

**SOLUCIONES DE SISTEMAS DE ECUACIONES NO LINEALES MEDIANTE EL
MÉTODO METAHEURÍSTICO PSO**

**CÉSAR AUGUSTO ARIAS VESGA
RICARDO JAVIER AGUILAR VARGAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA
Y TELECOMUNICACIONES
BUCARAMANGA**

2011

**SOLUCIONES DE SISTEMAS DE ECUACIONES NO LINEALES MEDIANTE EL
MÉTODO METAHEURÍSTICO PSO**

**CÉSAR AUGUSTO ARIAS VESGA
RICARDO JAVIER AGUILAR VARGAS**

Trabajo de Grado para optar al título de INGENIERO ELECTRÓNICO

**DR. CARLOS RODRIGO CELY
DIRECTOR TRABAJO DE GRADO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA
Y TELECOMUNICACIONES
BUCARAMANGA**

2011

AGRADECIMIENTOS

Primordialmente al gran creador Dios por darnos la vida y después a todas esas personas especiales de las cuales agradecemos su amistad, apoyo, ánimo y compañía en las diferentes etapas de este nuevo logro. Algunas están aquí, otras en nuestros recuerdos y en nuestro corazón, a todas ellas, simplemente gracias por formar parte de este proceso, por todo lo que nos han brindado y por sus bendiciones.

CONTENIDO

	Pág.
1. INTRODUCCIÓN	14
2. FUNDAMENTOS TEÓRICOS	15
3. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)	16
4. UNIFIED PARTICLE SWARM OPTIMIZATION (UPSO)	19
5. RESULTADOS Y ANÁLISIS	19
5.1 SISTEMA DE TRES ECUACIONES	20
5.1.1 PSO	20
5.1.2. UPSO	25
5.1.3. UPSO LINEAL	31
5.1.4. UPSO EXPONENCIAL	35
5.2. CONFRONTACIÓN ENTRE TRES MÉTODOS	37
5.3. SISTEMA DE CINCO ECUACIONES	39
5.4. SISTEMA DE SIETE ECUACIONES	44
6. OBSERVACIONES Y CONCLUSIONES	47
7. REFERENCIAS	49

TABLAS

	Pág.
TABLA 1. SISTEMA DE TRES ECUACIONES. PSO CON 20 PARTÍCULAS	20
TABLA 2. SISTEMA DE TRES ECUACIONES. UPSO CON 20 PARTÍCULAS	26
TABLA 3. SISTEMA DE TRES ECUACIONES. UPSO LINEAL	32
TABLA 4. SISTEMA DE TRES ECUACIONES. UPSO EXPONENCIAL	35
TABLA 5. SISTEMA DE TRES ECUACIONES	38

FIGURAS

	Pág.
FIGURA 1. TIEMPO Y NÚMERO DE ITERACIONES VS PARÁMETROS	22
FIGURA 2. PSO CON 20 PARTÍCULAS. $C1=0,7$ Y $C2=0,5$	22
FIGURA 3. PSO CON 50 PARTÍCULAS. $C1=0,7$ Y $C2=0,5$	24
FIGURA 4. PSO CON 100 PARTÍCULAS. $C1=0,7$ Y $C2=0,5$	25
FIGURA 5. TIEMPO Y NÚMERO DE ITERACIONES VS FACTOR DE UNIFICACIÓN	29
FIGURA 6. UPSO CON 20 PARTÍCULAS. FACTOR $U=0,5$	29
FIGURA 7. UPSO CON 50 PARTÍCULAS. FACTOR $U=0,6$	30
FIGURA 8. UPSO CON 100 PARTÍCULAS. FACTOR $U=0,5$	31
FIGURA 9. UPSO LINEAL CON 20 PARTÍCULAS	32
FIGURA 10. UPSO LINEAL CON 50 PARTÍCULAS	33
FIGURA 11. UPSO LINEAL CON 100 PARTÍCULAS	34
FIGURA 12. UPSO EXPONENCIAL CON 20 PARTÍCULAS	35
FIGURA 13. UPSO EXPONENCIAL CON 50 PARTÍCULAS	36
FIGURA 14. UPSO EXPONENCIAL CON 100 PARTÍCULAS	37
FIGURA 15. EFECTO DE LA PRECISIÓN.	38
FIGURA 16. UPSO CON 50 PARTÍCULAS. FACTOR $U=0,5$	39
FIGURA 17. UPSO CON 50 PARTÍCULAS. FACTOR $U=0,6$	40
FIGURA 18. UPSO CON 100 PARTÍCULAS. FACTOR $U=0,5$	41
FIGURA 19. UPSO CON 100 PARTÍCULAS. FACTOR $U=0,6$.	41
FIGURA 20. UPSO CON 150 PARTÍCULAS. FACTOR $U=0,5$.	42

FIGURA 21. UPSO CON 150 PARTÍCULAS. FACTOR $U=0,6$.	43
FIGURA 22. UPSO CON 150 PARTÍCULAS. FACTOR $U=0,5$.	44
FIGURA 23. UPSO CON 200 PARTÍCULAS. FACTOR $U=0,5$.	45
FIGURA 24. UPSO CON 250 PARTÍCULAS. FACTOR $U=0,5$.	46

RESUMEN

TÍTULO:

SOLUCIONES DE SISTEMAS DE ECUACIONES NO LINEALES MEDIANTE EL MÉTODO METAHEURÍSTICO PSO*

AUTORES:

CÉSAR AUGUSTO ARIAS VESGA**

RICARDO JAVIER AGUILAR VARGAS**

PALABRAS CLAVES:

PARTICLE SWARM OPTIMIZATION (PSO), COEFICIENTES COMPLEJOS, ALGORITMO, POLINOMIO, RAÍZ, SISTEMAS DE ECUACIONES NO LINEALES.

DESCRIPCIÓN:

Problemas fundamentales de la ciencia y la ingeniería pueden ser modelados mediante sistemas de ecuaciones no lineales, donde la solución numérica eficiente de estos sistemas, juega un papel importante en el avance de estas áreas. En el presente artículo, se muestra los resultados de simulación en la solución de sistemas de ecuaciones no lineales de la forma polinomial con coeficientes complejos, comúnmente utilizados en el procesamiento de señales, diseño de filtros, ecualizadores, desacopladores y otros componentes de telefonía móvil por mencionar. Encontrar una solución para tales sistemas requiere muchas veces un esfuerzo computacional bastante alto, haciendo necesario convertir este tipo de problemas en un problema de optimización. Por lo tanto se desarrolla una alternativa computacional basada en técnicas heurísticas y comportamientos naturales conocida como Optimización por enjambre de partículas unificado, la cual corresponde a una modificación del algoritmo estándar de Optimización por enjambre de partículas. Dicha técnica manifiesta una rápida convergencia, con lo que se concluirá que su rendimiento es mejor frente a métodos como Optimización por enjambre de partículas y Newton Raphson multivariable para sistemas de ecuaciones con coeficientes complejos. Además se muestra que incrementar el número de partículas fue necesario para cuando el grado de los polinomios, el orden de los sistemas de ecuaciones y la cantidad de variables aumentaban, elevando el tiempo de cómputo y de número iteraciones.

* Trabajo de Grado.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Director: Carlos Rodrigo Correa Cely.

SUMMARY

TITLE:

SOLVING SYSTEMS OF NONLINEAR EQUATIONS BY PSO METAHEURISTIC METHOD*

AUTHORS:

CÉSAR AUGUSTO ARIAS VESGA**

RICARDO JAVIER AGUILAR VARGAS**

INDEX TERMS:

PARTICLE SWARM OPTIMIZATION (PSO), COMPLEX COEFFICIENTS, ALGORITHM, POLYNOMIAL, ROOT, SYSTEMS OF NONLINEAR EQUATIONS.

DESCRIPTION:

The most of fundamentals problems present in science and engineering can be modeled by systems of nonlinear equations, where to get an efficient numerical solution to these systems plays an important role in advancing these areas. This article shows the simulation results to solving a system of nonlinear equations of the form polynomial with complex coefficients, frequently used in signal processing, when designing filters, equalizers, decouplers and other components of mobile phone for instance. Finding a solution for such systems often requires a fairly high computational effort, making it necessary to convert this kind of problems in an optimization task. Therefore an alternative computational based on heuristics techniques and behavior of nature is developed modifying the standard Particle Swarm Optimization algorithm, this new tool is known like Particle Swarm Optimization unified, which shows a fast convergence and performance better than methods such as Particle Swarm Optimization and multivariable Newton Raphson to solve systems of nonlinear equations with complex coefficients. Moreover this document shows to increase the number of particles was needed when the degree of polynomials, the order of the systems of equations and number of variables increased, using higher costs computational like the computation time and the number iterations.

* Degree Work.

** Faculty of Physic mechanical Engineering. School of electric, electronic and telecommunications. Project director: Carlos Rodrigo Correa Cely.

1. INTRODUCCIÓN

Este documento describe un método metaheurístico de optimización con el objetivo de solucionar sistemas de ecuaciones no lineales de forma polinomial con coeficientes complejos comúnmente utilizados en distintos campos de la ingeniería [1]. Con el ánimo de alcanzar dicho propósito se presenta una variante del algoritmo de optimización estándar; tal variante viene dada mediante la adición de nuevos parámetros y factores con los que se mostró una nueva alternativa de solución que arrojó mejores resultados en el momento de encontrar soluciones a esta clase de sistemas de ecuaciones satisfaciendo ciertos requisitos impuestos como lo es la precisión deseada. Cuando se habla de optimización se hace referencia al proceso de encontrar la mejor o las mejores soluciones para un problema. Optimización por enjambre de partículas (PSO) es una técnica metaheurística basada en un comportamiento natural que rige sobre una población de individuos, los cuales recorren el espacio de búsqueda del problema, con el fin de encontrar el mejor conjunto de soluciones. Dicho método y su variante UPSO (Optimización por Enjambre de Partículas Unificado) caracterizado por un factor de unificación, que favorece las habilidades de búsqueda dentro del espacio solución sin agregar otros requisitos a la función objetivo [2][3], fueron aplicados para la solución de sistemas. Bajo un enfoque propuesto en [4][18], asociamos a la función objetivo cada uno de los polinomios que conforma el sistema de ecuaciones, transformando el problema de solucionar un sistema de ecuaciones en un problema de optimización. Posteriormente se exponen los aspectos fundamentales y la teoría concerniente al manejo de polinomios con coeficientes complejos, así como una sencilla descripción de los métodos empleados. Como parte final del artículo se muestran algunos de los resultados obtenidos por los métodos implementados mediante tablas en donde se exponen variaciones de parámetros influyentes en el comportamiento del algoritmo, además de gráficas que ilustran el comportamiento de individuos (partícula-solución) en la búsqueda del conjunto solución, así como la función objetivo.

2. FUNDAMENTOS TEÓRICOS

Gran cantidad de problemas o fenómenos que se presentan frecuentemente en las ciencias, ingenierías o en el mundo real, permiten describirse mediante modelos matemáticos, los cuales pueden involucrar numerosos parámetros y variables con el objetivo de obtener una representación aproximada al sistema real. Estos modelos están formados por sistemas de ecuaciones que en muchos casos suelen ser no lineales, como lo es en el presente artículo. La finalidad es determinar valores o conjuntos solución para los cuales las funciones (*función objetivo*) se anulan [5] o cumplen con los requerimientos impuestos sobre el modelo. Dado un sistema no lineal de ecuaciones polinomiales como el mostrado en las ecuaciones (1) - (3).

$$P_1(x_1, x_2, \dots, x_i) = 0 \quad (1)$$

$$P_2(x_1, x_2, \dots, x_i) = 0 \quad (2)$$

⋮

$$P_i(x_1, x_2, \dots, x_i) = 0 \quad (3)$$

donde $i = 1, 2, \dots, n$ generando un sistema cuadrado de polinomios y $P_i \in C[x]^1$ [6]. Haciendo del sistema de ecuaciones un problema de minimización como el mencionado en [7], entonces la función objetivo se toma de la siguiente forma:

$$FO = \sum_{i=1}^n \|P_i(x)\|^2 \quad (4)$$

donde es posible hallar un conjunto de variables para los cuales las funciones P_i se hacen cero simultáneamente, dando como resultado el conjunto solución al sistema propuesto [8]. Al estudiar sistemas de ecuaciones no lineales polinómicas en varias variables, se requiere un buen entendimiento de lo que se pueda decir acerca de una ecuación polinomial en una variable.

¹ $C[x]$ es el conjunto de los números complejos.

Teorema: (Teorema Fundamental del Álgebra) Sea $P \in \mathbb{C}[x]$ un polinomio no nulo con coeficientes complejos de grado m mayor o igual que 1. Entonces P tiene exactamente m raíces (reales y/o complejas) contadas con su multiplicidad. [5][9].

Lo anterior indica que es factible predecir cuántas raíces tiene un polinomio de determinado grado, más no cuáles son sus raíces. Es posible encontrar las raíces complejas de polinomios $P(x) \in \mathbb{C}[x]$ de grado menor o igual a 4, por medio de ecuaciones que se obtienen a partir de los coeficientes del polinomio $P(x)$, mediante las operaciones elementales. Sin embargo, no hay una expresión que describa las raíces de un polinomio general cualquiera $P(x)$ de grado mayor o igual a 5 [5][10]. Tal inconveniente fue parcialmente resuelto por Évariste Galois, quien caracterizó cuáles eran los polinomios de grado mayor o igual a 5 para los cuales existe tal expresión, aunque no es fácilmente deducible de los coeficientes del polinomio, sino que tiene que ver con ciertos grupos generados por arreglos de las posibles raíces asociadas a los polinomios [6]. Extendiendo estos conceptos a sistemas de ecuaciones polinomiales, un factor clave fue aportado por Bezout en su teorema, donde establece que la cantidad máxima de soluciones no debe exceder el producto de los grados de cada función que conforman el sistema [11].

3. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)

Hacia el año de 1995 surgió un algoritmo metaheurístico que acapararía la atención por su semejanza con aspectos de la naturaleza y más exactamente con el comportamiento de grupos de animales como bandadas de pájaros, enjambres de abejas y cardúmenes. Es así como *J. Kennedy* y *R.Eberhart* [12] fijándose en el comportamiento social de estos animales desarrollaron esta herramienta conocida como *Optimización por enjambre de partículas*. Para empezar se supone una población de i partículas, donde cada partícula del enjambre se identifica con dos variables de estado inicializadas aleatoriamente [13]. Para un problema N -dimensional la posición y la velocidad están representadas por matrices de

tamaño $N_p \times N_i$ como se muestra en (5) y (6), siendo X y V las posiciones y velocidades respectivamente. Cuando se menciona la palabra partícula se hace referencia a cada una de las potenciales soluciones del sistema.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N_i} \\ x_{21} & x_{22} & \cdots & x_{2N_i} \\ \vdots & \vdots & & \vdots \\ x_{N_p1} & x_{N_p2} & \cdots & x_{N_pN_i} \end{bmatrix} \quad (5)$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1N_i} \\ v_{21} & v_{22} & \cdots & v_{2N_i} \\ \vdots & \vdots & & \vdots \\ v_{N_p1} & v_{N_p2} & \cdots & v_{N_pN_i} \end{bmatrix} \quad (6)$$

Cada fila de la matriz X representa la posición de la i -ésima partícula en cada iteración, donde a su vez graba cual ha sido su mejor posición personal u óptimo local obtenida hasta el momento, dando lugar a una matriz que almacena las mejores posiciones personales de cada partícula como se muestra en (7). De la matriz de los mejores personales de cada partícula se selecciona el óptimo global que constituye la mejor posición del enjambre escogida entre los mejores óptimos locales de cada partícula como se indica en (8).

$$BestP = \begin{bmatrix} x^l_{11} & x^l_{12} & \cdots & x^l_{1N_i} \\ x^l_{21} & x^l_{22} & \cdots & x^l_{2N_i} \\ \vdots & \vdots & & \vdots \\ x^l_{N_p1} & x^l_{N_p2} & \cdots & x^l_{N_pN_i} \end{bmatrix} \quad (7)$$

$$GBestP = [x^g_{11} \quad x^g_{12} \quad \cdots \quad x^g_{1N_i}] \quad (8)$$

Entonces se actualiza la posición y la velocidad como se muestra en (9) y (10):

$$x_k = x_{k-1} + v_k \quad (9)$$

$$v_k = \omega \times v_{k-1} + c_1 \times r_1(x_i^l - x_k) + c_2 \times r_2(x_i^g - x_k)^2(10)$$

La nueva posición para cada partícula depende únicamente de la posición anterior de la partícula i y de la velocidad de la misma partícula en la iteración actual; en cambio para la actualización de la velocidad se necesitan incluir otros términos como x_i^g (óptimo global) que representa la mejor posición de todas las que hasta el momento se han presentado y x_i^l (óptimo local) es la mejor posición individual de la partícula. c_1 y c_2 son llamados coeficientes de aceleración cognitivo y social respectivamente, los cuales toman valores mayores de cero; por otro lado r_1 y r_2 son valores aleatorios que favorecen el modelado y que están entre cero y la unidad. ω es llamado peso inercial [14], que toma un valor decreciente modelado por [14][15]:

$$w_i = w_{max} - \frac{w_{max} - w_{min}}{N_i} \times i \quad (11)$$

donde N_i es el número de iteraciones totales e i es la iteración actual. Con este factor se controla la tendencia de la partícula a continuar en la dirección en la que se estaba moviendo y se regula la relación entre capacidad de exploración del espacio de búsqueda y habilidad de convergencia hacia las soluciones locales o globales [16], dictadas por el segundo y tercer sumando de la ecuación (10). Para localizar tanto el óptimo local de cada partícula como el óptimo global de todo el enjambre, al final de cada iteración, cada partícula evalúa con su posición su respectivo valor en la función objetivo para así comparar que tan cerca se encuentra de la respuesta o de los requerimientos establecidos.

² Se hace un cambio en el cálculo de la velocidad con el fin de que las partículas se muevan en todo el plano de búsqueda y no solo en una línea recta, lo cual garantiza su rápida convergencia.

4. UNIFIED PARTICLE SWARM OPTIMIZATION (UPSO)

Con el fin de mejorar la búsqueda sobre el espacio de trabajo se realizan variantes sobre las velocidades de las partículas, dependiendo de los óptimos locales y globales de la población, mejorando así las propiedades de exploración y explotación del enjambre. Las velocidades vienen definidas como [2][3][14]:

$$G_i = \chi \left(V_i + c_1 \times r_1 \times (x_i^l - X_i) + c_2 \times r_2 (x_i^g - X_i) \right) \quad (12)$$

$$L_i = \chi \left(V_i + c_1 \times r_3 \times (x_i^l - X_i) + c_2 \times r_4 (x_i^{g_i} - X_i) \right) \quad (13)$$

En (12) y (13) aparece el factor de restricción χ , que asegura la convergencia a un punto estable del espacio, garantizando que la velocidad no exceda un valor máximo [17]. Ahora bien, tales velocidades son agregadas en una simple ecuación (14) que actualiza el espacio de búsqueda dada por:

$$U_i = u \times G_i + (1 - u) \times L_i \quad (14)$$

$$X_{i+1} = X_i + U_i \quad (15)$$

donde u es el factor de unificación y maneja la influencia local y global sobre la velocidad actual [2][3]; la posición de las partículas se muestra en (15).

5. RESULTADOS Y ANÁLISIS

Las pruebas se realizaron en un computador convencional DELL Inspiron N4010 con las siguientes características:

- Procesador: Intel(R) core(TM) i3 CPU M350 @ 2.27GHz
- Memoria RAM: 4GB
- Sistema Operativo: Windows 7, 64-bit

Se realizaron tres repeticiones por cada prueba y se definió como criterio de parada una precisión de 1×10^{-8} en la solución, además se cronometró el tiempo de simulación con la herramienta tic-toc de Matlab®.

5.1 SISTEMA DE TRES ECUACIONES

Se implemento el siguiente sistema de ecuaciones, donde $j = \sqrt{-1}$

$$f(x_a, x_b, x_c) = 5 \times x_a - (3 + 2j) \times x_b^2 + (9 - 8j) \times x_c^2 = 0 \quad (16)$$

$$g(x_a, x_b, x_c) = (1 + j) \times x_a^2 + 4j \times x_b - 6j \times x_c + 3 - j = 0 \quad (17)$$

$$h(x_a, x_b, x_c) = j \times x_a^3 - 9 \times x_b^2 + 27 \times x_c + 1 - 5j = 0 \quad (18)$$

cuya solución analítica es:

5.1.1 PSO

Se variaron parámetros como la aceleración cognitiva y social (c_1, c_2), así como el número de partículas (N_i).

a) 20 Partículas

Tabla 1. Sistema de tres Ecuaciones. PSO con 20 Partículas

Parám. ³	Prueba	Iter. ⁴	Tiempo [s]	Raíces	Función Evaluada
C1=0,7 C2=0,7	1	263	136,591	$x_a = -0,312827246342435 - 0,277855059901068i$ $x_b = -0,186349586345492 + 0,832096512460354i$ $x_c = -0,258483257448701 + 0,080261646334421i$	$fx = 1,175176617129736e-008 + 6,896809034984130e-008i$ $gx = 1,336435095922184e-008 - 8,240586646390824e-008i$ $hx = 5,054716289976113e-008 + 4,895024208195764e-008i$
	2	280	145,796	$x_a = -0,312827270110763 - 0,277855064131361i$ $x_b = -0,186349569700547 + 0,832096527319496i$ $x_c = -0,258483270066083 + 0,080261650637346i$	$fx = 4,852065060845234e-008 - 4,171154954768497e-008i$ $gx = -2,358972839289208e-008 + 8,825314923299743e-008i$ $hx = 9,241070131338347e-010 - 3,359928069102125e-008i$
	3	276	144,102	$x_a = -0,312827258047091 - 0,277855042254720i$ $x_b = -0,186349575251008 +$	$fx = -2,635385398797041e-008 + 6,189883716700706e-008i$ $gx = 7,304269855623602e-008 -$

³ Expresión corta que significa parámetro de aquí en adelante.

⁴ Expresión corta que significa iteración de aquí en adelante.

				0,832096507199858i xc =-0,258483260183650 + 0,080261649162908i	9,025041158849945e-009i hx =-5,986195450446985e-008 -6,842447941579621e-008i
	Prom. ⁵	273	142,163		
C1=0,7 C2=0,5	1	277	138,675	xa =-0,312827255139122 - 0,277855040656348i xb =-0,186349566066304 + 0,832096516078040i xc =-0,258483262921772 + 0,080261652909649i	fx = 5,504780109522045e-008 +4,245740869990300e-008i gx = 6,169528266042335e-008 +4,059534242095708e-008i hx = 2,837636436936464e-008 -7,570190163619373e-008i
	2	293	147,422	xa =-0,367596879382419 - 2,043359219672385i xb = 0,535493788879350 - 1,315120816076836i xc =-0,232658222751411 - 0,453004830396100i	fx = 7,467268803296179e-009 - 1,384377945612414e-009i gx = -9,817543045187449e-008 +2,355152073363342e-009i hx = 2,712943025784398e-008 +4,931697894505760e-008i
	3	278	138,960	xa =-0,312827286021970 - 0,277855052842787i xb =-0,186349567422585 + 0,832096521175197i xc =-0,258483266081305 + 0,080261654023484i	fx =-7,932134121757528e-008 - 1,143354677601849e-008i gx = 3,575312002013220e-008 +9,146368906698399e-008i hx =3,172487428315662e-008 - 3,775351586909892e-009i
	Prom.	283	141,685		
C1=0,5 C2=0,7	1	323	155,650	xa =-0,312827238145728 - 0,277855066402798i xb =-0,186349583416344 + 0,832096506079459i xc =-0,258483257407833 + 0,080261647017473i	fx = 3,470949827555359e-008 - 6,426527088265743e-009i gx = 3,473201326542608e-008 - 8,016302477376058e-008i hx = -3,796739100891955e-008 +6,016021458776777e-009i
	2	415	203,275	xa = 0,522258141993030 + 1,712332074046266i xb = 1,787882425979858 - 0,207022804892836i xc = 0,880126802408745 + 0,103299189706833i	fx =-5,079364484572579e-008 - 6,295037380255053e-008i gx = 5,794982183005004e-008 +5,947033354658515e-008i hx = 4,167157285905887e-008 - 1,545850825124262e-008i
	3	320	159,672	xa =-0,312827258099460 - 0,277855092933224i xb =-0,186349592977092 + 0,832096516101014i xc =-0,258483263024155 + 0,080261640611898i	fx = -1,034293084511972e-008 - 6,367604921120318e-008i gx =-7,373411614608472e-008 -5,927980262221411e-008i hx =-5,952690074195743e-008 +2,247875929839438e-008i
	Prom.	353	172,865		

Fuente: Autores del Proyecto.

La figura 1 muestra el número de iteraciones y tiempo de cómputo frente a variaciones de los coeficientes de aceleración donde:

(a): para $c_1=0,7$ y $c_2=0,7$.

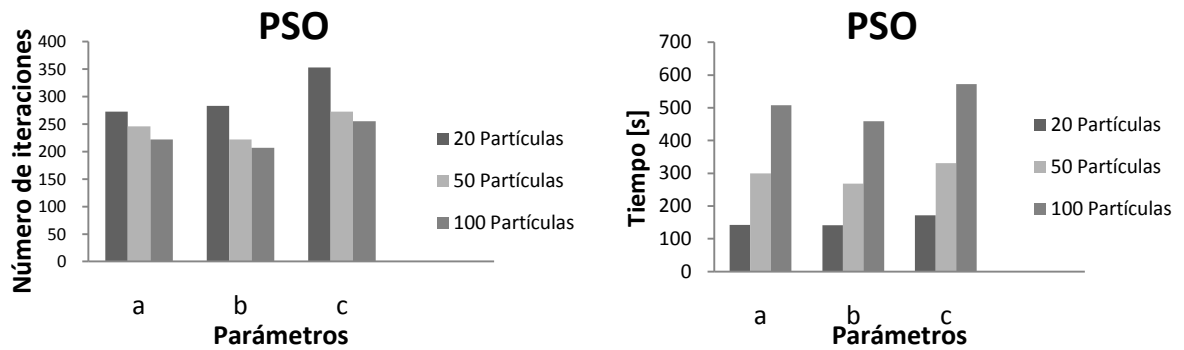
(b): para $c_1=0,7$ y $c_2=0,5$.

(c): para $c_1=0,5$ y $c_2=0,7$.

⁵ Expresión corta que significa promedio de aquí en adelante.

Se requieren menos recursos para los casos en que $(c_1 > c_2)$. En el caso (a) la variación en la cantidad de partículas no genera un aumento considerable en la cantidad de iteraciones, como si ocurre con el tiempo. En los casos b y c, para 50 y 100 partículas también hubo pequeñas diferencias en la cantidad de iteraciones.

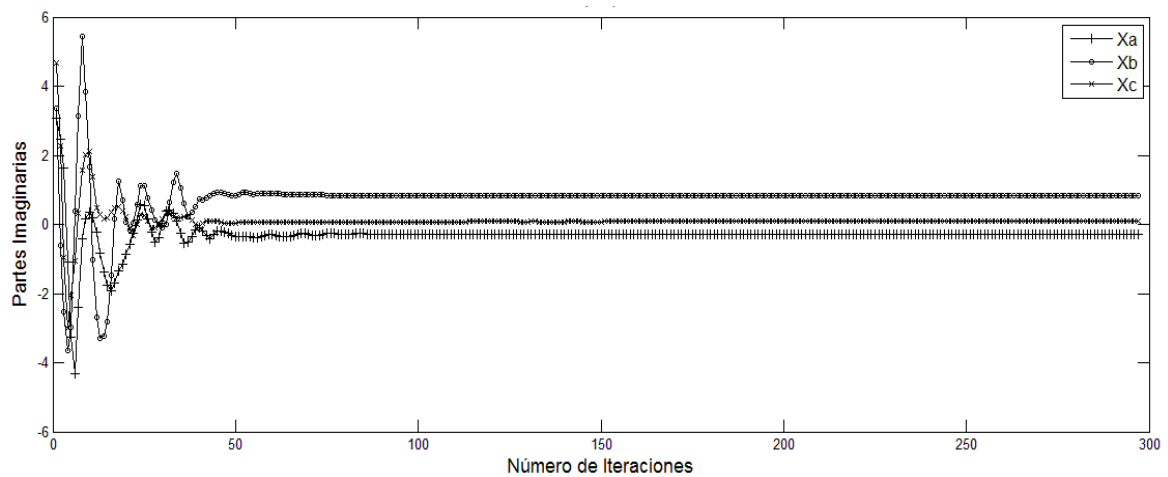
Figura 1. Tiempo y número de iteraciones vs parámetros.



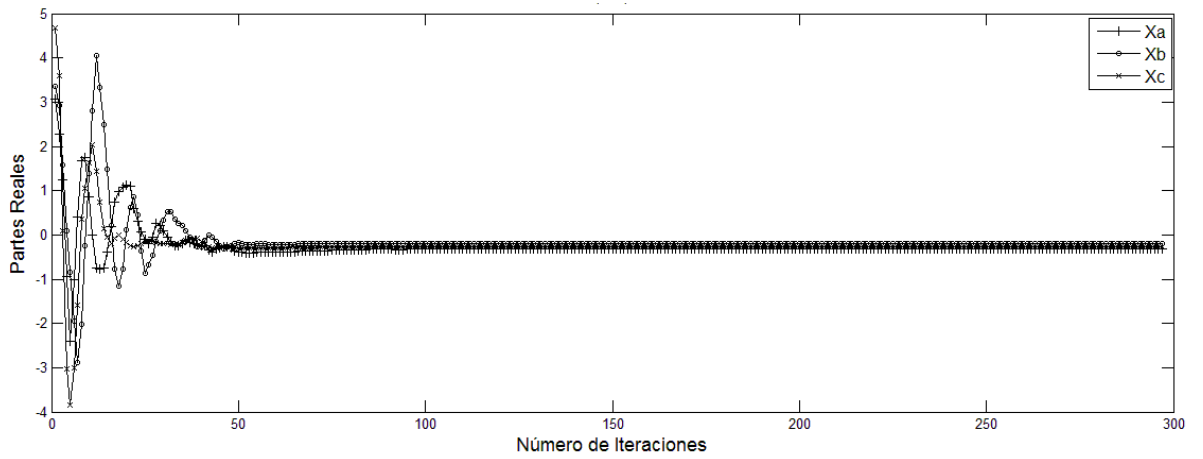
Fuente: Autores del Proyecto.

