

C-PSO: Optimización por cúmulo de partículas incrustando la ley de Coulomb

Ricardo Solano Monje¹, Nayeli Joaquinita Meléndez Acosta²,
Cosijopii García García², Homero Vladimir Ríos Figueroa³

¹ Instituto Tecnológico Superior de Ciudad Serdán, Cd. Serdán, Puebla, México

² Universidad del Istmo campus Ixtepec, Oaxaca, México

³ Universidad Veracruzana, Xalapa, Veracruz, México

rsolano@tecserdan.edu.mx,

{nayelimelendez,cosijopiigarci}@gmail.com, hrios@uv.com.mx

Resumen. Este artículo presenta una comparación del algoritmo PSO (Particle Swarm Optimization) contra una propuesta original a la cual hemos llamado C-PSO, esta versión hace uso de la Ley de Coulomb inspirados en el hecho que las aves tienen moléculas que les permiten detectar el campo magnético de la tierra y guiándolos hacia su destino. Por tanto cuando las aves migratorias realizan su recorrido utilizan su brújula interna. Saber esto nos llevó a pensar que quizás el algoritmo está incompleto y que era necesario agregar otra característica que guiara la dirección de la partícula mejorando el rendimiento del algoritmo PSO, así que recreamos la Optimización por enjambre de partículas incrustando la Ley de Coulomb. En la pruebas se ha incluido también al Algoritmo de Búsqueda Gravitacional, debido a su semejanza con nuestro algoritmo. Los algoritmos analizados y el propuesto fueron probados en 10 funciones de optimización con 1000 iteraciones para 10, 30, 50 y 100 dimensiones. Los resultados obtenidos muestran que el algoritmo propuesto C-PSO presenta mejores resultados en comparación con PSO y en algunos casos supera GSA.

Palabras clave: cúmulo de partículas, optimización por cúmulo de partículas PSO, ley de Coulomb, algoritmo de búsqueda gravitacional GSA.

C -PSO: Particle Swarm Optimization by Embedding Coulomb's Law

Abstract. This article presents a comparison of the classical PSO algorithm (Particle Swarm Optimization) and a novel proposal we have called C-PSO, this new approach makes use of the Coulomb Law inspired by the fact that birds have molecules in their eyes allowing them to detect the magnetic field of the earth then guiding them towards their destination. Therefore when the migratory birds make their journey they use their internal compass. Knowing this led us to think that perhaps the algorithm was incomplete and that it was

necessary to add another feature that would guide the direction of the particle then improving the performance of the PSO algorithm, so we recrafted Particle Swarm Optimization by embedding the Coulomb Law (Coulomb Particle Swarm Optimization C-PSO). The Gravitational Search Algorithm (GSA) has also been included in the tests due to its similarity to our algorithm. The algorithms were tested on 10 optimization functions with 1000 iterations for 10, 30, 50 and 100 dimensions. The results obtained show that the proposed algorithm C-PSO gives better results compared to PSO and in some cases surpasses GSA.

Keywords: particle swarm, particle swarm optimization (PSO), Coulomb's law, gravitational search algorithm (GSA).

1. Introducción

La optimización es el proceso usado para encontrar la mejor solución a un problema determinado. Existen dos tipos de técnicas de optimización las estocásticas y las determinísticas. Los métodos determinísticos conducen a enormes esfuerzos computacionales en contra parte con los algoritmos estocásticos, quienes presentan eficiencia computacional.

Los algoritmos inspirados en la naturaleza son métodos heurísticos, no determinísticos, que simulan el comportamiento de sistemas naturales y tienen la capacidad de resolver problemas complejos a partir de condiciones y reglas iniciales, con algún o ningún conocimiento del espacio de búsqueda [1].

Tres son los elementos más importantes en el diseño de un algoritmo bioinspirado, elegir la representación apropiada del problema, la función de aptitud que medirá la calidad de las soluciones y la definición de operadores para generar nuevas soluciones.

La optimización usando algoritmos bioinspirados tiene dos categorías principales: algoritmos basados en cúmulos de partículas y algoritmos evolutivos. El algoritmo basado en la optimización por cúmulo de partículas (PSO, Particle Swarm Optimization) emula el comportamiento colectivo de los animales y los algoritmos evolutivos emulan la evolución natural [2].

