

COMPARACIÓN DE VARIANTES DE PSO EN OPTIMIZACIÓN GLOBAL CON RESTRICCIONES.

RESUMEN. El presente artículo realiza una comparación del rendimiento de las variantes básicas del algoritmo de Optimización Mediante Cúmulos de Partículas (PSO) para resolver problemas de optimización con restricciones. Se utilizará el valor de inercia y el factor de estrechamiento en cada variante. El problema que se aborda es la optimización global con restricciones. Al igual que otras heurísticas bio-inspiradas, en su concepción original PSO no cuenta un mecanismo para manejo de restricciones. En trabajos encontrados en la literatura especializada se proponen mecanismos para el manejo de restricciones en PSO; sin embargo, las variantes de PSO se utilizan sin realmente tener la certeza de que esa variante sea la que provee el mejor desempeño. En el presente trabajo se utilizan las versiones más populares de PSO de la literatura especializada: la versión global best y local best, con valor de inercia y también ambas versiones pero con factor de estrechamiento. Para el manejo de restricciones se utiliza un mecanismo simple encontrado en la literatura especializada basado en reglas de factibilidad. En la fase experimental se utilizan 13 funciones propuestas también en la literatura especializada para evaluar el desempeño de heurísticas para espacios restringidos. Para evaluar dicho rendimiento, se muestran las estadísticas (mejor valor, media y desviación estándar) de 30 ejecuciones independientes de cada variante en cada función de prueba. El documento se organiza como sigue: En la Sección 1 se presenta una introducción a PSO, la Sección 2 incluye la definición del problema de optimización. Más adelante, la Sección 3 enumera el trabajo relacionado. Nuestra propuesta es presentada en la Sección 4 y en la Sección 5 se discuten los resultados obtenidos, finalizando en la Sección 6 con las conclusiones y el trabajo futuro.

PALABRAS CLAVES: Inteligencia en cúmulos (swarm intelligence), optimización numérica, manejo de restricciones

1. INTRODUCCIÓN A PSO

La Optimización mediante Cúmulos de Partículas es una heurística desarrollada por Kennedy y Eberhart en 1995 [1], la cual ha sido utilizada para resolver problemas de optimización (numérica principalmente), denominada PSO por sus siglas en inglés

(Particle Swarm Optimization). PSO es una técnica fácil de implementar y de funcionamiento sencillo. Por su naturaleza representa un paradigma diferente a los algoritmos evolutivos, catalogándose dentro del área de Inteligencia en Cúmulos [2].

PSO es concebido como un algoritmo bio-inspirado por tener sus orígenes en la idea básica del comportamiento social de las bandadas de pájaros o bancos de peces. En estos grupos animales se establecen relaciones sociales entre los individuos del grupo, definiéndose jerarquías de acuerdo a las características de cada uno de ellos, existiendo un líder que es reconocido y seguido por los individuos de su grupo.

Inicio

Generar un cúmulo aleatorio inicial de tamaño = *parsize*

Evaluar cada partícula en el cúmulo

Para i = 1 hasta MaxGenerations

 Seleccionar el líder del cúmulo

 de acuerdo con su valor de calidad de cada partícula

 j=0

Mientras (j <= *parsize*)

 Aplicar formula de vuelo a la partícula "j"

 Evaluar la nueva posición de la partícula

 Actualizar el valor pbest (memoria) de

 la partícula "j" usando sus correspondientes valores de calidad

 j = j + 1

Fin de Mientras**Fin de Para****Fin**

Figura 1. Algoritmo básico de PSO

El líder es un individuo que tiene características físicas o habilidades que le permiten mantener un control sobre los demás en el grupo, y ser el guía para desarrollar actividades como buscar comida o moverse a otro lugar cuando la comida se ha acabado. Los individuos confían en la capacidad del líder para dirigirlos, pero en estos grupos sociales se suele cambiar de líder si surge un individuo con mejores capacidades que el viejo líder, sustituyéndolo en la guía del grupo. Cada individuo dentro del grupo ve influenciado su comportamiento por dos factores: El primero es su conocimiento y habilidades propias adquiridas durante su vida o de manera innata, y el segundo es la

influencia del líder. Este comportamiento es modelado para resolver problemas de optimización numérica.