La figura 2 muestra los resultados de simulación para $c_1=0,7$ y $c_2=0,5$.

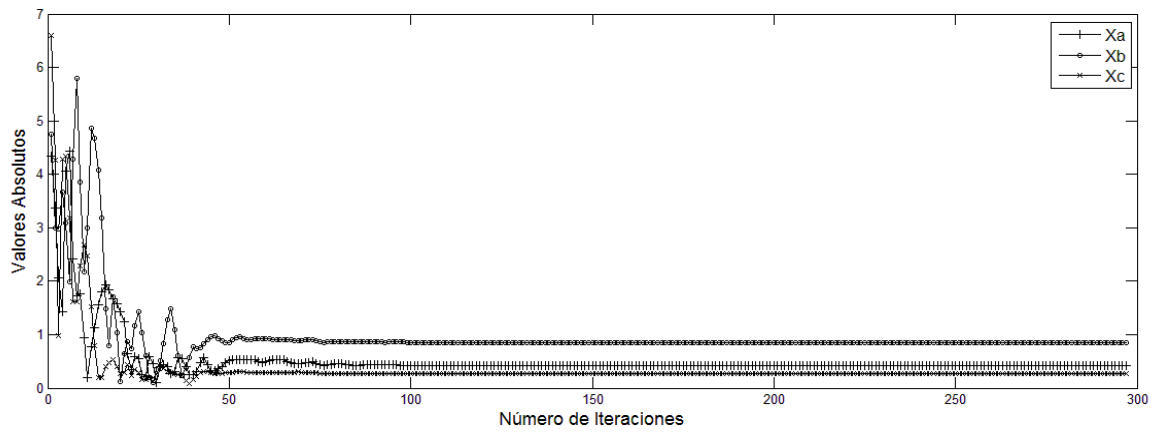
Figura 2. PSO con 20 partículas. $c_1=0,7$ y $c_2=0,5$.



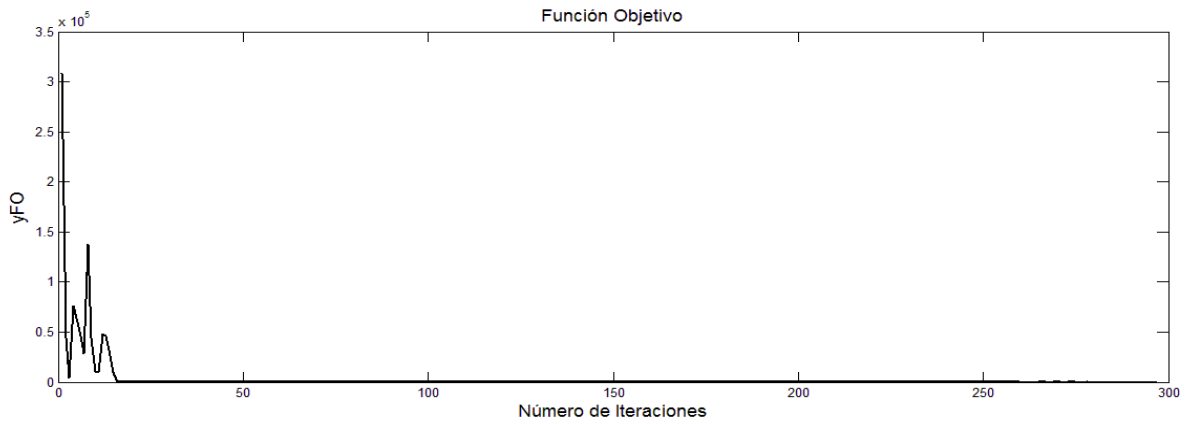
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.



(d) Función Objetivo.

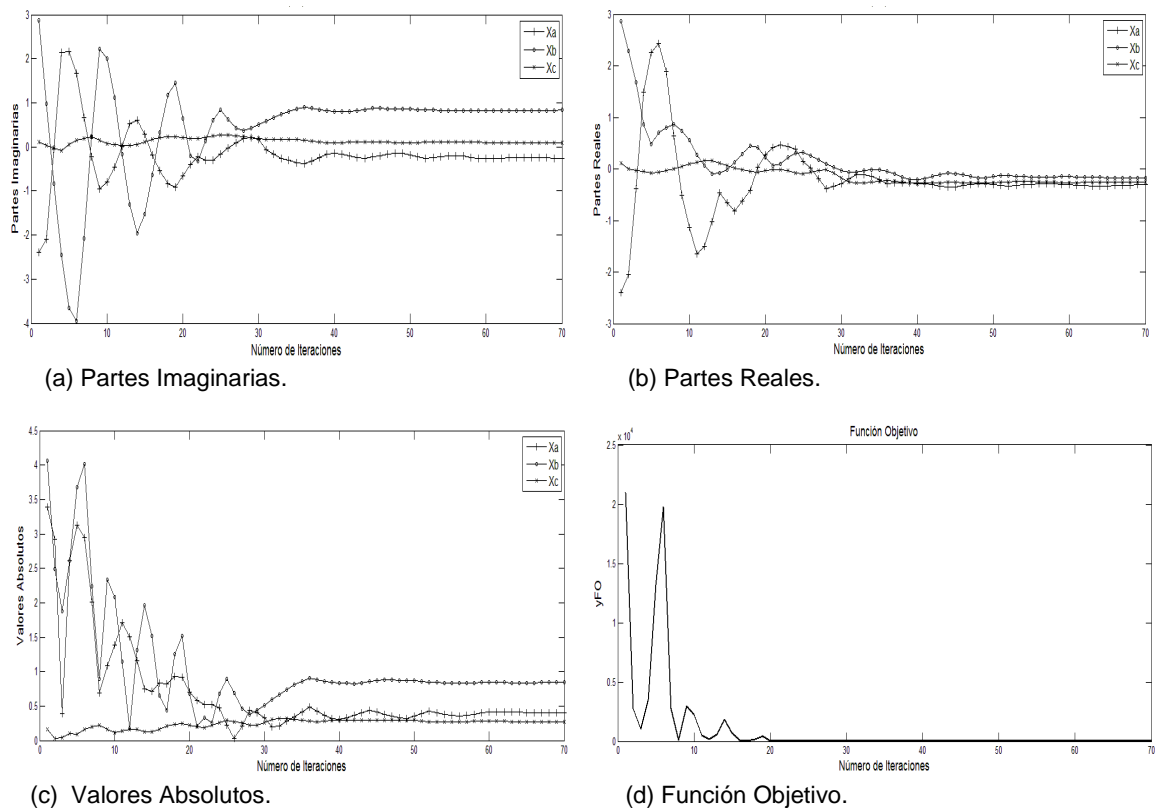
Fuente: Autores del Proyecto.

Puesto que las partículas no presentan gran variación aproximadamente después de la iteración número setenta, las figuras siguientes se muestran hasta dicho número de iteraciones, sin embargo, esto no indica que con ese número de iteraciones se llegó a la solución. De los resultados mostrados, se observa un menor tiempo de simulación cuando el coeficiente de aceleración cognitivo es mayor que el social ($c_1 > c_2$); además, a mayor cantidad de partículas presenta una disminución en la cantidad de iteraciones, pues se comparte mayor información entre ellas. Sin embargo, también aumenta el tiempo de respuesta y presenta un mayor consumo de recursos.

b) 50 Partículas

La figura 3 muestra los resultados de simulación para $c_1=0,7$ y $c_2=0,5$.

Figura 3. PSO con 50 partículas. $c_1=0,7$ y $c_2=0,5$.

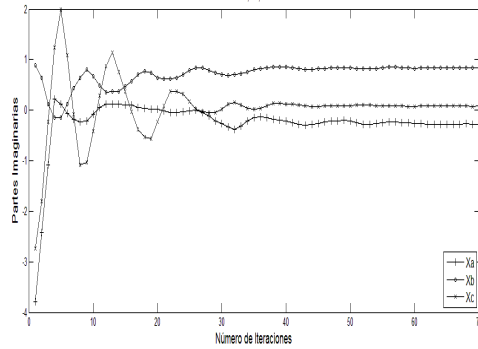


Fuente: Autores del Proyecto.

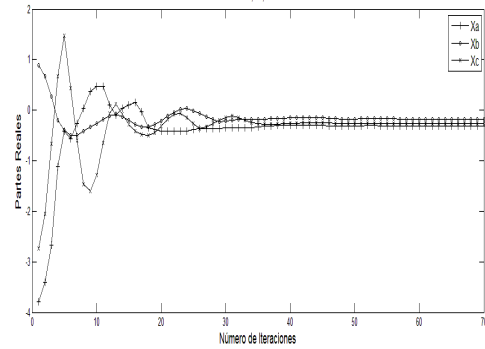
c) 100 Partículas

La figura 4 muestra los resultados de simulación para $c_1=0,7$ y $c_2=0,5$.

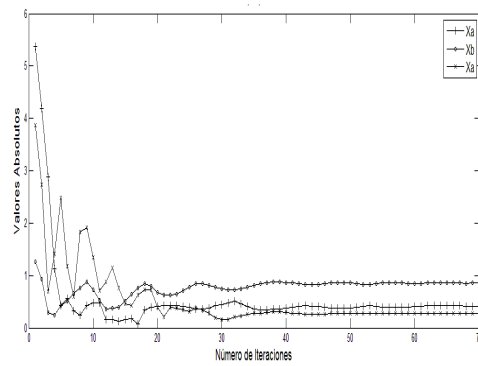
Figura 4. PSO con 100 partículas. $c_1=0,7$ y $c_2=0,5$.



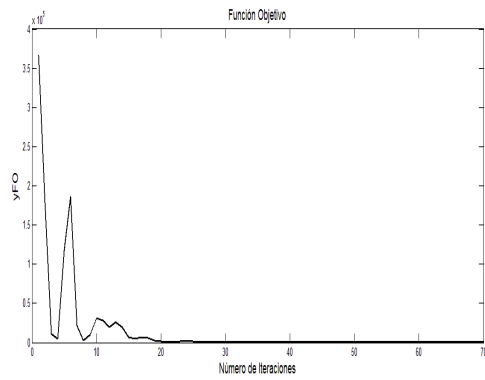
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.



(d) Función Objetivo.

Fuente: Autores del Proyecto.

5.1.2. UPSO

En este ítem, se varia el factor de unificación en un rango comprendido entre $u=0,1$ y $u=0,9$ con un paso de 0,1, además de variaciones lineales crecientes y exponenciales. Asimismo se varía la cantidad de partículas.

a) 20 Partículas

Tabla 2. Sistema de tres Ecuaciones. UPSO con 20 Partículas.

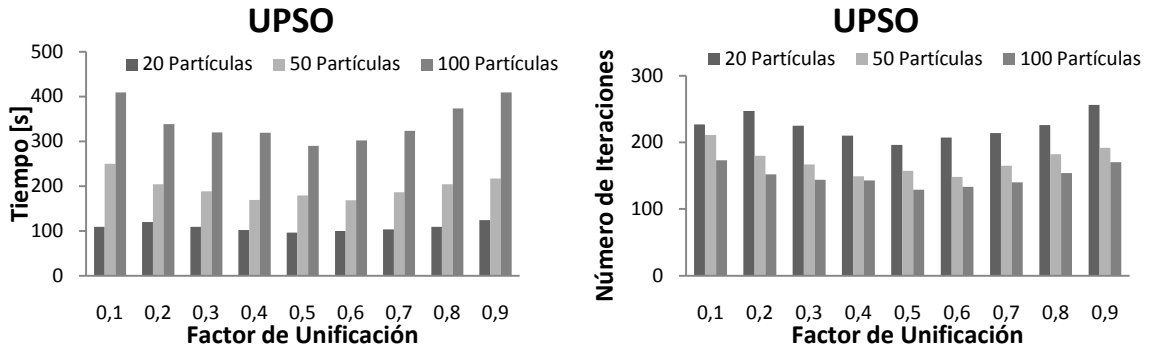
Factor De Unificación	Prueba	Iter.	Tiempo[s]	Raíces	Función Evaluada
0,1	1	229	111,339	$xa = -0,312827261025169 - 0,277855086649378i$ $xb = -0,186349592170362 + 0,832096515811543i$ $xc = -0,258483263122654 + 0,080261640762501i$	$fx = -2,312242985968993e-008 - 3,802725123680517e-008i$ $gx = -6,404446217800341e-008 - 5,244507639279306e-008i$ $hx = -6,267958685413078e-008 + 1,003260763354774e-008i$
	2	236	112,958	$xa = -0,312827275894129 - 0,277855072120177i$ $xb = -0,186349586958210 + 0,832096529646115i$ $xc = -0,258483270766596 + 0,080261641874385i$	$fx = 4,010358545247783e-009 + 2,759503181515299e-008i$ $gx = -9,450716387249258e-008 + 3,081661059667340e-008i$ $hx = -3,751822941211458e-008 - 1,063753529706446e-010i$
	3	217	104,029	$xa = -0,312827243678341 - 0,277855061984193i$ $xb = -0,186349583180688 + 0,832096504441325i$ $xc = -0,258483255858718 + 0,080261644842940i$	$fx = 8,053464062296456e-009 + 2,335137816444899e-008i$ $gx = 3,384431046526970e-008 - 8,228811543276038e-008i$ $hx = -1,727479759239259e-008 - 6,436802024722965e-008i$
	Prom.	227	109,442		
0,2	1	258	125,158	$xa = -0,312827246054004 - 0,277855079448394i$ $xb = -0,186349584362061 + 0,832096508314676i$ $xc = -0,258483257000166 + 0,080261647099271i$	$fx = -1,373322011666289e-009 - 5,569959693474402e-008i$ $gx = 1,142346350846424e-008 - 7,613687236229794e-008i$ $hx = 8,278727037236422e-009 + 3,620033162121672e-008i$
	2	229	111,345	$xa = -0,312827256628464 - 0,277855038905174i$ $xb = -0,186349572020848 + 0,832096510109800i$ $xc = -0,258483259414682 + 0,080261652254344i$	$fx = -1,238067132147691e-008 + 7,174413951815950e-008i$ $gx = 8,380935812013490e-008 - 2,628406159388419e-009i$ $hx = 1,437200047860188e-008 - 2,523434794454715e-008i$
	3	254	123,070	$xa = -0,312827258850014 - 0,277855060292380i$ $xb = -0,186349572747231 + 0,832096519412006i$ $xc = -0,258483263739742 + 0,080261650790054i$	$fx = 3,553028565805860e-008 - 9,947975332380565e-009i$ $gx = 1,270409200060385e-008 + 2,453680281178094e-008i$ $hx = 3,696887063142640e-008 - 1,167202334784179e-008i$
	Prom.	247	119,857		
0,3	1	235	114,313	$xa = -0,312827274920657 - 0,277855075202245i$ $xb = -0,186349580440275 + 0,832096529047551i$ $xc = -0,258483271414769 + 0,080261648032161i$	$fx = 3,137669235542262e-009 - 4,251930030285678e-008i$ $gx = -5,887538767623823e-008 + 5,984293949268249e-008i$ $hx = -4,243768714218277e-008 + 6,818960418542019e-008i$
	2	221	105,804	$xa = -0,312827269423410 - 0,277855087698855i$ $xb = -0,186349591306899 + 0,832096519746182i$ $xc = -0,258483264163539 + 0,080261640425163i$	$fx = -3,917441226963803e-008 - 3,411888571669408e-008i$ $gx = -8,245944904672342e-008 - 3,275112370193511e-008i$ $hx = -2,451028269234712e-008 + 1,216569067707951e-009i$
	3	220	106,966	$xa = -0,312827239363878 - 0,277855080150431i$ $xb = -0,186349585117908 +$	$fx = 8,573386778287784e-009 - 7,281796843283672e-008i$ $gx = 2,450258307362674e-008$

