

SIMULACIÓN 3D DE ALGORITMOS DE INTELIGENCIA DE ENJAMBRES

Cortés Hernández, José Miguel (1), García Capulín, Carlos Hugo (2)

1 [Licenciatura en Ingeniería en Sistemas Computacionales, Universidad de Guanajuato] | Dirección de correo electrónico: [jm.corteshernandez@ugto.mx]

2 [Departamento de Electrónica, División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato] | Dirección de correo electrónico: [carlosg@ugto.mx]

Resumen

La inteligencia de enjambres es una rama de la ciencia inspirada en el estudio del comportamiento de ciertos sistemas biológicos con el fin de desarrollar algoritmos computacionales capaces de mostrar un comportamiento inteligente. En general los sistemas de inteligencia de enjambre son formados por un conjunto de elementos simples, que siguen reglas sencillas de comportamiento, los cuales interactúan entre sí. Como resultado se obtiene que, en conjunto, el enjambre puede lograr comportamientos muy complejos, mucho mayores al comportamiento individual mostrado por cada uno de sus elementos. Entre los algoritmos de inteligencia de enjambres más populares se encuentran aquellos basados en el comportamiento de las aves, las abejas y las hormigas. Con el fin de lograr una mejor comprensión en el estudio de los algoritmos de inteligencia de enjambres en este trabajo se presenta una simulación tridimensional para facilitar la comprensión del funcionamiento de algunos de ellos, particularmente el algoritmo de optimización por enjambre de partículas. Con esta simulación se puede observar de manera gráfica el comportamiento del algoritmo y cómo afectan en su desempeño la variación de los parámetros ajustables del mismo.

Abstract

Swarm intelligence is a branch of science inspired by study the behavior of biological systems in order to develop computational algorithms capable of displaying intelligent behavior. Generally, swarm intelligence systems are formed by a set of simple elements that follow easy rules of behavior, which interact to each other. As a result, the swarm can achieve an intelligent global behavior, much greater than individual behavior shown by each of its elements. Among the most popular swarm intelligence algorithms we find those based on the behavior of birds, the bees and the ants. In order to achieve a better understanding in the study of intelligence algorithms in this work a three-dimensional simulation is presented, with the aim to clarify the functioning of some of them, particularly the particle swarm optimization algorithms. In this simulation, it can observe graphically the behavior of the algorithms and how affect their performance the variation of the adjustable parameters of the same.

Palabras Clave

Enjambre de Partículas; Optimización; Gráficos tridimensionales

INTRODUCCIÓN

La Inteligencia de Enjambres (IE), se considera una rama de la Inteligencia Artificial (IA), basada en el estudio del comportamiento de los sistemas biológicos, generando algoritmos para simular el comportamiento de estos, los cuales son formados por un conjunto de elementos simples que se rigen por un conjunto de reglas de comportamiento individual y grupal, logrando comportamientos muy complejos destacando el comportamiento grupal sobre el individual. Entre los algoritmos de IE más representativos se encuentran los siguientes: Optimización por Enjambre de Partículas (PSO por sus siglas en inglés) [1], Optimización por Colonia de Hormigas (ACO por sus siglas en inglés) [2] y Colonia de Abejas Artificiales (ABC por sus siglas en inglés) [3].

Particularmente, el algoritmo de PSO, es uno de los más utilizados por la comunidad científica, el cual ha mostrado mucho éxito en la solución de problemas de optimización [4, 5]. El algoritmo de PSO modela matemáticamente el comportamiento de las parvadas de aves, o los cardúmenes de peces. Donde cada uno de los miembros del grupo es denominado como una partícula, cada una de ellas esta definida por su posición, velocidad y la mejor posición en la que ha estado. El conjunto de partículas se denomina enjambre, el cual se mueve dentro de un espacio de búsqueda que está definido por un problema particular. Cada partícula del enjambre se desplaza en el espacio, en busca de mejores posiciones, que representan a su vez mejores soluciones al problema. La actualización de la posición se realiza en función del cambio de velocidad de cada partícula, el cual depende de una función en la que se involucran su inercia, el conocimiento individual de la partícula y la influencia social que ejerce todo el enjambre sobre cada partícula. Conforme el algoritmo avanza, el enjambre tiende a converger a una posición que representa la solución optima del problema particular. Con el desarrollado de investigaciones en el área, se han propuesto algunas mejoras al algoritmo de PSO original, entre las cuales podemos identificar el algoritmo de PSO con peso de inercia y el algoritmo PSO restringido. Básicamente estas variantes tratan de mejorar la convergencia del algoritmo, evitando el fenómeno de explosión del enjambre.