Exploración es la capacidad de un algoritmo de búsqueda, para explorar diferentes regiones del espacio de búsqueda con el fin de localizar una solución óptima. La explotación, por otra parte, es la capacidad de concentrar la búsqueda en un área prometedora para refinar una solución candidata [3].

En los problemas de optimización se buscan técnicas heurísticas y meta heurísticas que proporcionen buenos resultados en tiempo de cómputo razonables.

Cuando las aves migran viajan largas distancias, suelen viajar al mismo lugar año tras año, es decir son muy precisos en su navegación. Al principio se pensaba que se guiaban por el sol, pero entonces no podrían navegar en días nublados y sobre el océano, así que descubrieron que las aves pueden ver el campo magnético y orientarse en consecuencia [4,5], pero no solo en las aves, también los reptiles como las tortugas [6] y animales marinos como focas, leones marinos y elefantes marinos [7].

Saber esto nos llevó a buscar y pensar que quizás el algoritmo PSO estaba incompleto y que era necesaria agregar otra característica que guiara la dirección de la partícula, de esta forma es que creamos C-PSO, el algoritmo de Optimización por Cúmulo de Partículas usando la Ley de Coulomb.

Este artículo presenta una comparación del algoritmo PSO y una nueva versión llamado C-PSO, los cuales permiten resolver problemas numéricos de optimización. El objetivo del artículo es probar el desempeño del algoritmo bio-inspirado llamado C-PSO utilizando un conjunto de funciones para su optimización. Adicionalmente en las pruebas se agregó el algoritmo GSA. El contenido del artículo se encuentra organizado de la siguiente manera, primero en la segunda sección se describe el PSO básico, luego en la tercera sección se describe la nueva versión C-PSO, posteriormente en la cuarta sección muestran las pruebas y resultados realizados. Finalmente, en la quinta sección se presentan las conclusiones.

2. Algoritmo optimización por cúmulo de partículas (PSO)

El algoritmo basado en la optimización por cúmulo de partículas fue inventado por el Dr. Russell C. Eberhart y Dr. James Kennedy en 1995. Este algoritmo emula el comportamiento de las sociedades de animales que no tienen ningún líder en su cúmulo, como en el caso de las bandadas de aves y los bancos de peces [3].

Los miembros del cúmulo se comunican para informarse quien tiene la mejor solución y todas las partículas se mueven hacia ese lugar. Esto sucede siempre que se encuentre una mejor fuente de alimento (o solución).

El algoritmo consiste en un cúmulo de partículas que se colocan en el espacio de búsqueda de algún problema o función, donde cada partícula representa una solución potencial al problema. La solución optimizada para cada partícula es determinada por la función de aptitud. Cada partícula determina su movimiento a través de la combinación de dos capacidades esenciales: la memoria de su mejor posición (local best, *lbest*) y el conocimiento de la mejor posición encontrada por todas las partículas, es decir lo mejor del cúmulo (global best, *gbest*). La siguiente iteración tiene lugar después de que todas las partículas se han movido. El cúmulo colectivamente busca la mejor solución, así como las aves buscan su comida

Cada particular en el cúmulo está compuesta por tres vectores: la posición actual x_i , la mejor posición anterior *lbest* y la velocidad v_i . La posición es un conjunto de coordenadas que describen un punto en el espacio. En cada iteración del algoritmo, la posición actual se evalúa como una solución de problema. Si esa posición es mejor que la que se ha encontrado hasta ese momento, entonces las coordenadas se almacenan en el segundo vector *lbest*. El valor de la mejor solución encontrada por el cúmulo hasta ese momento se almacena en *gbest*, este también se actualiza si la solución es mejor. El objetivo del algoritmo es seguir encontrando mejores soluciones y actualizar *lbest* y *gbest*.