				0,832096501818890i xc =-0,258483256005842 + 0,080261645164816i	-9,298057646045521e-008i hx =-6,814759601070364e-008 -2,571710755461254e-008i
	Prom.	225	109,027		
0,4	1	213	104,714	xa = 2,458145474737599 + 0,464996035604188i xb =-1,020120348270717 + 0,582257012231228i xc = 0,505305377431875 - 0,701862319017955i	fx = 1,469358856809322e-009 -7,836682414108509e-009i gx = 9,742996720518704e-008 +5,553920878753615e-009i hx =-1,064353938318163e-008 -3,074129395486125e-008i
	2	214	104,324	xa =-0,312827284686030 - 0,277855061030616i xb =-0,186349577272893 + 0,832096527944145i xc =-0,258483269970131 + 0,080261645451747i	fx =-2,675828464360208e-008 +2,673399901453877e-008i gx =-5,251934975802897e-008 +7,438987847940837e-008i hx =-5,120015522663834e-009 -6,061896318243498e-008i
	3	203	97,373	xa =-0,312827257923304 - 0,277855085036762i xb =-0,186349592532586 + 0,832096508017982i xc =-0,258483257640532 + 0,080261641827842i	fx =-6,673606506302221e-008 - 3,607563736629516e-008i gx =-2,479003802235980e-008 - 9,056393213136005e-008i hx = -3,432485939924845e-008 +1,743144295573984e-008i
	Prom.	210	102,137		
0,5	1	199	99,149	xa =-0,312827285652664 - 0,277855060643393i xb =-0,186349575662856 + 0,832096530841940i xc =-0,258483270010291 + 0,080261647172600i	fx = -2,159283035707560e-008 +2,869616666512087e-008i gx =-5,326035124042505e-008 +8,218585101360532e-008i hx = 4,307871748210346e-008 -2,881244220986901e-008i
	2	197	94,736	xa =-0,312827247206737 - 0,277855059687443i xb =-0,186349586808619 + 0,832096508359215i xc =-0,258483258122562 + 0,080261643990440i	fx = 3,303058437875706e-009 +5,790284585138750e-008i gx =1,601783505833510e-008 - 7,920909506076157e-008i hx =-3,018880345706521e-008 -2,132206144977999e-008i
	3	191	93,111	xa =-0,312827245391606 - 0,277855061836385i xb =-0,186349585428512 + 0,832096509870623i xc =-0,258483257047998 + 0,080261645746012i	fx = 1,151800294030281e-008 +4,809989695431227e-008i gx =1,783999525883928e-008 -8,213007873791867e-008i hx =2,527772302585163e-008 +1,171041574821174e-008i
	Prom.	196	95,665		
0,6	1	210	101,122	xa =-0,312827259967269 - 0,277855094320662i xb =-0,186349591681212 + 0,832096513570401i xc =-0,258483260922358 + 0,080261640026229i	fx =-2,847947067063927e-008 - 7,366761312077585e-008i gx =-6,863410551005700e-008 - 6,440345690084115e-008i hx =-3,527439851325198e-008 - 2,062423298809790e-008i
	2	200	97,524	xa =-0,367596856647345 - 2,043359214427108i xb = 0,535493797791826 - 1,315120816271234i xc =-0,232658217949137 - 0,453004831071390i	fx = -1,117023540331275e-008 +6,416583531176912e-008i gx =3,985078933510522e-011 - 8,285544805808343e-008i hx =3,659972236391695e-008 - 5,526581325909774e-008i
	3	210	100,589	xa =-0,312827283059412 - 0,277855065811471i xb =-0,186349584402678 + 0,832096529011380i xc =-0,258483269562791 + 0,080261642107862i	fx =-2,850670355325491e-008 +5,139370795603782e-008i gx =-8,261330330938677e-008 +4,183946433755637e-008i hx =-2,604520155102819e-009 -3,794184877392581e-008i
	Prom.	207	99,745		
0,7	1	197	94,27	xa =-0,312827269587561 - 0,277855055639803i	fx = 5,313128054185157e-008 +6,545248609768350e-008i

				xb =-0,186349576559380 + 0,832096529345542i xc =-0,258483268470594 + 0,080261645393719i	gx =-5,316064166649426e-008 +5,003284941906827e-008i hx = 5,054227258938226e-008 -7,004757751616353e-008i
	2	229	109,120	xa = 0,495044029490876 + 2,745690728913358i xb =-0,040387906828264 - 0,826624946022809i xc =-0,956137495781180 + 0,617620811640264i	fx =-5,636758260152419e-008 -6,875938574069096e-008i gx =6,187259327106176e-008 -3,597816000677767e-008i hx =3,654538716091338e-008 +3,823209659969962e-008i
	3	217	104,087	xa =-0,312827276788535 - 0,277855051138129i xb =-0,186349573297199 + 0,832096530177910i xc =-0,258483270885060 + 0,080261648199776i	fx = 2,764098633312173e-008 +5,488086429750183e-008i gx=-3,383198432516110e-008 +8,576047982877810e-008i hx =1,223749368506333e-008 -4,314610535516295e-008i
	Prom.	214	102,511		
0,8	1	232	113,030	xa =-0,312827255940937 - 0,277855064773971i xb =-0,186349572317282 + 0,832096513227790i xc =-0,258483262085877 + 0,080261655049480i	fx =-3,618599420107671e-009 -6,679864139602643e-008i gx =5,749964682166819e-008 +1,321018583411160e-008i hx=-1,079955680438616e-008 +7,866669804457160e-008i
	2	212	101,546	xa =-0,312827247454431 - 0,277855057056943i xb =-0,186349578062674 + 0,832096509193588i xc =-0,258483258795553 + 0,080261650113665i	fx =1,259369097894592e-008 +1,323801812613823e-008i gx =5,254460022641183e-008 -4,007874332412342e-008i hx=-6,559923271254320e-009 +1,442206354340669e-008i
	3	233	112,812	xa =-0,367596868140952 - 2,043359202429211i xb =0,35493795517237 - 1,315120809857490i xc =-0,232658221929720 - 0,453004827745671i	fx=-2,597591186592752e-008 +3,634026768395415e-008i gx =1,367088042769638e-008 +2,756210948362536e-008i hx=-3,558133343517511e-009 +4,099762307419042e-009i
	Prom.	226	109,129		
0,9	1	248	119,255	xa =-0,312827279708680 - 0,277855076367637i xb =-0,186349583597536 + 0,832096523906953i xc =-0,258483265449505 + 0,080261647758489i	fx =-7,524106554468446e-008 -2,358370898569717e-008i gx =-4,099688188929918e-008 +1,716022479669732e-008i hx = 3,360883216174671e-008 +9,115706145479408e-008i
	2	264	128,078	xa =-0,312827263678801 - 0,277855044189704i xb =-0,186349567757334 + 0,832096520587022i xc =-0,258483267209711 + 0,080261654069772i	fx = 3,195070963357694e-008 +2,418537325965531e-008i gx =4,704321021620217e-008 +6,989446710292668e-008i hx =-2,086319828009664e-008 - 2,612840610538569e-009i
	3	257	124,447	xa =-0,312827245762071 - 0,277855080122131i xb =-0,186349581110163 + 0,832096500958983i xc =-0,258483254199191 + 0,080261646571732i	fx =-2,318632458253589e-008 - 8,817894703305740e-008i gx = 3,686465621655088e-008 - 8,023288566860742e-008i hx=-1,546925876283467e-008 - 5,105297162799616e-008i
	Prom.	256	123,926		

Fuente: autores del proyecto.

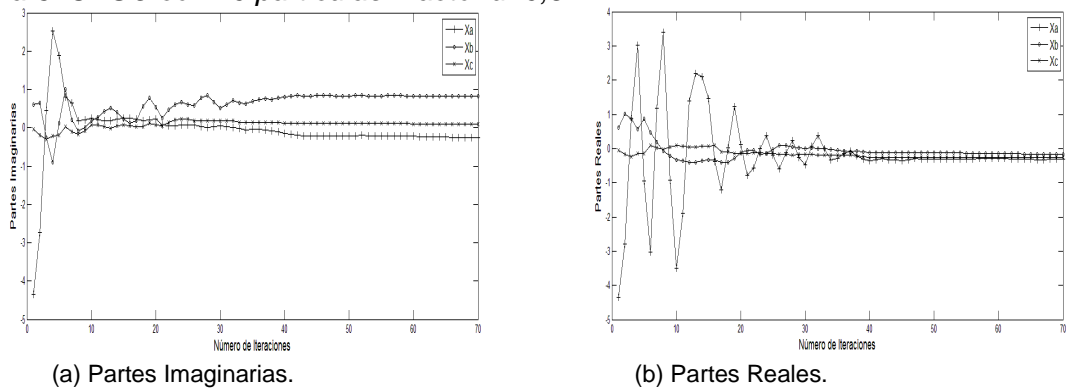
Figura 5. Tiempo y número de iteraciones vs factor de unificación.

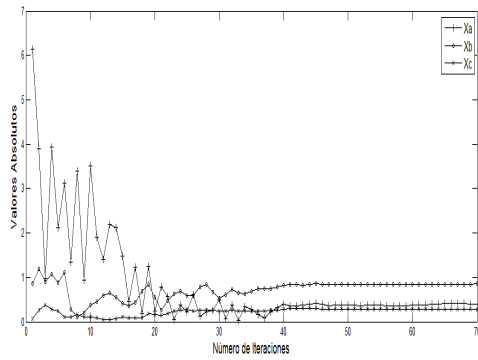


Fuente: Autores del Proyecto.

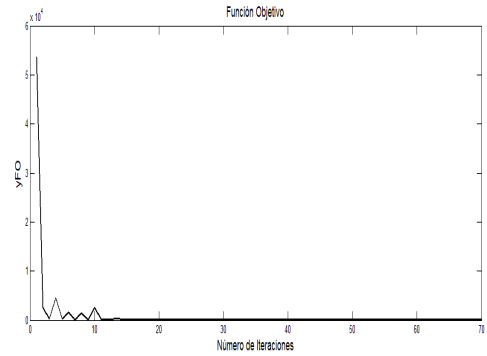
Para UPSO se observa un mejor comportamiento cuando el factor de unificación es igual a 0,5, utilizando un menor tiempo de cómputo e iteraciones que para otro valor que tome este parámetro. A mayor cantidad de partículas, menor cantidad de iteraciones pero mayor consumo de tiempo al hallar la solución a un sistema de ecuaciones. La figura 6 muestra los resultados para una simulación con un factor de unificación igual a 0,5.

Figura 6. UPSO con 20 partículas. Factor $u=0,5$.





(c) Valores Absolutos.



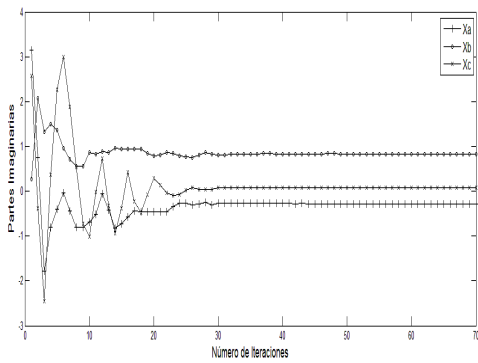
(d) Función Objetivo.

Fuente: Autores del Proyecto.

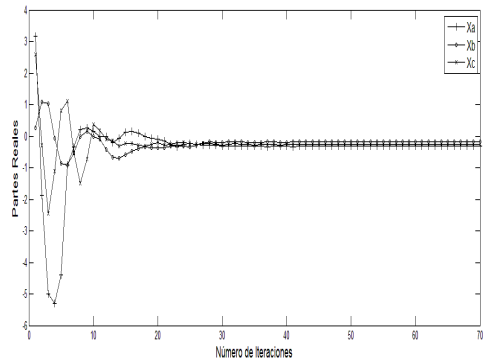
b) 50 Partículas

La figura 7 expone el resultado para una simulación con un factor de unificación igual a 0,6.

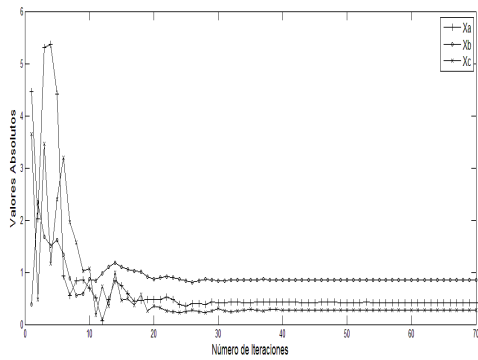
Figura 7. UPSO con 50 partículas. Factor $u=0,6$.



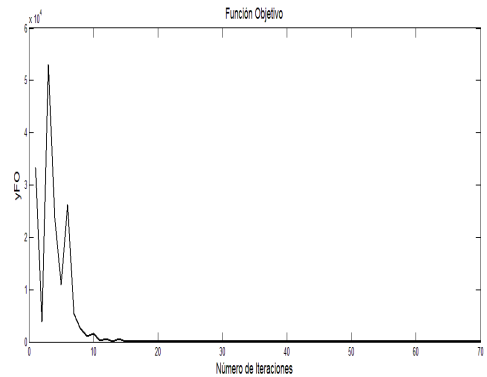
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.