En PSO se tiene un cúmulo de partículas donde cada una de ellas representa una posible solución al problema de optimización. Cada partícula tiene una memoria que le permite recordar la mejor posición por donde ha pasado. Además, cada una de ellas tiene asociada una velocidad que le sirve para definir hacia dónde se moverá en cada vuelo, donde el vuelo es la manera en que cada partícula se mueve. La actualización de la velocidad se calcula de acuerdo a 4 factores: la velocidad actual, la posición actual de la partícula, su valor en la memoria y la posición del líder del cúmulo. De este modelo se desprenden variantes que difieren en la forma de actualizar la velocidad de las partículas y en la forma de comunicación entre ellas. El algoritmo general de PSO se presenta en la Figura 1.

1.1 Variantes de acuerdo al cálculo de velocidad

Existen dos formas principales de calcular la velocidad en PSO. La primera es utilizando un valor de inercia, el cual determina la influencia del valor de la velocidad actual en el cálculo de la nueva velocidad. La segunda variante, en cambio, utiliza un valor conocido como factor de estrechamiento, el cual afecta a todos los factores involucrados en el cálculo de la velocidad y no tan solo al valor de la velocidad actual.

1.1.1 PSO con valor de inercia

Dada su naturaleza, PSO determina la nueva posición de cada partícula a partir una función de vuelo, actualizando su posición después de haber calculado la nueva velocidad. La velocidad representa en sí una dirección de búsqueda. En su concepción original, PSO no contaba con el valor de inercia [1], posteriormente se introdujo el valor de inercia [2] para reducir la influencia de la dirección de búsqueda que trae la partícula. La fórmula para calcular la velocidad de la partícula es la siguiente:

$$v(t+1) = w * v(t) + c_1 * \text{rand}() * (p\text{Best} - x) + c_2 * \text{Rand}() * (p_G - x)$$

La nueva velocidad ($v(t + 1)$) de la partícula es determinada en su forma general por los siguientes parámetros:

- Weight “w”. El peso de inercia o factor de inercia es un valor que regula la influencia de la velocidad anterior de la partícula $v(t)$ en el calculo de la nueva velocidad $v(t + 1)$, a manera de regular el vuelo de la partícula haciendo un

balance entre explotación y exploración del espacio de búsqueda. El factor de inercia promueve bajo ciertas condiciones la convergencia del cúmulo, es decir, que todas las partículas se acerquen al líder del cúmulo.

- $v(t)$. Representa la velocidad actual de la partícula que fue anteriormente calculada con la misma formula, se refiere a la dirección de vuelo que tiene la partícula. En un inicio todas las partículas tienen una velocidad de 0, lo que quiere decir toda partícula parte de un estado de reposo cuando inicia el proceso.
- Coeficientes de aceleración ($c_1 \cdot \text{rand}()$ y $c_2 \cdot \text{Rand}()$). $\text{Rand}()$ y $\text{rand}()$ representan un número real aleatorio con distribución uniforme entre 0.0 y 1.0. El coeficiente c_1 regula la influencia del conocimiento cognitivo de la partícula y el coeficiente c_2 regula la influencia del conocimiento social. c_1 regula la influencia que tiene la mejor posición alcanzada por la partícula (p_{Best}) para guiar su nueva dirección y c_2 regula la influencia del líder del cúmulo (p_G) en la dirección de búsqueda de la partícula.
- p_{Best} . Representa la memoria de la partícula, su mejor posición alcanzada hasta esa generación.
- p_G . Representa la mejor posición del cúmulo, es decir el líder.
- x . Representa la posición actual de la partícula que se toma como referencia para calcular la nueva velocidad (dirección de búsqueda).
- $v(t+1)$ Es la velocidad actualizada de la partícula que determinará la nueva dirección de búsqueda de la partícula en la generación $t + 1$. Una vez calculada la velocidad actualizada $v(t+1)$ se actualiza la posición de la partícula mediante la fórmula $x = x + v(t+1)$.