Con el fin de facilitar la comprensión del funcionamiento de este tipo de algoritmo, se presenta una simulación gráfica tridimensional, que nos permite observar el desempeño general del algoritmo en su ejecución [6], en la cual podemos analizar las trayectorias de las partículas, su convergencia, así como los efectos de la variación de los parámetros del PSO como son el tamaño del enjambre y los límites de velocidad de las partículas. Además, se puede analizar la diferencia en el desempeño entre las distintas variaciones del PSO y otros fenómenos que puedan ocurrir como, por ejemplo, la explosión del enjambre. Para realizar dicha simulación se desarrolló una librería en lenguaje C, que implementa un algoritmo genérico de PSO utilizando la versión original y las variantes de Peso de Inercia y de Restricción. De forma paralela se desarrolló un entorno gráfico tridimensional usando el estándar gráfico de OpenGL que nos permite llevar a cabo la simulación sobre distintas plataformas como lo son los sistemas operativos Windows, Linux y MacOS, esta es una cualidad muy importante del desarrollo, que es multiplataforma.

MATERIALES Y MÉTODOS

Algoritmo de optimización por enjambre de partículas

Para el algoritmo de PSO cada partícula X_{id} representa una posible solución a un problema en particular. El comportamiento del enjambre es modelado por la Ecuación 1, la cual contiene tres términos: el primero representa la inercia de la partícula v_{id} , esto es la tendencia de la partícula a seguir moviéndose en la misma dirección; el segundo representa el conocimiento individual de la partícula, es cual representa que tanto se aleja la partícula de la posición donde ha tenido su mejor resultado P_{id} y por último, el tercer término representa el

comportamiento social, el cual indica, que tanto la partícula es influenciada por el conocimiento del enjambre, particularmente el de la mejor partícula P_{gd} de todo el enjambre.

$$v_{id} = v_{id} + c_1\psi_1(P_{id} - X_{id}) + c_2\psi_2(P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + v_{id} \quad (2)$$

Además, nos encontramos con c_1 y c_2 que son valores constantes que regulan el comportamiento tanto individual como social de la partícula, ψ_1 y ψ_2 son valores aleatorios en $[0,1]$. El enjambre avanza dentro del espacio de soluciones del problema mediante un proceso iterativo, el cual actualiza la posición X_{id} de cada partícula del enjambre mediante la Ecuación 2, utilizando la velocidad v_{id} calculada con la Ecuación 1 en cada iteración. Conforme el algoritmo itera, el enjambre tiende a converger a la solución óptima del problema en particular.

El algoritmo de PSO original presenta un fenómeno llamado explosión del enjambre, es cual ocurre cuando todas partículas se alejan del espacio de búsqueda, esto debido a que la velocidad puede crecer sin control. Para prevenir esto se limita la velocidad, para que se mantenga dentro de un rango controlado. Otras soluciones fueron propuestas en las siguientes variantes del algoritmo: PSO con peso de inercia y PSO restringido. El primero de ellos agrega un peso de inercia en el calculo de la velocidad, de modo que la velocidad anterior es disminuida por el factor w de inercia como se puede observar en la Ecuación 3.

$$v_{id} = wv_{id} + c_1\psi_1(P_{id} - X_{id}) + c_2\psi_2(P_{gd} - X_{id}) \quad (3)$$

Por otro lado, el segundo agrega un factor χ de restricción el cual afecta tanto al término de inercia como a los términos de conocimiento individual y social de la partícula. El calculo de la velocidad para el PSO restringido esta dado por la Ecuación 4.

$$v_{id} = \chi\{v_{id} + c_1\psi_1(P_{id} - X_{id}) + c_2\psi_2(P_{gd} - X_{id})\} \quad (4)$$

Para poder ejecutar el PSO es necesario determinar cada uno de los parámetros antes mencionados, de los cuales depende el desempeño general del algoritmo. Para ayudar a entender este proceso y en general el comportamiento del PSO, en este trabajo se realizó una aplicación que nos permite simular de manera gráfica el funcionamiento del algoritmo, la cual se llevo a cabo de la siguiente manera.