La velocidad de la partícula se modifica en cada iteración de la ecuación (1) utilizando la velocidad anterior multiplicada por una constante w , un componente cognitivo y un componente social. El componente cognitivo representa la distancia entre la posición actual y la mejor posición conocida por esa partícula (*lbest*) y el

componente social representa la distancia entre la posición actual y la mejor posición del cúmulo (*gbest*) [8].

$$v_{i+1} = w \cdot v_i + c_l \cdot r_l \cdot (lBest_i - x_i) + c_g \cdot r_g \cdot (gBest_d - x_i). \quad (1)$$

Una partícula se mueve desde una posición del espacio de búsqueda hasta otra ecuación (2), añadiendo al vector posición x_i el vector velocidad v_i para obtener un nuevo vector posición de la siguiente manera:

$$x_i = x_{i-1} + v_i, \quad (2)$$

donde x_i es la posición de la partícula i , v_i es la velocidad de la partícula i .

2.1. Algoritmo de búsqueda gravitacional (GSA)

Rashedi en 2009 propuso el Algoritmo de Búsqueda Gravitacional (Gravitational Search Algorithm, GSA), está basado en la ley de gravitación universal que considera que dos objetos se atraen entre sí por la fuerza de la gravedad y esta fuerza provoca un movimiento global en todos los objetos, las masa más pesadas que corresponden a las mejores soluciones se mueven más lentamente que las masas ligeras para garantizar la exploración del algoritmo [9].

3. Optimización por cúmulo de partículas usando la ley de Coulomb (C-PSO)

C-PSO es una nueva versión de PSO que hace uso de la Ley de Coulomb, el uso de esta ley es meramente una decisión de cierta forma arbitraria, pero no del todo, ya que la idea surgió del objetivo de guiar la dirección que toma cada partícula, pensando en el hecho de saber que “los pájaros pueden ver el campo magnético de la tierra”. Los ojos de las aves tienen moléculas que les permiten detectar el campo magnético de la tierra. Por tanto, cuando las aves migratorias realizan su recorrido utilizan su brújula interna [5].

Esto nos llevó a pensar que quizás el algoritmo está incompleto y que era necesaria una característica que guiara la dirección de la partícula mejorando el algoritmo de PSO, de esta forma es que creamos el algoritmo de Optimización por Cúmulo de Partículas usando la Ley de Coulomb C-PSO (Coulomb Particle Swarm Optimizaton).

3.1. Ley de Coulomb

En 1784, el físico francés Charles Augustín Coulomb, descubrió la ley cuantitativa de las fuerzas entre dos cargas puntuales, ecuación (3). Cargas puntuales son aquellas cuyas dimensiones geométricas son despreciables comparadas con las distancias de separación entre ellas [10].

Suponga que dos cargas puntuales, m_1 y m_2 , están separadas a una distancia d en el vacío. Si m_1 y m_2 tienen el mismo signo, las dos cargas se repelen mutuamente; si poseen signos opuestos, se atraen una a la otra. La fuerza que experimenta una carga

debido a la otra se conoce como fuerza de Coulomb o eléctrica y está dada por la Ley de Coulomb:

$$F = k \frac{m_1 m_2}{d^2}, \quad (3)$$

donde:

- F es la fuerza magnética,
- m_1 y m_2 son las masas magnéticas de cada partícula, en este caso se utilizan los costos de las partículas,
- k es la constante de permisividad eléctrica, la cual fue remplazada por la permeabilidad magnética absoluta μ la cual se define como:
 $\mu = \mu_r \mu_0$ Donde u_r y u_0 son constantes $\mu = 5000 \cdot 4\pi \times 10^{-7}$ quedando la fórmula final como $F = \mu \frac{m_1 m_2}{d^2}$.

