0
 \end{aligned}$$

Es importante aclarar que el sistema no solo posee estas raíces, por lo que es normal que existan soluciones no comunes a ambas estrategias.

Para lograr estos resultados, PSO requirió un total de 1369600 evaluaciones, finalizando la búsqueda luego de 20,17 segundos. Sin embargo, UPSO logró hacerlo en 336300 evaluaciones, y en aproximadamente la tercera parte del tiempo (6,48 segundos).

Como un siguiente paso, se considera solucionar el sistema 5D mostrado en (13). Este conjunto de ecuaciones describe el equilibrio químico de un sistema [4]. Para este estudio las raíces complejas carecen de sentido en el problema real. No obstante, el objetivo de este trabajo se centra en verificar si PSO y/o UPSO logran encontrar las raíces del sistema.

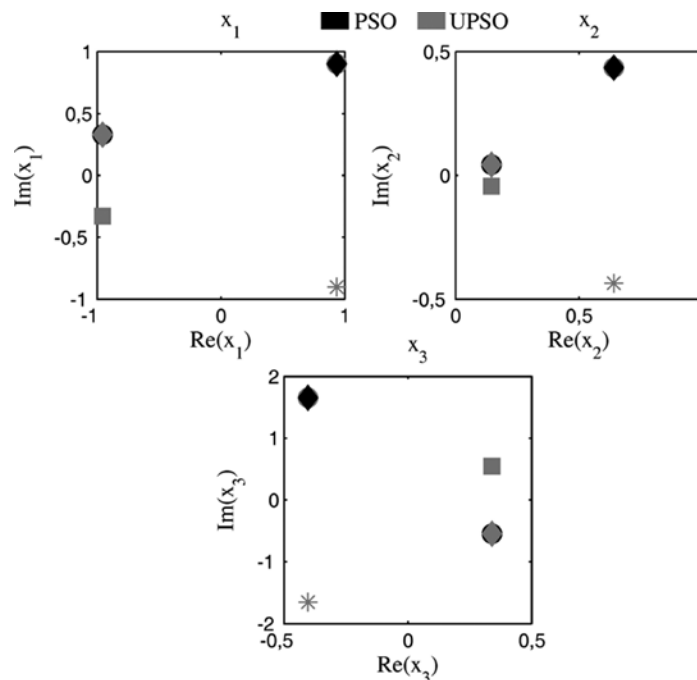


Fig. 2. Raíces obtenidas para el sistema 3D (12)

$$\begin{aligned}
 f_1 &= x_1 x_2 + x_1 - 3x_5 = 0 \\
 f_2 &= 2x_1 x_2 + x_1 + x_2 x_3^2 \\
 &\quad + R_8 \cdot x_2 - R x_5 + 2R_{10} x_2^2 \\
 &\quad + R_7 x_2 x_3 + R_9 x_2 x_4 = 0 \\
 f_3 &= 2x_2 x_3^2 + 2R_5 x_3^2 - 8x_5 \\
 &\quad + R_6 x_3 + R_7 x_2 x_3 = 0 \\
 f_4 &= R_9 x_2 x_4 + 2x_4^2 - 4R x_5 = 0 \\
 f_5 &= x_1(x_2 + 1) + R_{10} x_2^2 + x_2 x_3^2 \\
 &\quad + R_8 x_2 + R_5 x_3^2 + x_4^2 - 1 \\
 &\quad + R_6 x_3 + R_7 x_2 x_3 + R_9 x_2 x_4^2 = 0
 \end{aligned} \tag{13}$$

$$\begin{aligned}
 R &= 10; R_5 = 0,193; \\
 R_6 &= 0,002597/\text{sqrt}(40); \\
 R_7 &= 0,003448/\text{sqrt}(40); \\
 R_8 &= 0,00001799/40 \\
 R_9 &= 0,0002155/\text{sqrt}(40); \\
 R_{10} &= 0,00003846/40
 \end{aligned}$$