La simulación grafica se realizó en tres partes, una interfaz interactiva con el usuario, la representación del espacio de búsqueda, y la representación del enjambre. Para la representación del enjambre: se utilizó una estructura que representa un punto en el espacio donde se ubica cada partícula, para representar el movimiento se uso un sistema de curvas Bézier, el sistema requiere de N puntos para formar la ruta que seguirá una partícula, en este caso se uso 5 puntos por partícula. Los puntos se obtienen de las posiciones de la partícula del PSO en ejecución. Para la representación del espacio de búsqueda se creo una estructura llamada malla, la cual crea una malla graficando la función objetivo dentro del rango de búsqueda de la solución. De este modo se puede crear una malla que represente el espacio en 2 dimensiones dándole la funcionalidad de movimiento para distintas vistas. La interfaz de interacción con el usuario permite la interacción con el usuario a través de botones que permiten modificar algunos parámetros del PSO, iniciar, pausar y reiniciar el algoritmo, así como seleccionar distintas funciones de prueba.

En la simulación se plantean distintas circunstancias como lo es, el cambio de número de partículas, mostrar el comportamiento de estas cuando se quita el limite de velocidad, así como cambiar entre las variaciones planteadas: PSO original, PSO con peso de inercia y PSO restringido. De este modo podemos ver el compartimiento de cada una de las variantes cambiando los parámetros de inicialización del PSO, así como poner pausa para poder ver donde estaban las partículas y poder mostrar la ruta que está siguiendo cada partícula del enjambre, entre otras funcionalidades.

RESULTADOS Y DISCUSIÓN

Para probar la simulación del PSO se utilizaron 7 funciones de prueba típicas en la literatura [4, 5]. Los resultados obtenidos por la simulación muestran el PSO original en la Figura 1, el PSO con peso de inercia en la Figura 2 y el PSO restringido en la Figura 3. Como se observa en la Figura 1 el PSO original converge hacia el mínimo global, aunque algunas de las partículas se mantienen un poco alejadas. En cambio, el PSO con peso de Inercia (Figura 2) converge casi la totalidad del enjambre al punto óptimo. También el PSO restringido presenta el efecto antes mencionado donde la mayoría de sus partículas convergen (Figura 3).

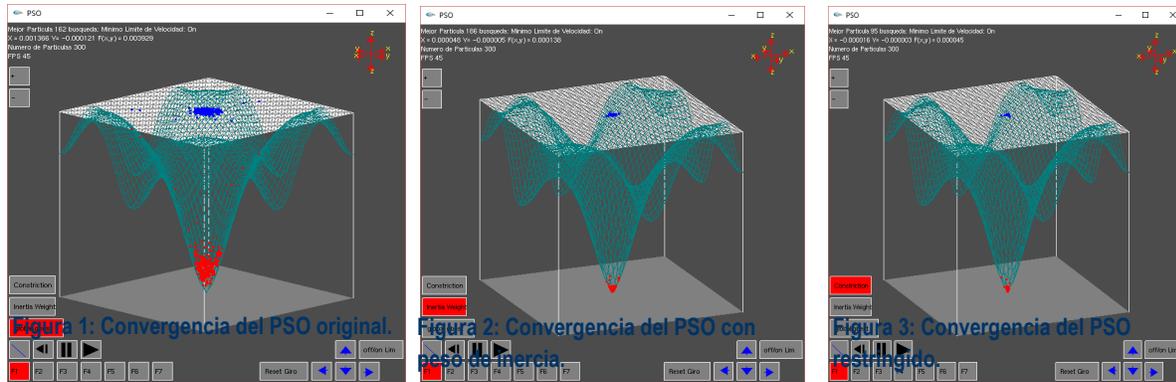


Tabla 1: Resultados obtenidos para los algoritmos de PSO probados.