Hemos usado la Ley de Coulomb porque nos hemos imaginado que las partículas son dos cargas puntuales. Ahora para calcular la distancia entre las dos cargas ecuación (4) se hace uso de la distancia entre dos puntos:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (4)$$

Utilizando la fórmula anterior, pero con dos valores:

$$d = \sqrt{(lBestPos_1 - gBestPos_1)^2 + (lBestPos_2 - gBestPos_2)^2}. \quad (5)$$

La ecuación (6) para n par de valores:

$$D = \sqrt{\sum_{i=1}^n (lBestPos_i - gBestPos_i)^2}. \quad (6)$$

Agregando esta característica a la fórmula del PSO original queda como se observa en la ecuación (7) de la siguiente manera:

$$v_{i+1} = w \cdot v_i + c_p \cdot r_p \cdot (pBest_i - x_i) + c_g \cdot r_g \cdot (gBest_d - x_i), \quad (7)$$

donde w ecuación (8) se define de la siguiente manera:

$$w = \begin{cases} \mu \frac{gBestCost * lBestCost}{d^2}, & d \leq 0.5 \\ 0, & d > 0.5 \end{cases}, \quad (2)$$

donde d es la distancia entre la posición del $lBest$ de cada partícula y el $gBest$ utilizando la ecuación (6), donde $gBestCost$ es el mejor costo global y $lBestCost$ es el mejor costo local.

La posición de la partícula es influenciada por la velocidad. Esta posición cambia añadiendo una velocidad, a la posición actual [1]. PSO es más sencillo que otros algoritmos de optimización, puesto que sólo existe un operador de mutación, la velocidad de la partícula.

4. Pruebas y resultados

Para probar el rendimiento de la nueva versión de algoritmo C-PSO implementado en este trabajo se han seleccionado 10 funciones de optimización ecuaciones (9-18), las cuales fueron seleccionadas de [11]. A continuación, se muestran las funciones utilizadas:

- Función Ackley

$$f(x) = 20 \exp \left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(-0.2 \frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i) \right) + 20 + \exp(1), \quad (3)$$

$$-32 \leq x_i \leq 32, \\ \min(f(x)) = f(0 \dots 0) = 0.$$

- Función Sphere

$$f(x) = \sum_{i=1}^d x_i^2, \quad (4)$$

$$-100 \leq x_i \leq 100, \\ \min(f(x)) = f(0 \dots 0) = 0.$$

- Función Step

$$f(x) = \sum_{i=1}^d (|x_i + 0.5|)^2, \quad (5)$$

$$-100 \leq x_i \leq 100, \\ \min(f(x)) = f(0 \dots 0) = 0.$$

- Función Rastrigin

$$f(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad (6)$$

$$-5.12 \leq x_i \leq 5.12, \\ \min(f(x)) = f(0 \dots 0) = 0.$$

- Función Schwefel 2.22

$$f(x) = \sum_{i=1}^d |x_i| + \prod_{i=1}^d |x_i|, \quad (7)$$

$$-10 \leq x_i \leq 10, \\ \min(f(x)) = f(0 \dots 0) = 0.$$

- Función Rosembrock

$$f(x) = \sum_{i=1}^{d-1} \left| 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right|, \quad (8)$$

$$-30 \leq x_i \leq 30, \\ \min(f(x)) = f(1 \dots 1) = 0.$$

- Función Griewank

$$f(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (9)$$

$$-600 \leq x_i \leq 600,$$

$$\min(f(x)) = f(0 \dots 0) = 0.$$

- Función Quartic

$$f(x) = \sum_{i=1}^d ix_i^4, \quad (10)$$

$$-1.28 \leq x_i \leq 1.28,$$

$$\min(f(x)) = f(0 \dots 0) = 0.$$

- Función Salomon

$$f(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^d x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^d x_i^2}, \quad (11)$$

$$-100 \leq x_i \leq 100,$$

$$\min(f(x)) = f(0 \dots 0) = 0.$$

- Función general penalizada

$$f(x) = 0.1 \times \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{d-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_d - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^d u(x_i, a, k, m), \quad (12)$$

donde

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

$$a = 5, k = 100, m = 4,$$

$$-50 \leq x_i \leq 50,$$

$$\min(f(x)) = f(1 \dots 1) = 0.$$

En las pruebas se han ejecutado tres algoritmos PSO, GSA y la nueva versión propuesta C-PSO. Estas pruebas han sido diseñadas bajo las competencias del CEC [12] con el objetivo de comparar el rendimiento de los algoritmos bajo un conjunto de prueba más grande y complejo. Las configuraciones de los parámetros para realizar las pruebas se muestran en la tabla 1.