$$\begin{aligned}
 f_1 &= x_1 - 0,25428722 \\
 &\quad - 0,18324757x_4 x_3 x_9 = 0 \\
 f_2 &= x_2 - 0,37842197 \\
 &\quad - 0,16275449x_1 x_{10} x_6 = 0 \\
 f_3 &= x_3 - 0,27162577 \\
 &\quad - 0,16955071x_1 x_2 x_{10} = 0 \\
 f_4 &= x_4 - 0,19807914 \\
 &\quad - 0,15585316x_7 x_1 x_6 = 0 \\
 f_5 &= x_5 - 0,44166728 \\
 &\quad - 0,19950920x_7 x_6 x_3 = 0 \\
 f_6 &= x_6 - 0,14654113 \\
 &\quad - 0,18922793x_8 x_5 x_{10} = 0 \\
 f_7 &= x_7 - 0,42937161 \\
 &\quad - 0,21180486x_2 x_5 x_8 = 0 \\
 f_8 &= x_8 - 0,07056438 \\
 &\quad - 0,17081208x_1 x_5 x_6 = 0 \\
 f_9 &= x_9 - 0,34504906 \\
 &\quad - 0,19612740x_{10} x_6 x_8 = 0 \\
 f_{10} &= x_{10} - 0,42651102 \\
 &\quad - 0,21466544x_4 x_8 x_1 = 0
 \end{aligned} \tag{14}$$

En la Fig. 3 se muestra la ubicación de las raíces encontradas con ambas técnicas. Esta vez, PSO se demoró 780,59 segundos para cumplir las 30588400 evaluaciones realizadas, mientras que UPSO utilizó 851,25 segundos para un total de 27820400 ejecuciones de la función objetivo. Como se observa, para este problema particular, la versión original del algoritmo aventajó a la versión unificada, a pesar de haber realizado un mayor número de ciclos (caso contrario a las anteriores pruebas).

Se analiza un último sistema, (14), como muestra del comportamiento de la estrategia propuesta frente a un conjunto de 10 ecuaciones, llamado *Interval Arithmetic Benchmark*, el cual se estudia en [4].

La Tabla IV muestra las raíces encontradas con PSO, mientras que la Tabla V sintetiza las halladas con UPSO. Para llegar a estos resultados, se requirió de 20948200 y de 31904100 evaluaciones, con tiempos de 487,92 y 971,54 segundos, respectivamente. Con UPSO, las partes imaginarias de la raíz encontrada presentan órdenes de magnitud menores a 1×10^{-8} , por lo que sería posible decir que UPSO ha encontrado una solución real durante toda la ejecución de la estrategia de estudio. Sin embargo, es importante observar que solamente se encontró una raíz, por lo que se podría decir que no se desempeñó adecuadamente al compararlo con su versión original (para este caso en particular).

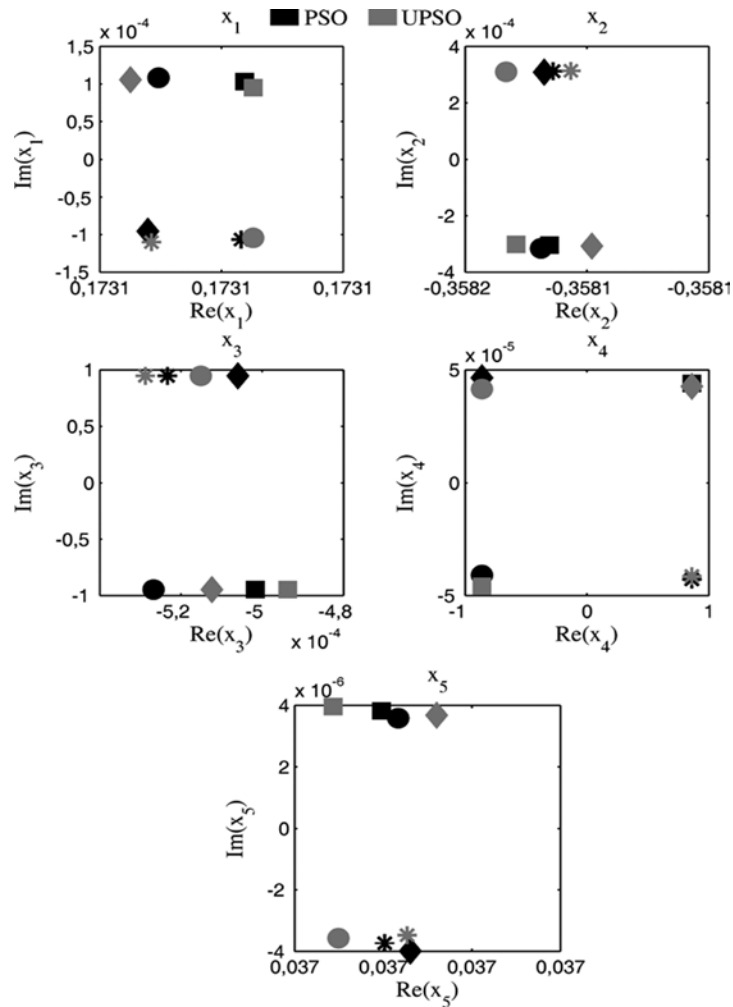


Fig. 3. Raíces obtenidas para el sistema 5D (13)