1.1.2 PSO con factor de estrechamiento

Esta variante de PSO fue propuesta por Parsopoulos y Vrahatis en [3]. No utiliza el valor de inercia, como en la versión básica de PSO, donde el factor de inercia (w) es utilizado para controlar la velocidad de la partícula y reducir la influencia de la dirección de búsqueda previamente recorrida por la partícula. La variante con el factor de estrechamiento representado por k ("constriction factor") no multiplica la velocidad actual de la partícula sino que afecta a la velocidad actualizada calculada, la fórmula de vuelo modificada es la siguiente:

$$v(t+1) = k[v(t) + c_1 \cdot \text{rand}() \cdot (p - x) + c_2 \cdot \text{Rand}() \cdot (p_G - x)]$$

Donde el factor de estrechamiento (k) es calculado mediante los factores de aceleración c_1 y c_2 . La variante con factor de estrechamiento bajo ciertos parámetros ofrece un mejor control de la velocidad y asegura la convergencia para problemas de optimización numérica sin restricciones [2].

1.2 Variantes de acuerdo con la comunicación entre partículas

Existen dos algoritmos básicos de PSO que se clasifican de acuerdo a la manera en que un individuo se relaciona socialmente con los demás individuos del cúmulo. El primer algoritmo se denomina global best (p_g) en donde solo existe un solo líder en el cúmulo y las partículas se relacionan socialmente con todos los individuos del cúmulo debido a la influencia de un líder global. El segundo algoritmo se conoce como local best ($lbest$), para esta variante el cúmulo completo se divide en un número n de vecindarios con igual número de individuos, y en cada vecindario existe un líder local el cual influencia el vuelo de las partículas dentro de su vecindario. Por ende, en la variante global best, cada partícula se comunica con todas las demás en el cúmulo, pues puede saber quién es la mejor de entre todas ellas. En contraste, en el modelo local best, las partículas sólo se comunican con sus vecinas e ignoran las posiciones de las demás. Véase la figura 2, donde en la parte (a) se muestra el modelo de comunicación para global best y en la parte (b) el modelo para local best, donde el vecindario es de 3 partículas.

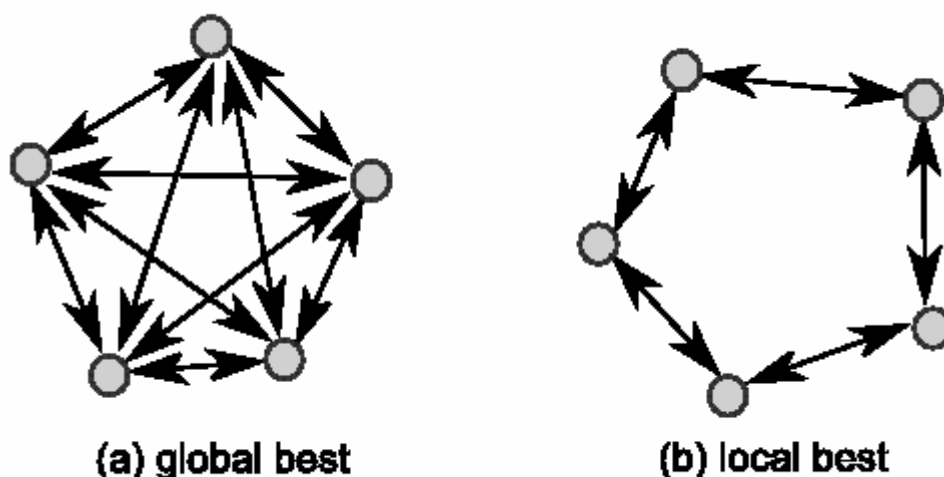


Figura 2. Modelos de comunicación básicos en PSO.