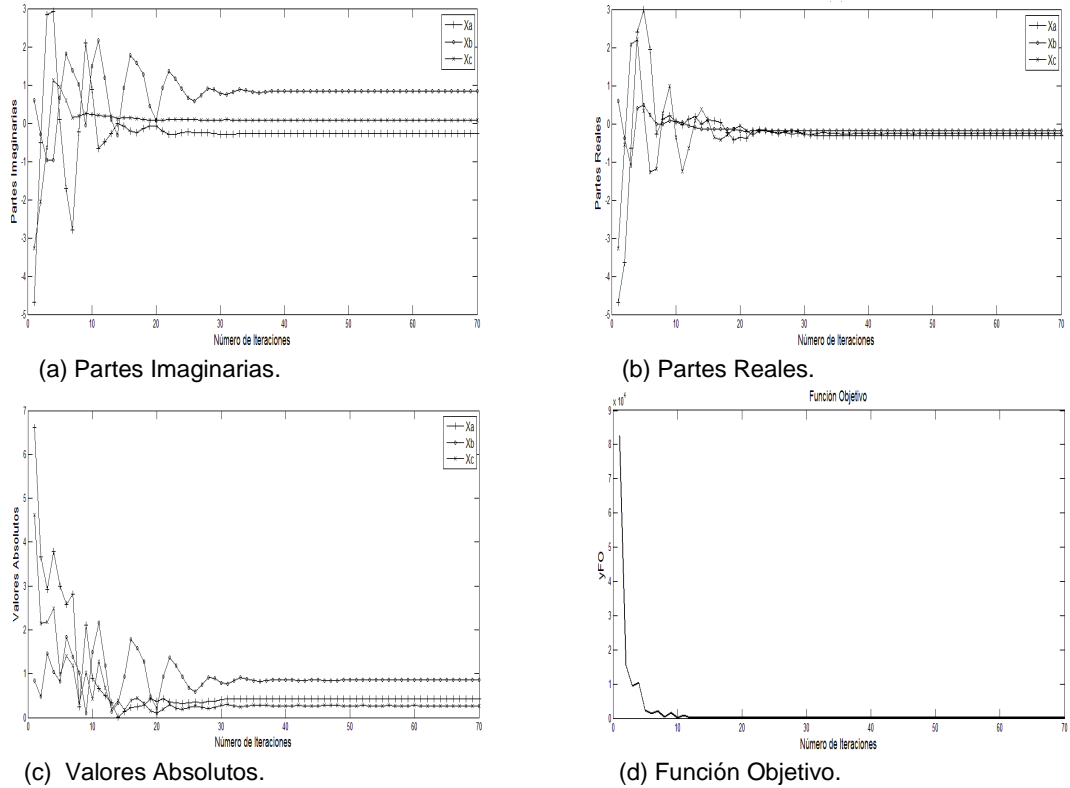
(d) Función Objetivo.

Fuente: Autores del Proyecto.

c) 100 Partículas

La figura 8 presenta los resultados para una simulación con el factor de unificación igual a 0,5.

Figura 8. UPSO con 100 partículas. Factor $u=0,5$.



Fuente: Autores del Proyecto.

5.1.3. UPSO LINEAL

Se dice que es lineal debido a que el factor de unificación muestra un comportamiento dado por [2][14]:

$$u(t) = \frac{t}{N} \quad (19)$$

donde $t \in Z$ para la iteración actual y N es el número total de iteraciones.

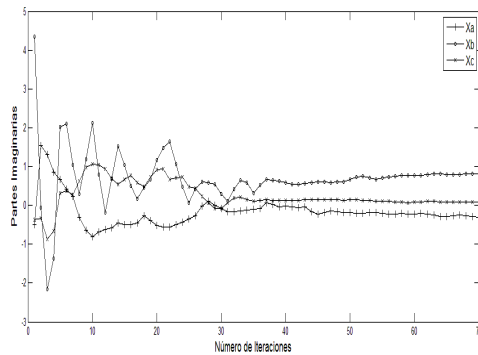
a) 20 Partículas

Tabla 3. Sistema de tres Ecuaciones. UPSO Lineal

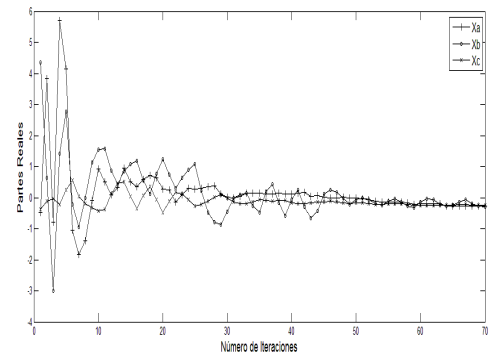
Prueba	Iter.	Tiempo[s]	Raíces	Función Evaluada
1	301	146,069	$x_a = -0,312827280044972 - 0,277855057382771i$ $x_b = -0,186349575833448 + 0,832096530985046i$ $x_c = -0,258483271114622 + 0,080261648192182i$	$fx = 4,325125813053887e-009 + 3,676300064370253e-008i$ $gx = -4,425552280196143e-008 + 8,127667694424190e-008i$ $hx = 1,170635233194162e-008 + 3,984039764759473e-010i$
2	294	145,782	$x_a = -0,312827244573505 - 0,277855058249466i$ $x_b = -0,186349569711442 + 0,832096506836295i$ $x_c = -0,258483255357974 + 0,080261653230149i$	$fx = 2,514920943719190e-008 - 2,999015946691941e-008i$ $gx = 7,906236509924725e-008 - 3,061930031744709e-008i$ $hx = 7,753170105928575e-008 - 3,362201361767347e-008i$
3	292	144,249	$x_a = 0,495044026454341 + 2,745690726789157i$ $x_b = -0,040387900260851 - 0,826624941187528i$ $x_c = -0,956137492754327 + 0,617620804452198i$	$fx = 5,125615398782202e-008 + 7,207068186687593e-008i$ $gx = 2,683936095593253e-008 - 3,798918957187425e-008i$ $hx = 2,938354626280670e-008 + 2,915460228791744e-008i$
Prom.	296	145,366		

Fuente: Autores del Proyecto.

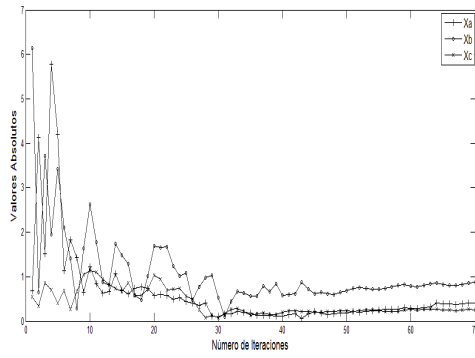
Figura 9. UPSO Lineal con 20 partículas.



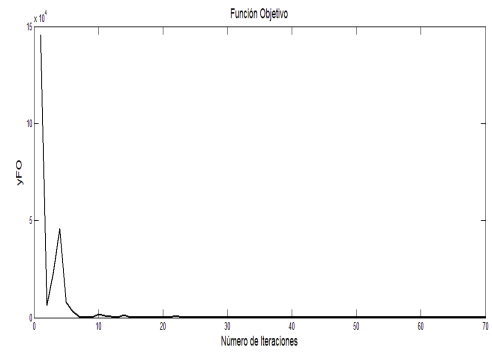
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.



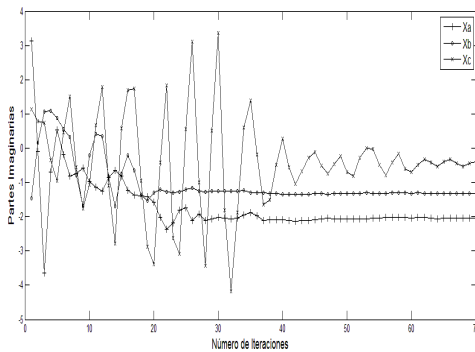
(d) Función Objetivo.

Fuente: Autores del Proyecto.

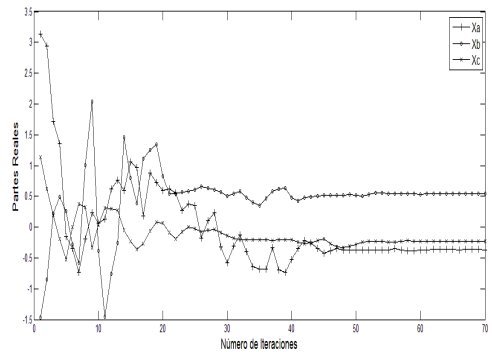
La figura 9 muestra el comportamiento en la solución para una simulación cuando el factor de unificación presenta un comportamiento lineal creciente, presentando un aumento en el número de iteraciones así como en el tiempo del cálculo de las raíces que cuando se utiliza UPSO con el factor de unificación establecido para 0,5.

b) 50 Partículas

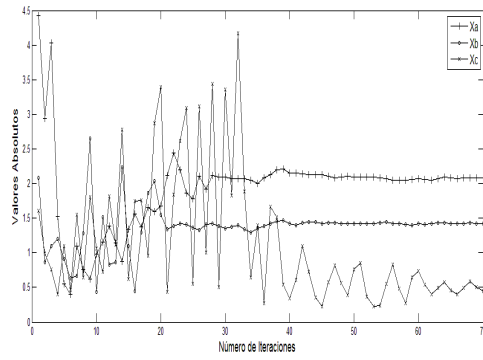
Figura 10. UPSO Lineal con 50 partículas.



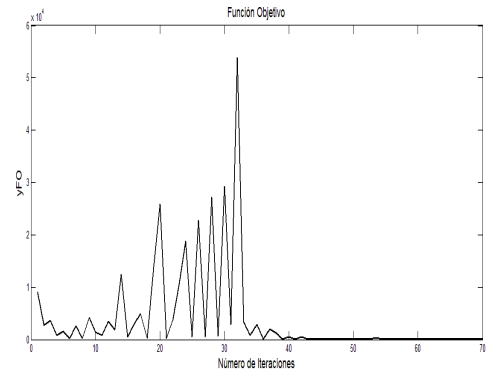
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.

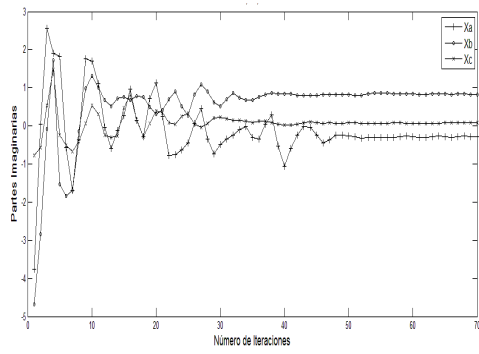


(d) Función Objetivo.

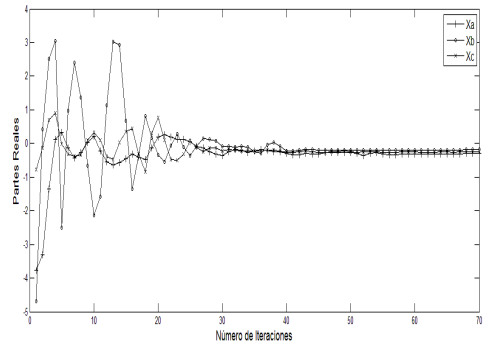
Fuente: Autores del Proyecto.

c) 100 Partículas

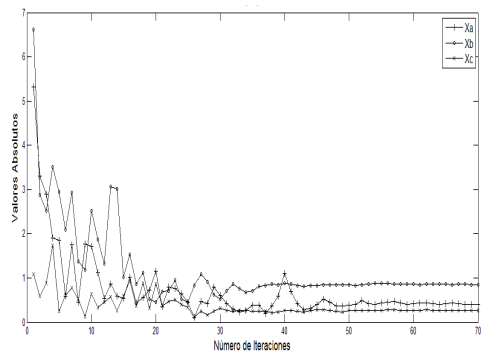
Figura 11. UPSO Lineal con 100 partículas.



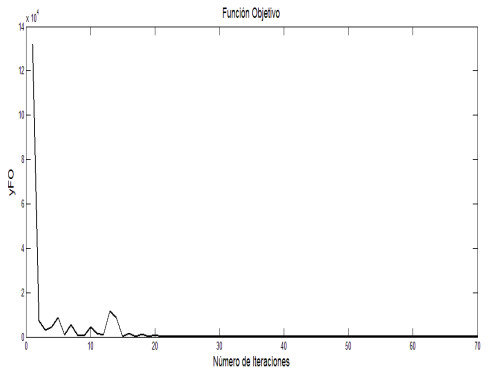
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.



(d) Función Objetivo.

Fuente: Autores del Proyecto.

5.1.4. UPSO EXPONENCIAL

Por otra parte, el modelo exponencial viene dado por [2][14]

$$u(t) = \exp\left(\frac{t \times \log(2)}{N}\right) - 1 \quad (20)$$

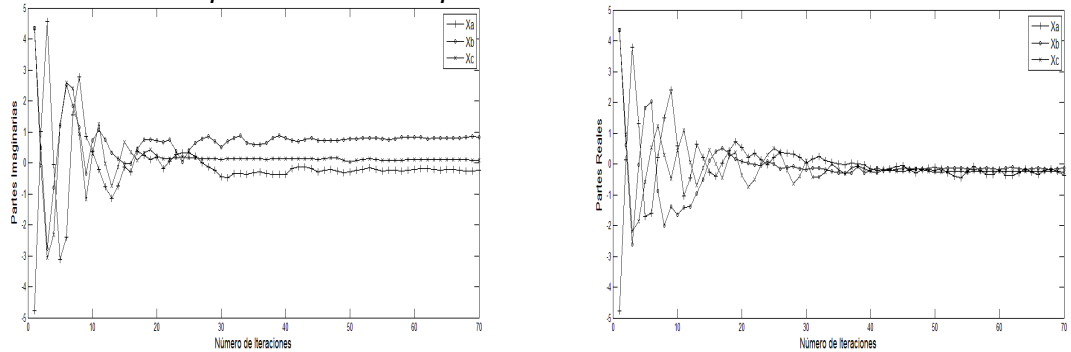
a) 20 Partículas

Tabla 4. Sistema de tres Ecuaciones. UPSO Exponencial

Prueba	Iter.	Tiempo[s]	Raíces	Función Evaluada
1	292	154,197	$x_a = -0,312827269850655$ $- 0,277855042015802i$ $x_b = -0,186349569720944$ $+ 0,832096515711475i$ $x_c = -0,258483264110246$ $+ 0,080261654711517i$	$fx = -4,236886169683984e-008$ $+3,685031113587201e-008i$ $gx = 7,339573171449842e-008$ $+5,058239294086775e-008i$ $hx = -1,370479640883104e-008$ $+2,625449901927368e-008i$
2	286	149,957	$x_a = -0,312827272982945$ $- 0,277855066793026i$ $x_b = -0,186349575910835$ $+ 0,832096529890332i$ $x_c = -0,258483271209184$ $+ 0,080261650290997i$	$fx = 2,324792619523564e-008 -$ $2,242480667824509e-008i$ $gx = -3,889466793083329e-008$ $+7,384986333569543e-008i$ $hx = -1,060255883089667e-008$ $+5,989883611334790e-008i$
3	258	129,556	$x_a = -0,312827260697721$ $- 0,277855084311970i$ $x_b = -0,186349591848953$ $+ 0,832096513373068i$ $x_c = -0,258483259091397$ $+ 0,080261638796746i$	$fx = -3,302225110024537e-008 -$ $9,430074499050534e-009i$ $gx = -6,334666435847680e-008 -$ $7,589730222079538e-008i$ $hx = 1,040397368257118e-008 -$ $5,723485596575983e-008i$
Prom.	279	144,569		

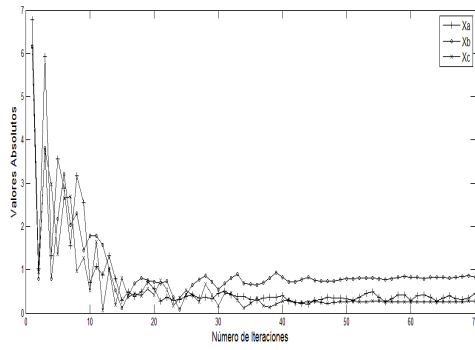
Fuente: Autores del Proyecto.