Los algoritmos fueron probados con 10 funciones de optimización corriendo cada algoritmo 50 veces para 10, 30, 50 y 100 dimensiones (véase tablas 2-5). En las tablas se presenta el mejor, el peor, el promedio, la mediana y la desviación estándar. Los resultados muestran que a C-PSO le fue mucho mejor que a PSO, también nos dimos cuenta que C-PSO y GSA son casi iguales, siendo mejor C-PSO en algunas funciones.

Una ventaja del algoritmo C-PSO es que al añadir la Ley de Coulomb no se estanca mucho en los mínimos locales, además también es más rápido en dimensiones grandes comparado con su versión normal PSO.

Tabla 1. Parámetros de configuración.

Método	Parámetro	Valor
PSO C-PSO	Condición de paro	No. Máximo de Iteraciones Error
		1000 $r < 10e - 8$
		Coefficiente de decisión personal
		1.8
		Coefficiente de decisión global
		1.8
		Coefficiente de inercia (Solo PSO)
	1	
GSA	Población: número de partículas	100
	Espacio de búsqueda	Indicado en la formula
	Condición de paro	No. Máximo de Iteraciones Error
		1000 $r < 10e - 8$
		Constante gravitacional inicial
		100
		Población
	100	
	Espacio de búsqueda	Indicado en la formula

Una limitación de nuestro algoritmo es que en espacios de búsqueda muy grandes el error disminuye cada vez más lento conforme se va acercando al objetivo, esto se compensa con el hecho que hay muy poco estancamiento en mínimos locales.

Tabla 2. 10 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F1	PSO	4.6434	4.1835	4.2823	2.8901	3.3688e-01
	GSA	9.1844e-10	7.1027e-10	7.0143e-10	5.2529e-10	9.0299e-11
	C-PSO	1.3973	2.7947e-02	8.1978e-09	4.7481e-09	1.9761e-01
F2	PSO	7.2494e+01	4.9548e+01	4.8874e+01	2.5178e+01	1.1167e+01
	GSA	4.1972e-09	2.5243e-09	2.5670e-09	8.6872e-10	6.9139e-10
	C-PSO	9.9438e-09	6.0263e-09	5.9261e-09	1.3679e-09	2.0151e-09
F3	PSO	96	6.4440e+01	6.5500e+01	20	1.8031e+01
	GSA	0	0	0	0	0
	C-PSO	0	0	0	0	0
F4	PSO	2.5666e+01	1.8389e+01	1.8552e+01	1.2017e+01	2.7875
	GSA	2.9849	5.5718e-01	0	0	8.5660e-01
	C-PSO	6.6891e-01	1.7328e-01	1.2931e-01	0	1.9470e-01
F5	PSO	2.5393	1.9580	1.9622	1.4372	2.7298e-01
	GSA	9.9871e-09	8.7683e-09	8.9258e-09	6.3721e-09	9.8703e-10
	C-PSO	9.8754e-09	8.4571e-09	8.5493e-09	5.7628e-09	1.0619e-09
F6	PSO	2.7844e+05	6.9823e+04	5.7794e+04	8.2261e+03	4.7123e+04
	GSA	1.9798e+02	1.8520e+01	5.4451	5.1949	4.1633e+01
	C-PSO	6.0333e+02	3.3051e+01	9	7.4515	9.3606e+01
F7	PSO	8.1878e-01	6.3663e-01	6.5501e-01	4.7698e-01	8.6228e-02
	GSA	9.8573e-03	9.3676e-04	8.2554e-09	5.3021e-09	2.5825e-03
	C-PSO	1.3527e-01	5.1029e-02	5.1650e-02	0	3.3510e-02
F8	PSO	2.7773e-04	1.2729e-04	1.1703e-04	1.6621e-05	5.7889e-05