2. Optimización global

La optimización con restricciones en su forma más general es conocida como el problema general de programación no lineal que puede definirse como sigue:

Minimizar $f(\mathbf{x})$,

Sujeto a:

$g(\mathbf{x}) \leq 0; i = 1, 2, \dots, m$

$h(\mathbf{x}) \leq 0; i = 1, 2, \dots, l$

Donde \mathbf{x} es el vector de variables de decisión ($[x_1, x_2, \dots, x_d]$) las cuales tiene un rango inferior (L_i) y superior (U_i). La zona factible (F) representa el vector solución que cumple todas las restricciones, y se encuentra en el espacio de búsqueda (S) que pertenece a un espacio de continuo de n dimensiones dentro del rango de las variables.

3. Trabajo relacionado

Existen diferentes propuestas que utilizan PSO para resolver el problema general de programación no lineal (PNL), pero ninguna de ellas presenta una reflexión acerca de la variante de PSO utilizada, más bien, modifican una variante existente en algunas de sus mecanismos. Toscano y Coello [4] diseñan un operador de turbulencia y utilizan el valor de la función objetivo junto con las restricciones para determinar la calidad de una solución. Liang y Sugathan [5] utilizan un esquema de PSO multi cúmulos, cada subcúmulo se encarga de una restricción que es asignada dinámicamente. Parsopoulos y Vrahatis [6] utilizan la variante de PSO que utiliza un enfoque conjunto de global best y local best con factor de estrechamiento, utilizan parámetros que varían dinámicamente. Li, Tian y Kong [7] proponen un algoritmo de PSO que utiliza mutación como una alteración del vuelo de la partícula. Cagnina, Esquivel y Coello [8] estudian una variante de PSO que utiliza un enfoque conjunto de global best y local best con valor de inercia. Lu y Chen [9] proponen un algoritmo que optimiza dos objetivos, primero se buscan soluciones factibles para luego optimizar la función objetivo. Krohling y dos Santos Coelho [10] proponen un enfoque basado coevolutivo, dos subcúmulos interactúan para optimizar la función objetivo y cumplir las restricciones. He, Prempan y Wu [11] investigan la conservación de soluciones factibles, mediante un mecanismo de retroceso del vuelo de las partículas.

4. Nuestro estudio empírico

En los trabajos discutidos anteriormente se utilizan tanto versiones de PSO con valor de inercia como con factor de estrechamiento. Sin embargo, no se especifica si el utilizar el valor de inercia o el factor de estrechamiento podría ayudar a mejorar el desempeño del algoritmo propuesto. El presente trabajo pretende mostrar el desempeño de los algoritmos básicos de PSO utilizando el valor de inercia o el factor de estrechamiento, para poder evaluar su desempeño al resolver problemas con restricciones. A cada algoritmo sólo le será añadido un mecanismo de selección para evaluar dos partículas. Este mecanismo está basado en las reglas de factibilidad propuestas por Deb [12]. Las reglas de factibilidad que se aplican al momento de elegir al líder y en la actualización de la memoria de cada partícula son:

- a) Cuando dos soluciones son factibles, gana aquella solución con el valor de aptitud más alto
- b) Si una solución es factible y la otra no, gana aquella que sea factible
- c) Si las dos soluciones son no factibles gana aquella con un número menor en la suma de violaciones normalizada

4.1 Diseño experimental

Se utilizaron 13 funciones de prueba descritas en la literatura especializada. Por restricciones de espacio no se incluyen los detalles de estos problemas, pero pueden encontrarse en [13]. El conjunto de problemas cuenta con características diferentes en la función objetivo (lineales, cuadráticas, no lineales), así como en las restricciones (lineales, no lineales, igualdad, desigualdad) y también dimensionalidades diferentes. Se determinaron los valores estadísticos como el mejor, la media y la desviación estándar de 30 ejecuciones por cada función, por cada algoritmo. Las variantes implementadas son las siguientes cuatro: Local best con valor de inercia y global best con factor de estrechamiento, además de global best con valor de inercia y con factor de estrechamiento. Los parámetros utilizados para las 4 variantes son: 80 partículas, 8 vecindarios (para local best únicamente), 2000 generaciones, c_1 de 2.7, c_2 de 2.5, valor de inercia de 0.7, factor de estrechamiento de 0.729. Para manejar restricciones de igualdad se manejó una tolerancia $\epsilon=0.0001$.