Figura 12. UPSO Exponencial con 20 partículas.

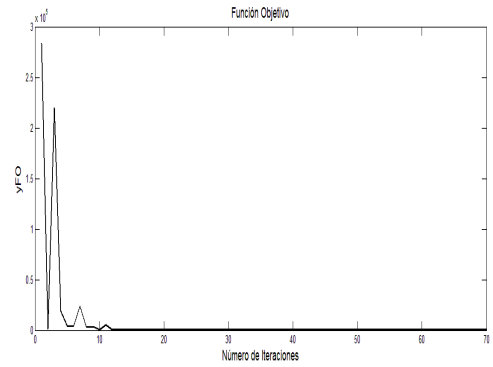


(a) Partes Imaginarias.

(b) Partes Reales.



(c) Valores Absolutos.

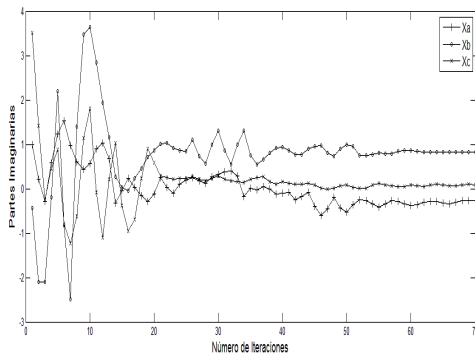


(d) Función Objetivo.

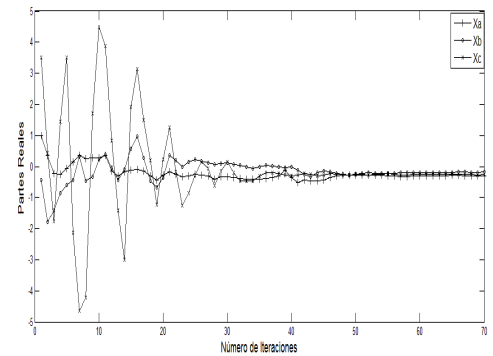
Fuente: Autores del Proyecto.

b) 50 Partículas

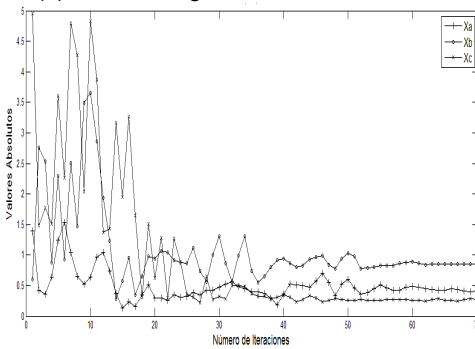
Figura 13. UPSO Exponencial con 50 partículas.



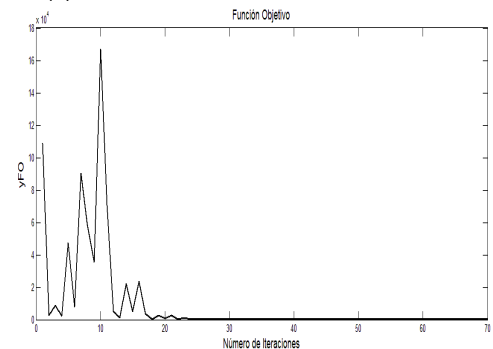
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.

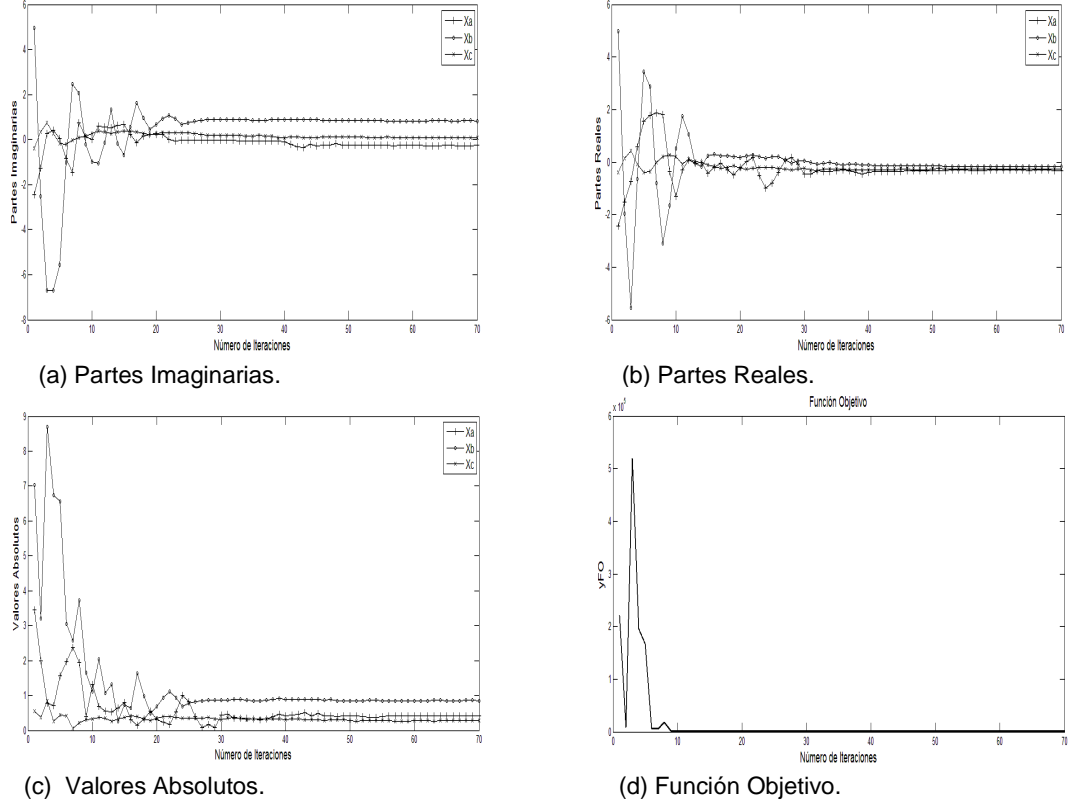


(d) Función Objetivo.

Fuente: Autores del Proyecto.

c) 100 Partículas

Figura 14. UPSO Exponencial con 100 partículas.



Fuente: Autores del Proyecto.

Bajo un comportamiento exponencial en el factor de unificación, las demandas de recursos se hacen mayores al incrementar tanto el número de iteraciones como el tiempo al hallar las raíces que solucionan el sistema.

5.2. CONFRONTACIÓN ENTRE TRES MÉTODOS

La tabla 5 presenta los datos más relevantes para comparar el rendimiento de un método frente a otro. Aun así con estas características no es suficiente para definir cuál es mejor, pues pruebas realizadas utilizando Newton Raphson Multivariable [18][19] permitieron ver que tanto el tamaño del sistema de

ecuaciones como los puntos de arranque influyen sobre la búsqueda de la potencial solución. Así por ejemplo para el sistema de 3x3 Newton invierte alrededor de 2 segundos y 6 iteraciones para resolverlo, siendo mejor que los métodos metaheurísticos antes mencionados.

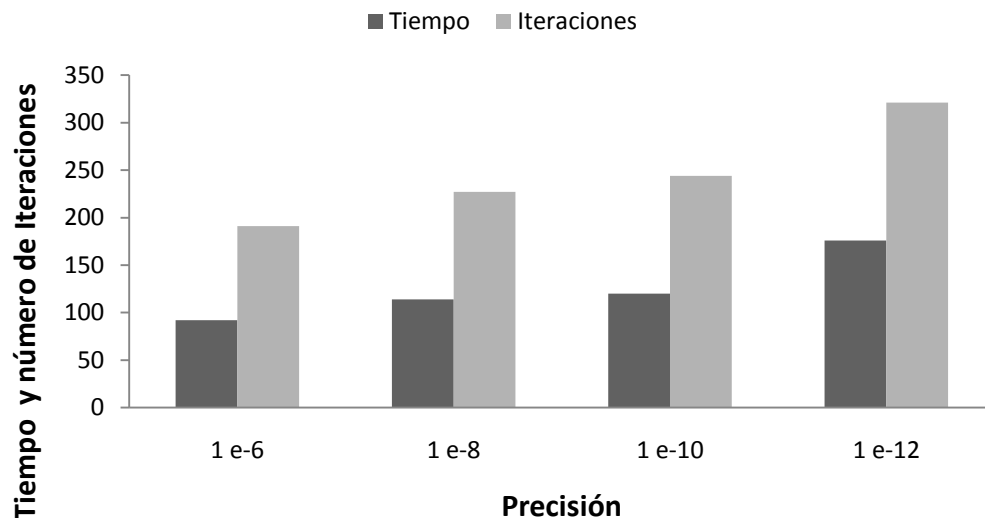
Tabla 5. Sistema de tres Ecuaciones

Partículas	PSO		UPSO		UPSO LINEAL		UPSO EXPONENCIAL	
	Iter.	Tiempo[s]	Iter.	Tiempo[s]	Iter.	Tiempo[s]	Iter.	Tiempo[s]
20	283	142	196	96	296	145	279	144
50	222	268	157	179	230	268	225	270
100	207	459	129	290	199	465	207	467

Fuente: Autores del Proyecto.

Con el fin de obtener una solución más aproximada, se realizaron pruebas variando la precisión, exhibiendo un aumento en el número de iteraciones y en el tiempo de computó al hallar las raíces del sistema expuesto en las ecuaciones (16)-(18).

Figura 15. Efecto de la Precisión.



Fuente: Autores del Proyecto.

5.3. SISTEMA DE CINCO ECUACIONES

$$f(x_a, x_b, x_c, x_d, x_e) = 10j \times x_a^3 + (8 - 7j) \times x_b - (3 + 8j) \times x_c^5 + 22 \times x_d + 18j \times x_e^2 + 9 - 5j = 0 \quad (21)$$

$$g(x_a, x_b, x_c, x_d, x_e) = (20 - 6j) \times x_a^4 + 42j \times x_b^2 + 7j \times x_c^2 + 2j \times x_d^2 - (14 + 6j) \times x_e^4 + 31 - j = 0 \quad (22)$$

$$h(x_a, x_b, x_c, x_d, x_e) = j \times x_a^2 + 5 \times x_b^4 + 24 \times x_c + (16 + 4j) \times x_d^4 - 6 \times x_e^5 + 13 = 0 \quad (23)$$

$$k(x_a, x_b, x_c, x_d, x_e) = 5,3 \times x_a^5 + (2 + 2j) \times x_b^3 + (5 - 13j) \times x_c^3 + 7j \times x_d^5 - 9j \times x_e^3 - 20j = 0 \quad (24)$$

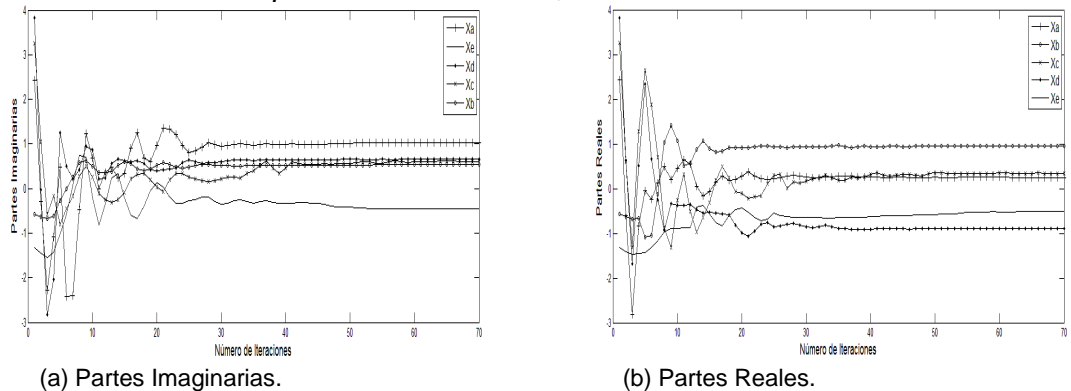
$$l(x_a, x_b, x_c, x_d, x_e) = (5 - 3j) \times x_a + (31 + 12j) \times x_b^5 - (4 - 7j) \times x_c^4 + (4 - 4j) \times x_d^3 + (2,5 + 6j) \times x_e + 35 - 14j = 0 \quad (25)$$

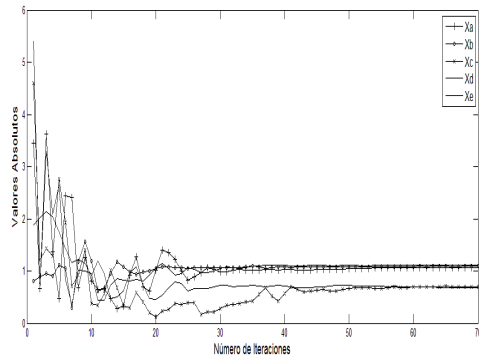
Para este sistema, debido a los resultados obtenidos anteriormente, solo se hicieron simulaciones para UPSO dado que presentó un mejor rendimiento en la búsqueda de la solución del sistema frente a PSO convencional. Se utilizó un factor de unificación de $u=0,5$ y $u=0,6$.

a) 50 partículas

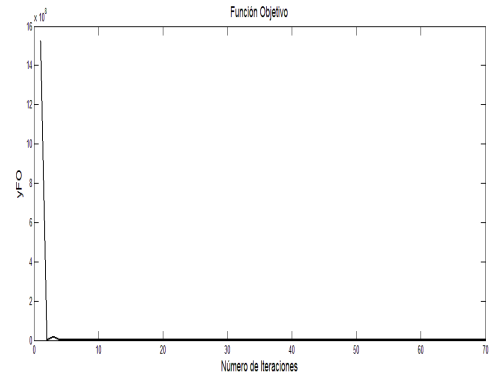
A continuación se muestran gráficas para una simulación en cada valor respectivo del factor de unificación mostrado anteriormente.

Figura 16. UPSO con 50 partículas. Factor $u=0,5$.