CONCLUSIONES

PSO y UPSO son métodos alternativos para la búsqueda de soluciones en sistemas de ecuaciones no lineales. Aun siendo estrategias de optimización, pueden ser fácilmente utilizados para encontrar la solución a un sistema de ecuaciones, aplicando una transformación de problema a través del teorema de las raíces reales. A su vez, permiten la libre modificación del algoritmo base para la búsqueda de raíces complejas, e incluso de múltiples de ellas. Este trabajo representa un punto ideal para el inicio de investigaciones dedicadas

a posibles aplicaciones en fenómenos físicos específicos.

Por otra parte, se encontró que para sistemas de menos de 5 dimensiones, PSO resulta ser aproximadamente 3 veces más costoso en cuanto a evaluaciones realizadas a la función objetivo que la versión unificada. Cuando se analiza el sistema de 5D, la relación entre el número de evaluaciones en la función objetivo requeridas por PSO y UPSO es de aproximadamente 1:1, por lo cual se puede inferir que necesitan un esfuerzo similar para llegar a los resultados. Por último, en el sistema de

TABLA IV. RAÍCES OBTENIDAS PARA EL SISTEMA 10D (14)

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
0,26-9,7e-12i	0,38-4,6e-13i	0,28+1,2e-11i	0,2-1e-11i	0,45-2,8e-11i	0,15-1,7e-11i	0,43+1,8e-11i	0,07+5,9e-12i	0,35-1,9e-11i	0,43-8,2e-12i
2,1-0,3i	0,66+1,9i	1,8-0,62i	-1,5+1,5i	-1,2+2,1i	-2,4+0,26i	2,6-1,3i	-1,8+1,6i	-1,5-2i	0,24-2,3i
-1,1-2,2i	-1,8+1,4i	-0,98+2,1i	0,78-2,1i	-2+1,1i	-2,5+0,36i	-1,4-1,7i	0,7-2,2i	-2,6-0,63i	-0,13+2,6i
1,8+1e-10i	2+3,5e-11i	1,6+1,3e-10i	2,1-4,4e-12i	2,6-6e-11i	2,4-1e-10i	2,7-1,2e-10i	2,1-7,9e-11i	2,6-1,1e-10i	2,2+4,1e-11i
-1,9+1,1i	-0,67-2,1i	-0,92+1,5i	2-1i	1,6-1,9i	2,8+0,35i	-2,2+0,27i	2,1-1,1i	1,7+2,8i	0,19+2,4i
2,2-0,52i	-0,36-2,1i	-1,7+0,68i	1,7-1,5i	-0,92+1,7i	2,4-0,41i	2,4-0,87i	1,8-1,6i	-1,8-1,9i	0,18-2,6i
2,2+0,52i	-0,36+2,1i	-1,7-0,68i	1,7+1,5i	-0,92-1,7i	2,4+0,41i	2,4+0,87i	1,8+1,6i	-1,8+1,9i	0,18+2,6i
1,3+1,7i	-1,6+1i	-0,48+1,5i	0,7-2,1i	2,3-0,74i	-2,5-0,16i	1,7+1,8i	0,62-2,3i	2,9+1,2i	0,27-2,4i
-1,6+0,94i	2,1-1,1i	-0,75+1,5i	-0,53+2,1i	1+2,4i	-2,7+0,16i	-2,2+1,7i	-0,73+2,3i	1,2-3i	2,4+0,074i
0,72-1,9i	0,77-2,2i	-1,3-1,3i	-1,7-1,2i	-1,7+1,4i	2,6-0,43i	0,92-2,5i	-2-1,3i	-2,6-1,2i	2,5-0,062i
-1,2+1,6i	1,8+1,5i	0,5+1,8i	-0,7-2,1i	-2,3-0,71i	2,7+0,12i	-1,2+2,4i	-0,92-2,3i	-2,7+0,67i	0,28-2,3i
-1,1+2,2i	-1,8-1,4i	-0,98-2,1i	0,78+2,1i	-2-1,1i	-2,5-0,36i	-1,4+1,7i	0,7+2,2i	-2,6+0,63i	-0,13-2,6i
-1,9+0,72i	-1,7+1,1i	1,4-1,3i	0,5-2i	1+2,4i	2,8-0,46i	-1,4+1,7i	0,4-2,2i	0,56-3,1i	2,5+0,14i
1,3-1,3i	1,6+1,2i	0,4+1,3i	-0,54-2,1i	2,6+0,3i	2,6-0,33i	1,5-2,5i	-0,73-2,3i	2,9-0,98i	-0,12+2,1i
1,1-2i	-0,59+2,1i	1,7+1,6i	1,8+1,1i	-1,6+1,6i	-2,3+0,3i	0,61-2,3i	1,8+1,2i	-1,9-1,4i	2,6+0,2i
1,3-1,7i	-1,6-1i	-0,48-1,5i	0,7+2,1i	2,3+0,74i	-2,5+0,16i	1,7-1,8i	0,62+2,3i	2,9-1,2i	0,27+2,4i