4.2 Resultados

Los resultados se muestran en la Tabla 1, donde se destacan con negritas los mejores resultados. Cabe remarcar que las estadísticas se calcularon de aquellas ejecuciones (de las 30 realizadas) donde se obtuvieron soluciones factibles. Por ende, una variante que en la mayoría, o en todas, las ejecuciones obtenga soluciones factibles se considerará con un mejor desempeño.

5. Discusión de los resultados

Con base en los resultados de la Tabla 1, se observa que, entre las 2 versiones de global best la que obtuvo mejores resultados fue claramente la que utiliza factor de estrechamiento. De las versiones local best, la mejor fue, también de manera clara, la versión con factor de estrechamiento. De aquí, se puede sugerir que el factor de estrechamiento, sin importar el modelo de comunicación, parece tener un mejor desempeño en este conjunto de problemas de prueba. Por otro lado, con respecto a las 2 versiones más competitivas (global best y local best, ambas con factor de estrechamiento) se puede observar que los resultados de mejor calidad (los más cercanos al óptimo) y consistencia (mejores valores en la media y desviación estándar) fueron obtenidos por la versión global best para la mayoría de las funciones. Sin embargo, fue la versión local best la que, para todos los problemas, pudo localizar soluciones factibles en la mayoría de las ejecuciones (4 de 30 fue el valor más bajo y fue en la función g13, donde ninguna de las otras 3 variantes pudo obtener soluciones factibles). Además, los resultados obtenidos por la versión local best con factor de estrechamiento están muy cercanos a los obtenidos por su contraparte global best. Por lo tanto, podemos concluir que los resultados sugieren que la variante más estable y que provee resultados competitivos es la variante local best con factor de estrechamiento y que la variante global best con factor de estrechamiento es capaz de proveer los mejores resultados, pero no de manera estable.

<p>Tabla 1. Estadísticas de 30 ejecuciones independientes para las 4 variantes de PSO</p> <p>“(n)” indica que no se obtuvieron soluciones factibles en las 30 ejecuciones y “n” es el número de ejecuciones donde si se alcanzó la zona factible. En negrita se destaca el mejor resultado obtenido. “-” indica que en ninguna ejecución se encontraron soluciones factibles</p>					
Problema y mejor solución conocida		Variantes			
		global best (w=0.7)	global best (k=0.729)	local best (w=0.7)	local best (k=0.729)
G01 -15	Mejor	-14.731	-15.0	-14.707	-14.923
	Media	-11.658	-11.381	-13.071	-12.585
	Desviación Est.	2.373	2.727	1.267	1.630
G02 -0.803619	Mejor	-0.566165	-0.584067	-0.538737	-0.558303
	Media	-0.507430	-0.542963	-0.487891	-0.518260
	Desviación Est.	0.019176	0.018466	0.020589	0.022105
G03 -1	Mejor *	-0.042 (21)	-0.055 (17)	-0.067 (26)	-0.014 (26)
	Media	-0.010	-0.008	-0.004	-0.001
	Desviación Est.	0.013	0.018	0.013	0.002
G04 -30665.538672	Mejor	-30663.218	-30665.538	-30655.976	-30664.282
	Media	-30656.435	-30665.538	-30649.130	-30661.590
	Desviación Est.	2.973	7.400E-12	5.392	1.861
G05 5126.498110	Mejor *	-	-	-	5170.276 (10)
	Media	-	-	-	5914.671
	Desviación Est.	-	-	-	326.087
G06 -6961.813876	Mejor	-6959.239	-6957.279	-6956.412	-6956.006
	Media	-6919.254	-6941.739	-6921.738	-6943.246
	Desviación Est.	23.169	10.388	19.717	8.818
G07 24.306209	Mejor	114.447	46.257	81.161	40.265
	Media	190.799	59.771	145.964	65.418
	Desviación Est.	40.861	6.886	33.059	11.146
G08 -0.095825	Mejor	-0.095825	-0.095825	-0.095825	-0.095825
	Media	-0.095824	-0.095825	-0.095825	-0.095825
	Desviación Est.	2.184E-07	4.234E-17	4.905E-09	4.234E-17
G09 680.630057	Mejor	729.405	710.697	732.618	709.680
	Media	834.741	744.651	842.602	754.325
	Desviación Est.	54.654	15.949	57.186	25.746
G10 7049.3307	Mejor	9584.123	8644.47	9681.355	8765.143
	Media	10489.763	9051.668	10546.128	9370.445
	Desviación Est.	511.287	234.597	375.149	266.173
G11 0.75	Mejor *	0.750 (28)	0.749 (16)	0.749	0.750 (28)
	Media	0.844	0.787	0.823	0.821
	Desviación Est.	0.084	0.066	0.075	0.088
G12 -1	Mejor	-0.999 (28)	-0.999 (16)	-0.999	-1.0 (28)
	Media	-0.999	-0.999	-0.999	-0.999
	Desviación Est.	3.767E-05	4.419E-05	4.742E-05	4.330E-05
G13 0.0539498	Mejor	-	-	-	0.831 (4)
	Media	-	-	-	0.952
	Desviación Est.	-	-	-	0.080