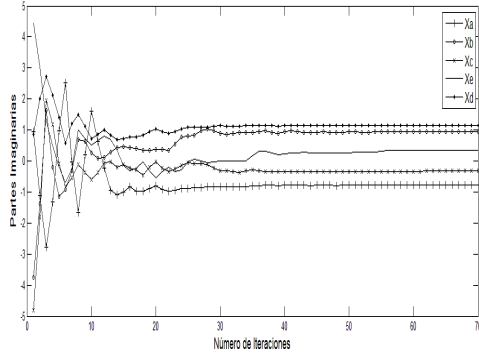
(c) Valores Absolutos.



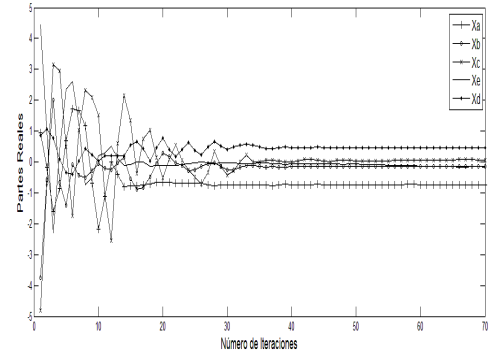
(d) Función Objetivo.

Fuente: Autores del Proyecto.

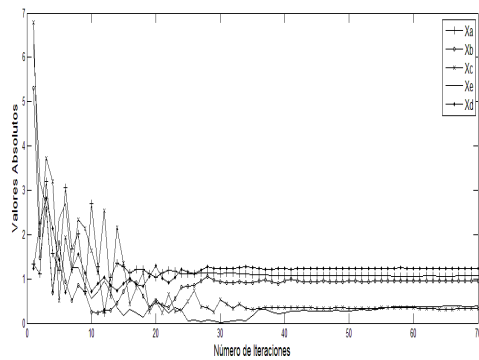
Figura 17. UPSO con 50 partículas. Factor $u=0,6$.



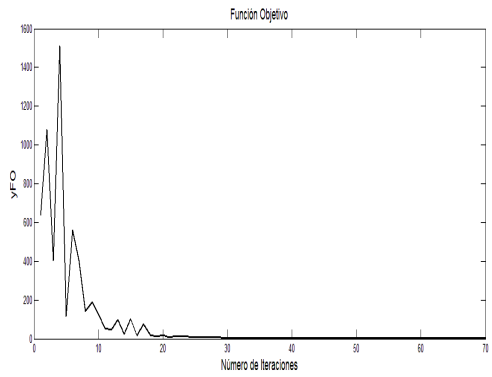
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.

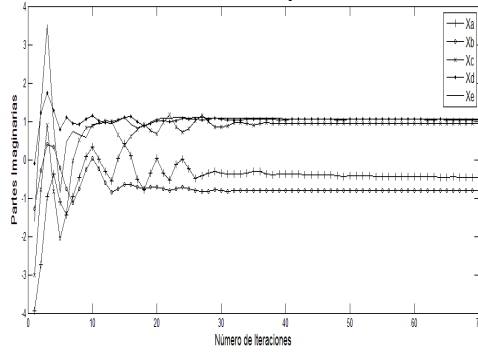


(d) Función Objetivo.

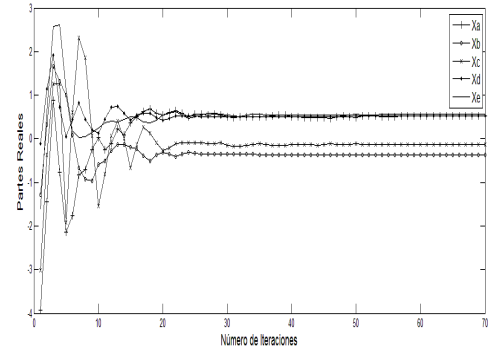
Fuente: Autores del Proyecto.

b) 100 Partículas

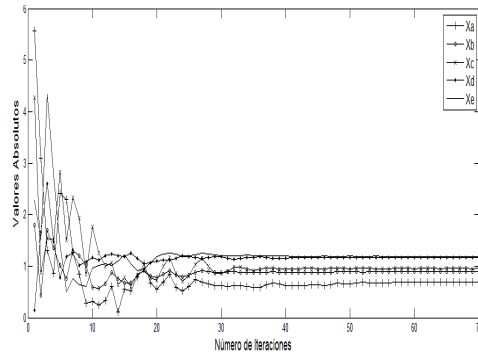
Figura 18. UPSO con 100 partículas. Factor $u=0,5$.



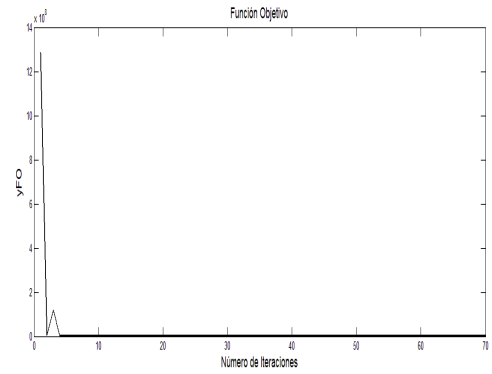
(a) Partes Imaginarias.



(b) Partes Reales.



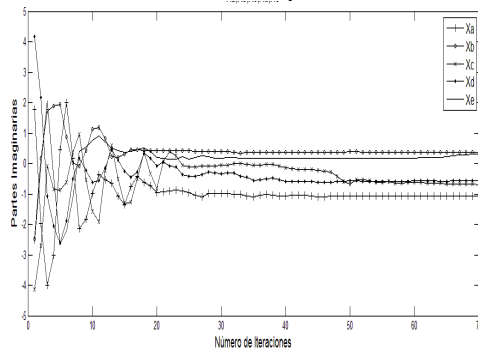
(c) Valores Absolutos.



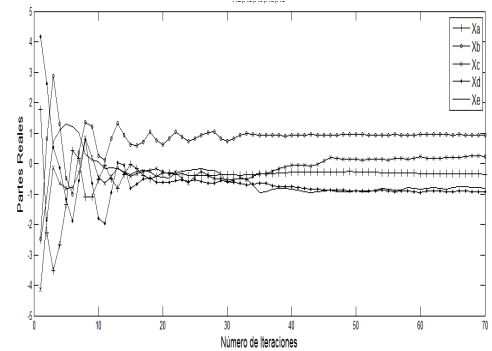
(d) Función Objetivo.

Fuente: Autores del Proyecto.

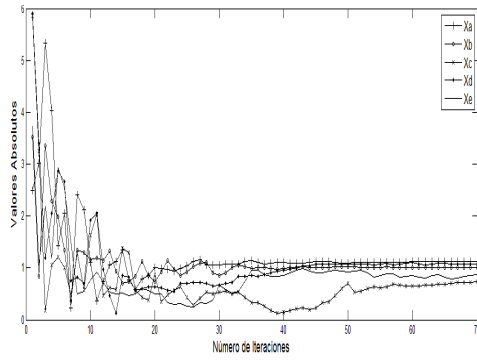
Figura 19. UPSO con 100 partículas. Factor $u=0,6$.



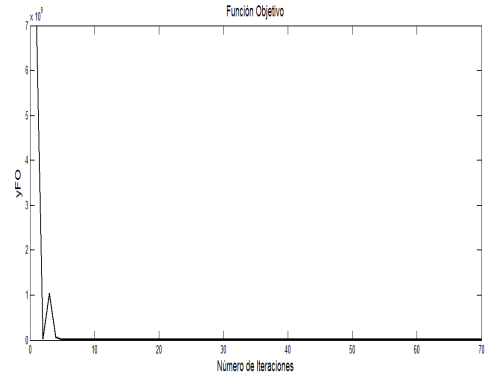
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.

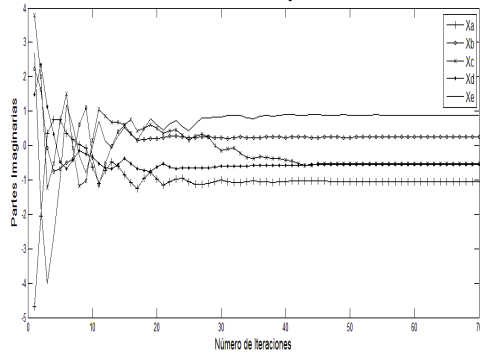


(d) Función Objetivo.

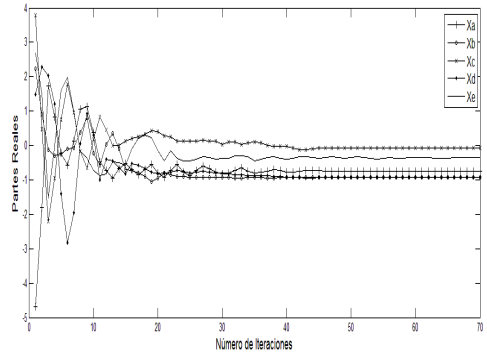
Fuente: Autores del Proyecto.

c) 150 Partículas

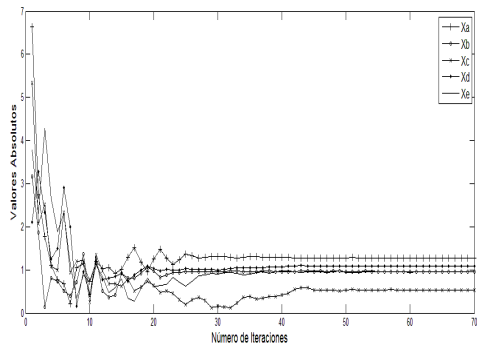
Figura 20. UPSO con 150 partículas. Factor $u=0,5$.



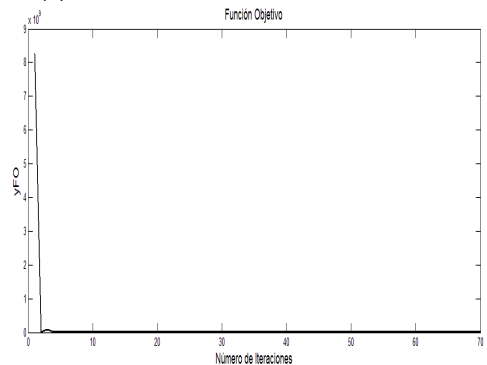
(a) Partes Imaginarias.



(b) Partes Reales.



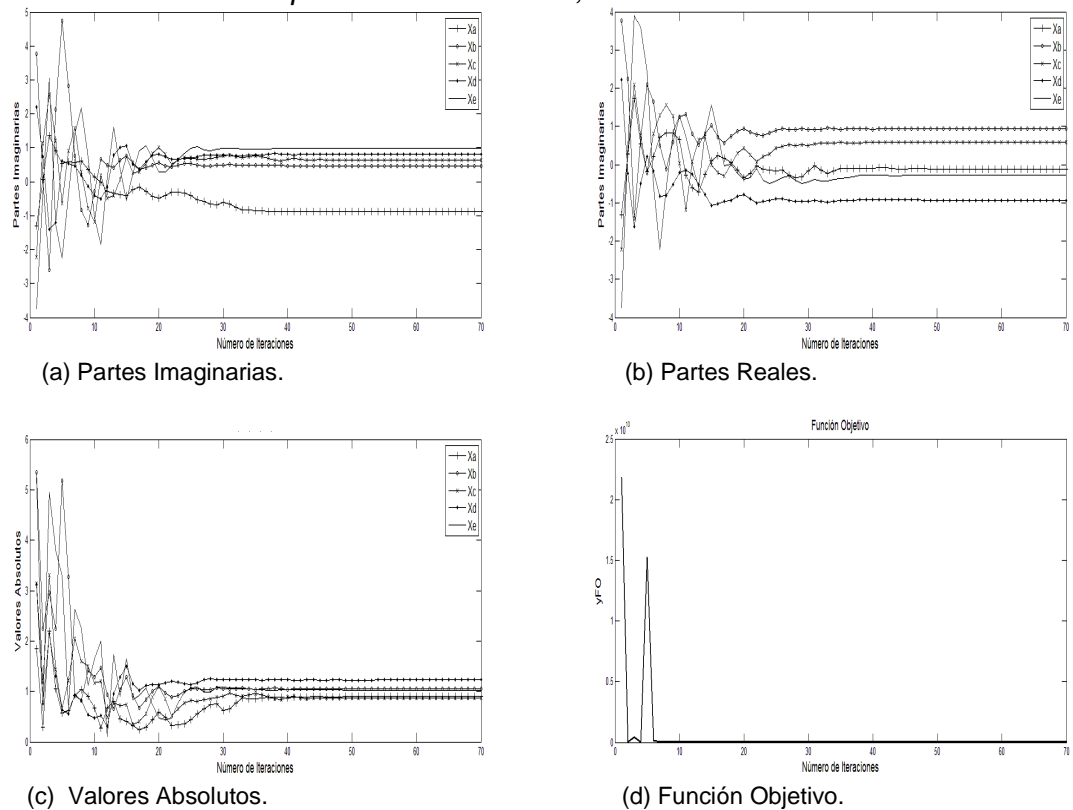
(c) Valores Absolutos.



(d) Función Objetivo.

Fuente: Autores del Proyecto.

Figura 21. UPSO con 150 partículas. Factor $u=0,6$.



Fuente: Autores del Proyecto.

Nuevamente, UPSO muestra un mejor comportamiento con un factor de unificación $u=0,5$, independiente del número de partículas del enjambre. Para este sistema de ecuaciones también se hicieron pruebas utilizando el método de Newton, pero en este caso no se obtuvo ninguna solución, es decir, el método tradicional multivariable no convergió a ningún posible conjunto solución, dado que dependía de valores iniciales definidos por el usuario. A manera de prueba se utilizaron los resultados obtenidos mediante UPSO con el fin de dar un punto de partida inicial adecuado para el método de Newton Raphson, sin embargo no se llegó a los resultados esperados.