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F9	GSA	9.9541e-09	7.0221e-09	7.5196e-09	6.4431e-10	2.4099e-09
	C-PSO	9.9528e-09	5.0270e-09	5.2708e-09	6.9464e-10	2.8921e-09
	PSO	1.3137	1.0519	1.0957	7.6368e-01	1.3454e-01
	GSA	1.9994e-01	1.0809e-01	1.0050e-01	9.9873e-02	2.3646e-02
	C-PSO	1.9987e-01	9.9901e-02	9.9873e-02	0	6.6964e-02
	PSO	3.4320	2.2167	2.0974	1.0840	6.0645e-01
F10	GSA	9.8203e-09	7.5572e-09	8.3043e-09	2.6701e-09	1.6825e-09
	C-PSO	1.0988e-02	4.4254e-04	6.8807e-09	1.8090e-09	2.1745e-03

Tabla 3. 30 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F1	PSO	9.0590	8.5251	8.5657	7.7965	0.3014
	GSA	9.9785e-09	9.2255e-09	9.1271e-09	8.2675e-09	5.1678e-10
	C-PSO	2.4895	0.4284	9.3258e-09	6.3903e-09	0.8165
F2	PSO	1.6459e+03	1.2974e+03	1.3051e+03	915.8696	147.8678
	GSA	9.9943e-09	8.8652e-09	9.0403e-09	5.8492e-09	9.0711e-10
F3	C-PSO	5.6686	0.2138	7.6648e-09	9.8626e-11	1.0602
	PSO	1592	1.3031e+03	1.3315e+03	758	170.8434
	GSA	0	0	0	0	0
F4	C-PSO	1	0.0200	0	0	0.1414
	PSO	221.8564	179.4923	180.1052	152.2287	13.5775
	GSA	11.9395	7.1438	6.9647	2.9849	2.1148
F5	C-PSO	47.6264	29.0552	29.0450	0	10.4963
	PSO	17.5114	15.2100	15.4030	12.0971	1.1574
	GSA	1.2879e-08	1.0228e-08	9.8634e-09	8.3128e-09	1.0127e-09
F6	C-PSO	3.3713	1.7896	1.9736	0	0.9532
	PSO	2.0607e+07	1.2343e+07	1.2673e+07	5.1112e+06	3.3507e+06
	GSA	354.2852	52.6966	26.0432	25.7199	76.3510
F7	C-PSO	492.2854	130.6455	100.2836	29	111.4637
	PSO	1.3891	1.3181	1.3258	1.1634	0.0494
	GSA	0.0221	0.0015	9.0379e-09	5.2854e-09	0.0047
F8	C-PSO	0.0441	0.0085	0.0037	0	0.0107
	PSO	0.0730	0.0522	0.0520	0.0207	0.0121
	GSA	9.9521e-09	8.0200e-09	8.2608e-09	4.8328e-09	1.3746e-09
F9	C-PSO	9.8256e-09	6.9951e-09	7.1661e-09	1.6977e-09	1.8737e-09
	PSO	4.4008	4.0113	4.0323	3.5050	0.1960
	GSA	0.7999	0.4931	0.4999	0.3410	0.1026
F10	C-PSO	0.5999	0.3379	0.3999	0	0.1307
	PSO	9.8178e+03	2.8002e+03	2.2841e+03	59.9781	2.4078e+03
	GSA	0.0108	4.2203e-04	8.7322e-09	5.3589e-09	0.0019
	C-PSO	0.0974	0.0054	8.0324e-09	5.0897e-13	0.0154

Tabla 4. 50 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F1	PSO	10.6956	10.1705	10.2193	9.2502	0.2617