Función	Rango	Valor óptimo	PSO original	PSO peso de inercia	PSO restringido
$f1 = -20 \exp \left\{ -0.2 \sqrt{\frac{1}{N} \sum_{i=1}^2 x_i^2} \right\} - \exp \left\{ \frac{1}{N} \sum_{i=1}^2 \cos(2\pi X_i) \right\} + 20 + e$	-1 a 1	0.0	0.000425	0.0000000	0.0000003
$f2 = -\sum_{i=1}^2 x_i \sin(\sqrt{x_i})$	-100 a 100	-127.269996	-127.269556	-127.296996	-127.269996
$f3 = \sum_{i=1}^5 j \cos((i+1)x_1 + i) \sum_{i=1}^5 j \cos((i+1)x_2 + i)$	-2 a 2	-186.730909	-186.657739	-186.730909	-186.730909
$f4 = \sum_{i=1}^2 (X_i^2 - 10 \cos(2\pi X_i) + 10)$	-1 a 1	41.002433	41.002228	41.002533	41.002533
$f5 = \sum_{i=1}^2 \left(\sum_{j=1}^i x_j \right)^2$	-500 a 500	0.0	0.142863	0.000008	0.00000
$f6 = \sum_{i=0}^1 \{(i+1)X_i^4\} + \text{rans}[0,1]$	-1.12 a 1.12	0.0	0.000352	0.000324	0.000077
$f7 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	-1.5 a 1.5	-1.031628	-1.031605	-1.031628	-1.031628

La configuración del PSO es la siguiente, se utilizaron 100 partículas para todas las variantes. Tanto par el PSO original y con peso de inercia c_1 y c_2 se asignaron a 1.8, en el PSO restringido fueron cambiados a 2.05 cada uno, para el Peso de Inercia se uso $w=0.6$ como peso y en PSO restringido se tomó $X=0.7$. Utilizando los parámetros antes mencionados cada una de las variaciones usadas en las pruebas fueron capaces de encontrar

el óptimo global como se puede observar en la Tabla 1, destacando el PSO con peso de Inercia. Dado que, simulando su operación, es el que llega más rápido y poco a poco se observa como va convergiendo. Lo mismo sucede con el PSO restringido, solo tarda un poco más en la convergencia, superando al PSO original.

CONCLUSIONES

De acuerdo con las simulaciones realizadas, las partículas del PSO original se mueven con suavidad hacia su objetivo dependiendo de los límites de velocidad, estas se pueden mover de manera más rápida, aunque pueden no llegar a converger. En cuanto a las versiones de PSO con peso de inercia y PSO restringido, presentan un comportamiento más limitado, siendo que, por su propia naturaleza, van limitando la velocidad a la que puede moverse una partícula con forme se acerca a su objetivo, lo cual les permite obtener una mejor convergencia del enjambre con una mayor precisión en la solución obtenida.

AGRADECIMIENTOS

Los autores agradecen el apoyo de la Universidad de Guanajuato a través de la Dirección de Apoyo a la Investigación y Posgrado (DAIP), así como el apoyo del programa para el Desarrollo Profesional Docente (PRODEP) a través del proyecto "Desarrollo de Técnicas para Mejorar el Desempeño Computacional de Algoritmos Bio-inspirados y su Aplicación en la Solución de Problemas de Optimización" con número 511-6/17-8074.

REFERENCIAS

1. Kennedy, J. & Eberhart, R. C. (1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, 4, pp. 1942-1948. doi: 10.1109/ICNN.1995.488968
2. Dorigo, M., Maniezzo, V. & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, 26 (1), pp 29-41. doi: 10.1109/3477.484436
3. Karaboga, D. (2005). An Idea Based On Honey Bee Swarm For Numerical Optimization, TECHNICAL REPORT-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
4. Shi, Y. & Eberhart, R. C. (1999). Empirical study of particle swarm optimization, Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, 3, pp. 1950. doi: 10.1109/CEC.1999.785511
5. Pant, M., Thangaraj, R. & Abraham, A. (2009) Particle Swarm Optimization: Performance Tuning and Empirical Analysis. In: Abraham A., Hassanien AE., Siarry P., Engelbrecht A. (Ed.) Foundations of Computational Intelligence Volume 3. Studies in Computational Intelligence, vol. 203. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-01085-9_5
6. Farooq, H. & Tariq, M. (2014) A Comparative Study on User Interfaces of Interactive Genetic Algorithm, Procedia Computer Science, 32, pp. 45-52. doi: 10.1016/j.procs.2014.05.396.