6. Conclusiones y trabajo futuro

En este artículo se presentó un comparativo empírico de cuatro variantes de PSO para resolver problemas de optimización con restricciones. Se utilizaron 13 funciones de prueba con distintas características. Los resultados obtenidos muestran que las variantes con factor de estrechamiento fueron claramente superiores que aquellas con factor de inercia. Además, la variante local best con factor de estrechamiento fue la que presentó la mayor estabilidad con resultados competitivos, encontrando soluciones factibles para todos los problemas. Por último, la variante global best con factor de estrechamiento obtuvo los mejores resultados, pero no fue lo suficientemente estable y falló en encontrar soluciones factibles para algunas de las funciones de prueba. Como trabajo futuro se estudiarán con mayor detalle las versiones con factor de estrechamiento con la intención de diseñar un mecanismo más adecuado para el manejo de las restricciones. Además, se utilizarán otras funciones de prueba para aumentar la capacidad de experimentación con estas técnicas.

Referencias

- [1] J. Kennedy and R. C. Eberhart. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, IEEE Service Center, 1995.
- [2] A.P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.
- [3] K.E. Parsopoulos and M.N. Vrahatis. Upso: A unified particle swarm optimization scheme. *Proc. Int. Conf. Computational Methods in Sciences and Engineering (ICCMSE2004)*, VSP International Science Publishers, Vol 1, pages 868–873, 2004.
- [4] G. Toscano-Pulido and C.A. Coello-Coello. A constrained-handling mechanism for particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, Portland, Oregon, USA, pages 1396–1403, IEEE Service Center, 2004.
- [5] J. J. Liang and P.N. Suganthan. Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, Canada, pages 316–323, IEEE Service Center, 2006.
- [6] K.E. Parsopoulos and M.N. Vrahatis. Unified particle swarm optimization for solving constrained engineering optimization problems. *Lecture Notes in Computer Science*, Vol. 3612, pages 582–591, 2005.
- [7] P. Tian M. Kong and X. Li. Novel particle swarm optimization for constrained optimization problems, *Lecture Notes in Computer Science* Vol. 3809, pages 1305–1310, 2005.
- [8] C. Coello-Coello C. A. Cagnina, L. C. Esquivel. A particle swarm optimizer for constrained numerical. *Lecture Notes in Computer Science*, Vol. 4193, pages 910–919, 2006.
- [9] W. Chen Q. Lu. Dynamic-objective particle swarm optimization for constrained optimization problems. *Journal of combinatorial optimization*, 12:409–419, 2006.
- [10] L. Krohling, R. A. dos Santos Coelho. Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 36:1407–1416, 2006.
- [11] E. Prempan S. He and Q.H. Wu. An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization*, 36(5):585–605, 2004.
- [12] K. Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.
- [13] J. J. Liang, T.P. Runarsson, E. Mezura-Montes, P.N. Suganthan M. Clerc, C.A. Coello Coello, and K. Deb. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical Report, Nanyang Technological University, Singapore, March 2006.