	GSA	9.9817e-09	9.5161e-09	9.6041e-09	8.7679e-09	3.1137e-10
	C-PSO	3.2808	0.5174	9.8090e-09	5.8261e-09	1.0096
	PSO	4.2407e+03	3.5573e+03	3.6241e+03	2.5155e+03	419.4516
F2	GSA	9.9635e-09	8.9518e-09	9.0464e-09	7.0667e-09	7.3516e-10
	C-PSO	2.3243	0.0465	5.3584e-09	1.0093e-11	0.3287
	PSO	4310	3.7190e+03	3764	2950	316.6848
F3	GSA	0	0	0	0	0
	C-PSO	1	0.1000	0	0	0.3030
	PSO	449.5038	372.1244	373.2912	330.7323	19.9571
F4	GSA	29.8488	15.6408	15.4219	6.9647	4.2501
	C-PSO	89.4245	51.6254	50.5354	20.9171	14.3382
	PSO	37.7666	34.4328	34.4231	29.8963	1.7293
F5	GSA	2.7137e-08	2.0071e-08	2.0071e-08	1.5849e-08	2.0431e-09
	C-PSO	11.3297	3.0373	2.3185	1.0268	2.0830
	PSO	1.0428e+08	7.3024e+07	7.2840e+07	4.6470e+07	1.2407e+07
F6	GSA	381.8985	75.7824	46.3782	44.8286	72.9732
	C-PSO	828.0664	219.5308	183.5675	49	164.3921
	PSO	2.0316	1.9181	1.9304	1.7586	0.0753
F7	GSA	0.0148	0.0014	9.2736e-09	7.1085e-09	0.0036
	C-PSO	0.1331	0.0135	8.7100e-09	0	0.0242
	PSO	0.7144	0.4612	0.4527	0.2953	0.0879
F8	GSA	9.9410e-09	8.2150e-09	8.5915e-09	2.8050e-09	1.4767e-09
	C-PSO	9.9674e-09	7.8372e-09	8.5355e-09	1.6058e-09	2.0493e-09
	PSO	7.2225	6.6977	6.6966	6.2010	0.2421
F9	GSA	2.3044	1.7802	1.7103	1.0027	0.2954
	C-PSO	0.7999	0.5400	0.5999	0	0.1979
	PSO	4.4740e+05	1.4408e+05	1.4149e+05	2.5241e+04	6.5484e+04
F10	GSA	0.4084	0.0103	9.2555e-09	7.1621e-09	0.0576
	C-PSO	0.0440	0.0042	5.7271e-09	2.0569e-11	0.0077

Tabla 5. 100 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F1	PSO	1.2824e+01	1.2225e+01	1.2197e+01	1.1767e+01	2.5636e-01
	GSA	2.7213e-09	2.3306e-09	2.3404e-09	1.8929e-09	1.7055e-10
	C-PSO	2.5943	1.2989	1.5372	7.1054e-15	8.5017e-01
F2	PSO	1.3481e+04	1.1526e+04	1.1532e+04	1.0151e+04	7.3413e+02
	GSA	2.4663e-07	1.7871e-07	1.7669e-07	1.3286e-07	2.6073e-08
	C-PSO	1.1100e-02	2.2200e-04	3.7623e-09	6.3007e-12	1.5698e-03
F3	PSO	15212	1.3467e+04	1.3621e+04	11259	8.8689e+02
	GSA	94	3.0100e+01	2.6500e+01	7	1.6134e+01
	C-PSO	6	1.2400	1	0	1.3486
F4	PSO	9.5977e+02	8.4649e+02	8.4605e+02	7.7342e+02	3.3430e+01
	GSA	3.3829e+01	2.5949e+01	2.5869e+01	1.3929e+01	4.5254
	C-PSO	2.2528e+02	1.0938e+02	1.0688e+02	5.2293e+01	3.0030e+01
F5	PSO	1.1095e+02	9.6681e+01	9.6231e+01	8.3235e+01	5.3801e+00
	GSA	5.1731e-01	4.3299e-02	2.2635e-03	4.7920e-08	9.1704e-02
	C-PSO	2.7166e+01	5.7383	3.9847	2.0030	5.1023
F6	PSO	8.2480e+08	5.3770e+08	5.3109e+08	3.3714e+08	9.7281e+07
	GSA	1.8937e+05	4.6546e+04	3.8344e+04	9.4960e+03	3.4632e+04