TABLA V. RAÍCES COMPLEJAS ENCONTRADAS USANDO EL ALGORITMO DE MÚLTIPLES RAÍCES CON UPSO EN (14)

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
0,26-6e-12i	0,38-2,1e-12i	0,28+9,8e-11i	0,2-1,3e-13i	0,45-1,1e-14i	0,15-2,5e-12i	0,43+1,3e-11i	0,073+7,7e-13i	0,35-2e-14i	0,43-1,8e-12i

dimensión 10, la estrategia con el método del enjambre unificado resulta no ser adecuada para la búsqueda de múltiples raíces, ya que requirió de aproximadamente 1,5 veces más evaluaciones para llegar a

una sola raíz que PSO en hallar 16 soluciones. Por todo lo anterior, se puede decir que UPSO hace parte de una buena estrategia para sistemas menores o iguales a 5 dimensiones, mientras PSO resulta apro-

piado para sistemas mayores a este límite (o, por lo menos, para mayores o iguales a 10).

En este trabajo igualmente destacamos la capacidad de los dos algoritmos utilizados, pues constituyen en sí una herramienta con muchas posibilidades de aplicación en la ingeniería. Se está aún en la etapa de conocimiento sobre su convergencia, tiempo de procesamiento y dependencia de estos con la naturaleza de la función objetivo.

REFERENCIAS

- [1] I. Amaya et al., "Solution of the Mathematical Model of a Nonlinear Direct Current Circuit Using Particle Swarm Optimization", *Revista Dyna*, vol. 79, no. 172, 2011, pp. 77-84.
- [2] J. Cruz et al., "Solution of the mathematical model of a DC nonlinear electronic circuit using an optimization strategy: Application of the original and unified Particle Swarm Metaheuristics," in *4th IEEE Colombian Workshop on Circuits and Systems (CWCAS)*, (2012), pp. 1-6.
- [3] I. Amaya et al., "Real Roots of Nonlinear Systems of Equations Through a Metaheuristic Algorithm," *Revista Dyna*, vol. 78, no. 170, 2011, pp. 15-23.
- [4] C. Grosan and A. Abraham, "A New Approach for Solving Nonlinear Equations Systems," *IEEE transactions on systems, man, and cybernetics*, vol. 38, no. 3, pp. 698-714, 2008.
- [5] L. Gómez, "Propuesta de demostración del teorema sobre la relación entre sistemas de ecuaciones y el problema de optimización (comunicación interna)". Bucaramanga, 2010, pp. 1-2.
- [6] O. Roa et al., "Solution of nonlinear circuits with the Central Force Optimization algorithm," in *4th IEEE Colombian Workshop on Circuits and Systems (CWCAS)*, (2012), pp. 1-6.
- [7] S. Reyes et al., "Real and/or Complex Roots Calculation of Nonlinear Equations Systems through Modified Particle Swarm Optimization," *Ingenieria y Universidad*, vol. 16, no. 2, Oct. 2012, pp. 349-362.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942-1948.
- [9] H. Babaei and A. Khosravi, "An Improve PSO Based Hybrid Algorithms," in *International Conference on Management and Service Science*, (2011), pp. 1-5.
- [10] J. Huang and D. Cao, "Particle swarm optimization method for solving quadratic eigenvalues," in *3rd International Conference on Biomedical Engineering and Informatics* (2010), pp. 2932-2935.
- [11] K. E. Parsopoulos and M. N. Vrahatis, "Unified Particle Swarm Optimization in Dynamic Environments," *Applications of Evolutionary Computing*, vol. 3449, 2005, pp. 590-599.
- [12] R. Correa et al., "Uso de algoritmos metaheurísticos híbridos para la minimización de entropía en problemas de transferencia de calor en circuitos electrónicos," *Revista Ingeniería y Universidad*, vol. 15, no. 2, 2011, pp. 403-421.
- [13] Z. Zhan et al., "Experimental Study on PSO Diversity," *Computational Intelligence*, 2010, pp. 310-317.