5.4. SISTEMA DE SIETE ECUACIONES

En este sistema de ecuaciones, se hicieron simulaciones para el algoritmo UPSO con un valor de unificación establecido en 0,5 y se vario la cantidad de partículas del enjambre.

$$f(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = (3 + j) \times x_a^2 - 5j \times x_b^3 + (9 - 3j) \times x_c + 8 \times x_d^5 - (5 + 3,2j) \times x_e^4 + x_f^7 - (15 + 2j) \times x_g^6 - 8 = 0 \quad (26)$$

$$g(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = 2j \times x_a + (4 - 2j) \times x_b^2 + 7j \times x_c^3 - 3 \times x_d^7 - j \times x_e^6 + (5 - 3j) \times x_f^5 + 25j \times x_g^4 + j = 0 \quad (27)$$

$$h(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = x_a^7 + 5j \times x_b^4 - (7 + 2j) \times x_c^6 - (2 + j) \times x_d - 9 \times x_e^5 + 18j \times x_f^3 - (6 + 3,2j) \times x_g^2 + 22 = 0 \quad (28)$$

$$k(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = j \times x_a^3 - x_b^7 + 5j \times x_c^4 - (4 - 7j) \times x_d^6 + 15 \times x_e - (18 + 3j) \times x_f^2 - x_g^5 - 12 = 0 \quad (29)$$

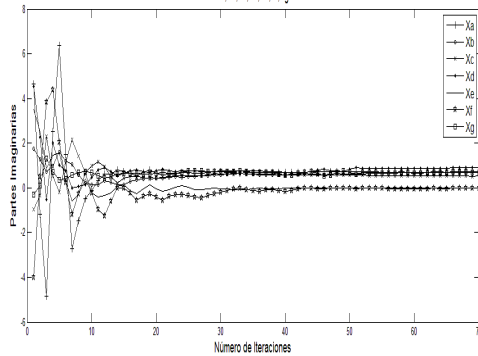
$$l(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = x_a^6 + 3j \times x_b^5 + (16 + 4j) \times x_c^2 + (7 - 3j) \times x_d^4 + j \times x_e^3 + 21 \times x_f - (5 - 3j) \times x_g^7 + 3 + 15j = 0 \quad (30)$$

$$m(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = 3 \times x_a^4 + 5j \times x_b + 4j \times x_c^5 - 5 \times x_d^2 + 9 \times x_e^7 + 8j \times x_f^6 - 20 \times x_g^3 - 32 = 0 \quad (31)$$

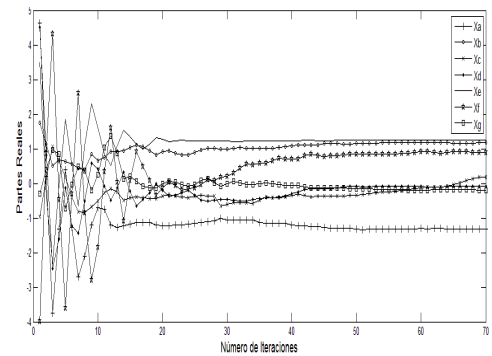
$$n(x_a, x_b, x_c, x_d, x_e, x_f, x_g) = (4 - j) \times x_a^5 - x_b^6 + 3 \times x_c^7 + (25 - 3j) \times x_d^3 - 14 \times x_e^2 + 2 \times x_f^4 + (3 - 2j) \times x_g - 18 = 0 \quad (32)$$

a) 150 Partículas

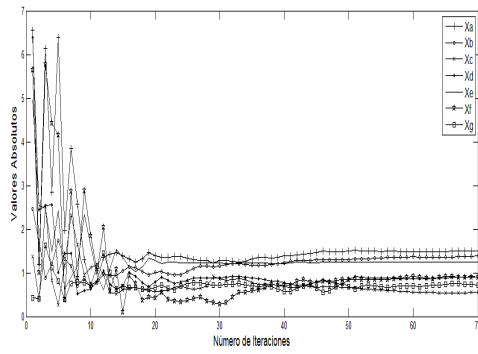
Figura 22. UPSO con 150 partículas. Factor $u=0,5$.



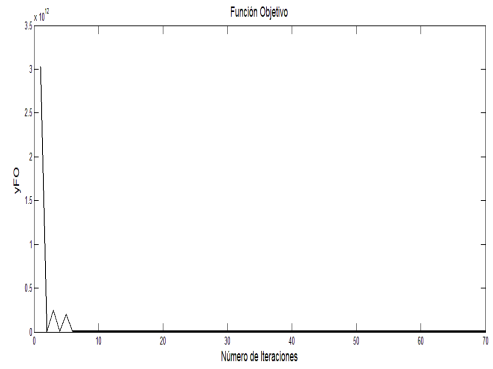
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.

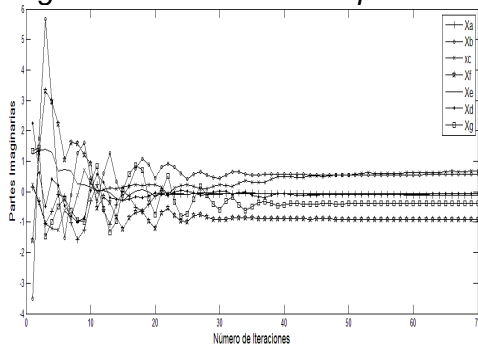


(d) Función Objetivo.

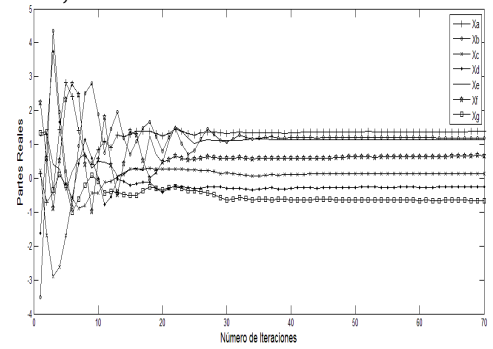
Fuente: Autores del Proyecto.

b) 200 Partículas

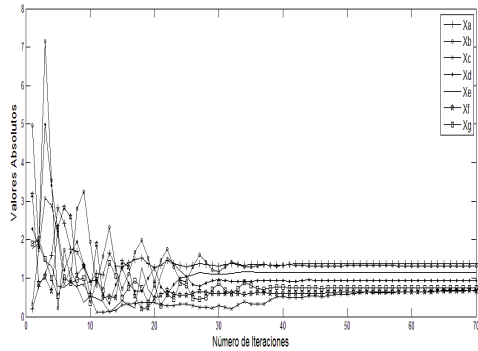
Figura 23. UPSO con 200 partículas. Factor $u=0,5$.



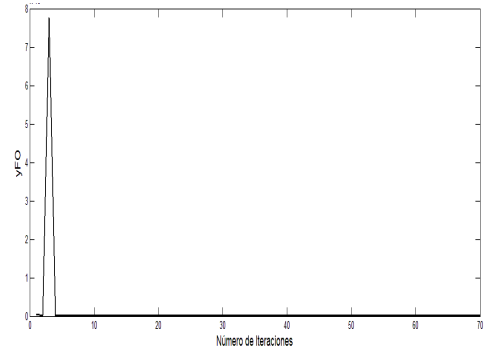
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.

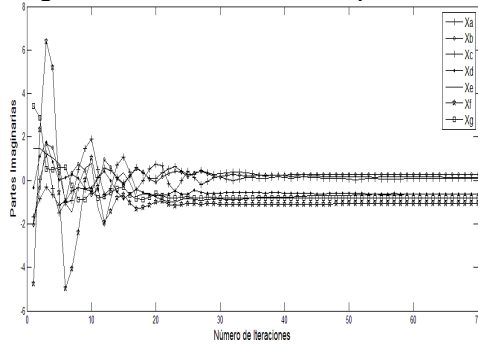


(d) Función Objetivo.

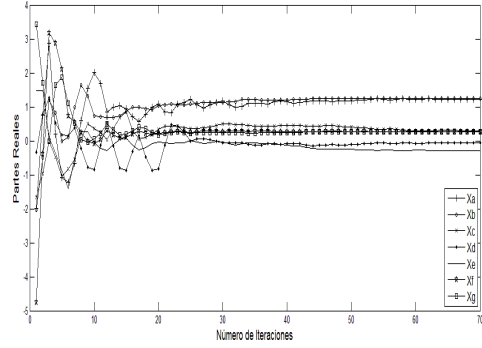
Fuente: Autores del Proyecto.

c) 250 Partículas

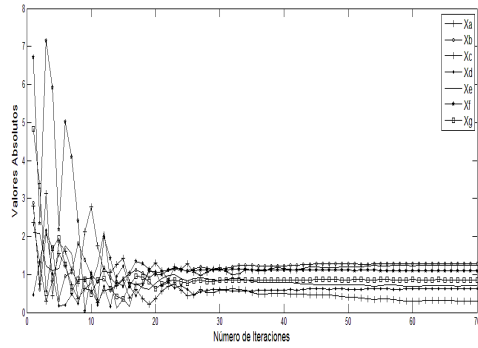
Figura 24. UPSO con 250 partículas. Factor $u=0,5$.



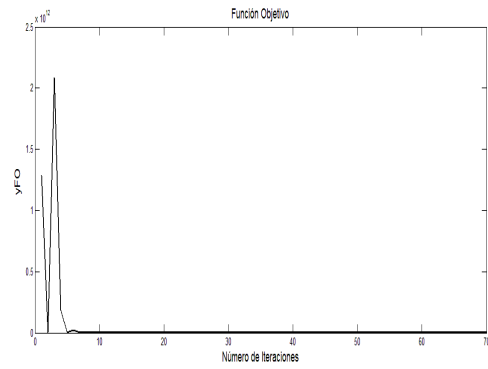
(a) Partes Imaginarias.



(b) Partes Reales.



(c) Valores Absolutos.



(d) Função Objetivo.

Fuente: Autores del Proyecto.

A medida que incrementa la complejidad de los sistemas, se hace necesario aumentar el número de partículas del enjambre, evitando que el algoritmo encuentre mínimos locales que no necesariamente sean la solución al sistema. UPSO muestra un mejor comportamiento cuando el factor de unificación es igual a 0,5, comparado frente a PSO. Por su parte Newton-Raphson multivariable no converge debido a la complejidad del sistema, indiferente de sus puntos de arranque.

6. OBSERVACIONES Y CONCLUSIONES

Dada la complejidad de los sistemas de ecuaciones no lineales, UPSO es un método alternativo y eficiente para la solución de sistemas no lineales con coeficientes complejos, pues debido a sus capacidades de explotación y exploración en el espacio de búsqueda complejo, presenta una mejora considerable frente a PSO y Newton Raphson. El éxito de UPSO frente a las otras herramientas computacionales, está basado a parte de su sencilla implementación en que no necesita de cálculos matemáticos exigentes como una matriz jacobiana, ni tampoco de un conocimiento previo o suposición de las soluciones del sistema como si lo demanda Newton Raphson multivariable para alcanzar resultados satisfactorios. El hecho de variar parámetros en el algoritmo, tales como aumentar el número de partículas o el factor de unificación, provocó un incremento del número de iteraciones y como consecuencia del tiempo de cómputo. La velocidad de convergencia del algoritmo depende de la complejidad de los sistemas (cantidad de variables y en especial el grado de los polinomios), y de las posiciones iniciales del enjambre, las cuales son distribuidas aleatoriamente. Los mejores resultados para UPSO se obtuvieron cuando el factor de unificación tomó valores para $u=0.5$ y $u=0.6$, distribuyendo de manera equitativa la información tanto personal como de todas las partículas que conforman el enjambre. Gráficamente se pudo constatar para sistemas 3×3 , 5×5 y 7×7 que UPSO presenta mayor velocidad de convergencia frente a PSO, mostrando cómo las partículas en sus partes real e imaginaria, tendían rápidamente al valor de cada una de las soluciones del sistema de ecuaciones. A partir de la iteración sesenta las partículas oscilaban muy cerca de las soluciones del sistema; solo bastaba cumplir con la condición de precisión para encontrarlas. Newton Raphson tuvo mejor desempeño frente a los métodos metaheurísticos, para sistemas de baja complejidad como el de 3×3 . UPSO requirió de menor cantidad de iteraciones, bajo tiempo de cómputo y mayor aproximación que los demás métodos. Debido a

la gran cantidad de soluciones que presentan los sistemas de ecuaciones expuestos, para un mismo sistema se encontraron diferentes conjuntos solución que demandaban más recursos que otros. Los recursos de memoria con los que se disponía fueron insuficientes para resolver sistemas de orden mayor.

7. REFERENCIAS

- [1] M. Hromcik, M. Sebek y J. Jezek, "Complex polynomials in communications: Motivation, algorithms, software", presentado al *IEEE. International symposium on computer aided control system design proceedings*, Glasgow, Scotland, pp. 1-2, Sept. 2002.
- [2] Parsopoulos, K.E., Vrahatis, M.N., "Unified Particle Swarm Optimization in Dynamic Environments", Department of Mathematics, University of Patras, Greece, pp. 591-593.
- [3] Parsopoulos, K.E., Vrahatis, M.N., "Parameter Selection and Adaptation in Unified Particle Swarm Optimization", *Mathematical and Computer Modelling*, 46 (1-2), Elsevier, 2007, pp. 198-213.
- [4] C. Grosan, "Multiple Solutions for a System of Nonlinear Equations", Department of Computer Science School Of Technology, *International Journal Of Innovate Computing, Information and Control*, Babes-Bolyai University, Romania, pp. 1-3, 2008.
- [5] T. Krick, "Polinomios y Raíces", en *Polinomios en $C[X]$* , Departamento de Matemática, Universidad de Buenos Aires, Argentina, pp. 11-12, 1997.
- [6] B. Sturmfelds, "Solving Systems of Polynomial Equations", Department of Mathematics, University of California at Berkeley, USA, pp. 2-3,7-8, 2002.
- [7] L. A. Gómez Ardila, "Demostración del teorema sobre la relación entre sistemas de ecuaciones en números complejos y el problema de optimización", UIS, Noviembre 2010.
- [8] C. Castillo, "Funciones de Variable Compleja con Aplicaciones a la Teoría de Números", en *El Teorema de Cauchy*, pp. 401-409, 2010.
- [9] F. Sabaté, "El Teorema Fundamental del Algebra", Universidad de Barcelona, España, pp. 1-11, Marzo 2005.
- [10] C. D'Andrea, "Calculo de Raíces Reales de Polinomios", Departamento de Algebra y Geometría, Universidad de Barcelona, pp.7-13, 2006.
- [11] W.Zulehner, "A Simple Homotopy Method For Determining All Isolated Solutions To Polynomial Systems", *Mathematics of Computation*, Volume 50, Number 181, pp. 167-177, January 1988.

- [12] J. Kennedy y R. Eberhart, "Particle swarm optimization", Presentado *al Proc. Of the IEEE Int. Conf. on Neural Networks IV. Piscataway, NJ*, 1995.
- [13] X. Cui, T. Potok y P. Palathingal, "Document Clustering using Particle Swarm Optimization", *IEEE Applied Software Engineering Research Group Computational Sciences and Engineering Division Oak Ridge National Laboratory*, 2005.
- [14] K. Parsopoulos y M. Vrahatis, "Particle Swarm Optimization and Intelligence", en *Particle Swarm Optimization*, IGI Global, Hershey, USA, pp. 26-39, 89-99, 2010.
- [15] W.Han, P. Yang, "Coparison Study of Severals Kinds of Inertia Weights for PSO", *IEEE Special Scientific Research Project of Science and Technology Commission of Shanghai Municipality*, Shanghai, China, 2010.
- [16] J. Pérez, "Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo", Tesis de Doctorado, Universidad de Cantabria, Santander, España, pp. 147-148, Octubre de 2005.
- [17] Cagnina L., "Optimización mono y multiobjetivo a través de una heurística de inteligencia colectiva", en *La heurística Particle Swarm Optimization*, Tesis de Doctorado, Doctorado en ciencias de la computación, Universidad Nacional de San Luis, Argentina, pp. 27-42, 2010.
- [18] Richard L. Burden y J. Douglas Faires, "Análisis numérico", en *Soluciones numéricas de sistemas de ecuaciones no lineales*, Thomson, Ed. 7ma ed., pp. 600-634, 2007.
- [19] C. Sáez, F. Rodríguez, "Resolución Numérica de Sistemas de Ecuaciones no Lineales", Universidad Pontificia Comillas, Capitulo 5, Madrid, España 2005.