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F7	C-PSO	1.5969e+03	5.2881e+02	5.6819e+02	99	3.0394e+02
	PSO	4.9550	4.3444	4.3553	3.6990	2.4480e-01
	GSA	3.9781e-01	5.3710e-02	3.8489e-02	1.3178e-03	6.0581e-02
	C-PSO	2.1215e-01	3.1499e-02	4.9296e-03	0	5.1660e-02
F8	PSO	9.4849	6.8017	6.8324	4.9220	1.0094
	GSA	9.9931e-09	9.3163e-09	9.3224e-09	8.0422e-09	4.6729e-10
	C-PSO	9.9761e-09	5.1623e-09	5.5475e-09	1.5223e-11	3.4967e-09
F9	PSO	1.3726e+01	1.2354e+01	1.2341e+01	1.1360e+01	5.0474e-01
	GSA	5.7491	4.6803	4.7005	4.0999	3.9002e-01
	C-PSO	1.9999	1.5931	1.5999	9.9987e-01	1.8692e-01
F10	PSO	1.0215e+07	4.2791e+06	4.1347e+06	2.1372e+06	1.4783e+06
	GSA	4.1637e+01	2.2993e+01	2.2407e+01	1.0649e+01	6.6393
	C-PSO	5.3282	1.1469e-01	6.5795e-04	3.7046e-07	7.5250e-01

5. Conclusiones

PSO consiste en un enjambre de partículas, donde la partícula representa una solución potencial (mejor condición). La partícula se moverá a través de un espacio de búsqueda multidimensional para encontrar la mejor posición en ese espacio (la mejor posición posible para los valores máximo o mínimo).

En este artículo se presenta una comparación del algoritmo PSO y una nueva versión llamado C-PSO, en esta nueva versión hacemos uso de la ley de coulomb de ahí su nombre C- PSO. C-PSO ha sido inspirado en el hecho de que las aves pueden ver el campo magnético de la tierra y este guía su destino, así que decidimos agregar otra característica que guiará la dirección de la partícula mejorando el rendimiento del algoritmo original PSO y dando origen a C-PSO.

Los resultados de las pruebas muestran un rendimiento favorable de C-PSO comprado con PSO y GSA. En promedio el algoritmo C-PSO encuentra la solución en la iteración 100, mientras que los otros dos requieren de muchas más iteraciones. Nuestro algoritmo ha mostrado buenos resultados.

Referencias

1. Palupi-Rini, D., Mariyam-Shamsuddin, S., Sophiyati-Yuhaniz, S.: Particle Swarm Optimization: Technique, System and Challenges. *International Journal of Computer Applications* 14(1) (2011)
2. Chavan, S. D., Adgokar, N. P.: An Overview on Particle Swarm Optimization: Basic Concepts and Modified Variants. *International Journal of Science and Research (IJSR)* 4, pp. 255–260 (2015)
3. Bansal, J. C., Singh Mukesh Saraswat, P. K., Verma A., Singh Jadon, S., Abraham, A.: Inertia Weight Strategies in Particle Swarm Optimization. In: *Third World Congress on Nature and Biologically Inspired Computing*, pp. 640–647 (2011)
4. National Geographic News Homepage, <http://news.nationalgeographic.com/news/2007/09/070927-magnetic-birds.html> (2007)

5. Science X Homepage, <https://phys.org/news/2009-11-birds-earth-magnetic-field.html> (2009)
6. Science X Homepage, <https://phys.org/news/2015-01-sea-turtles-magnetic-home.html> (2015)
7. Science X Homepage, <https://phys.org/news/2014-12-antarctic-earth-magnetic-field.html> (2014)
8. Kumar, A, Kumar Singh, B., Patro, B. D. K.: Particle Swarm Optimization: A Study of Variants and Their Applications. *International Journal of Computer Applications* (0975 – 8887) 135(5) (2016)
9. Suarez, A. J., Rodríguez, J. E.: Algoritmos Meta heurísticos basados en la ley de gravitación universal para agrupamiento de datos. *Sistema, Cibernética e Informática* 13(1) (2016)
10. Barco Ríos, H., Rojas Calderón, E., Restrepo Parra, E.: *Física: principios de electricidad y magnetismo*. Primera Edición. Universidad Nacional de Colombia, Colombia (2012)
11. Viveros-Jiménez, F., Mezura-Montes, E., Gelbukh, A.: Empirical analysis of a micro-evolutionary algorithm for numerical optimization. *International Journal of Physical Sciences* 7(8), pp. 1235–1258 (2012)
12. Awad, N. H., Ali, M. Z., Suganthan, P. N., Liang, J. J., Qu, B. Y.: Problem Definitions and Evaluation Criterial for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report (2016)