Algoritmos Basados en Nubes de Partículas y Evolución Diferencial para el Problema de Optimización Continua: Un estudio experimental

Pablo David Gutiérrez, Isaac Triguero y Francisco Herrera

Resumen— **Los algoritmos evolutivos han demostrado ser una técnica de optimización muy potente para un gran conjunto de problemas de optimización que los métodos analíticos no pueden abordar. Concretamente, para problemas con variables reales podemos destacar a los algoritmos basados en nubes de partículas y evolución diferencial como aquellos que están tomando mucha importancia en la literatura.**

En este trabajo se realiza un estudio de estos métodos, incluyendo las versiones originales y las variantes más relevantes propuestas en la literatura. Este estudio permite analizar las distintas técnicas en función de sus características. Además, se compara el rendimiento de estos algoritmos sobre el conjunto de funciones de prueba desarrolladas para la sesión de optimización continua del CEC'2005. Los resultados obtenidos, contrastados mediante técnicas estadísticas, ponen de manifiesto la necesidad de realizar experimentos más rigurosos, incluyendo aquellos algoritmos que han destacado por su buen rendimiento.

Palabras clave— **Evolución diferencial, nubes de partículas, algoritmos evolutivos, optimización numérica/continua, CEC'2005**

I. Introducción

En muchos problemas de ingeniería se encuentran problemas de optimización de diversa índole. Dentro de estos problemas, un grupo importante, por sus aplicaciones, es el que forman aquellos cuya solución pasa por encontrar el valor máximo o mínimo de una función real. Los llamados problemas de optimización continua.

La resolución de este tipo de problemas se puede abordar desde un enfoque puramente analítico, desde el que se obtiene la solución exacta. Sin embargo, requiere que la función objetivo cumpla determinadas características de continuidad y derivabilidad, además de ser excesivamente costoso cuando aumenta la complejidad de la función. Por otra parte, estos problemas también pueden resolverse desde un enfoque aproximado donde se sacrifica precisión en aras de reducir el coste de la obtención de la solución.

En la actualidad, los algoritmos evolutivos y de inteligencia de enjambre destacan por su buen rendimiento en este ámbito. Concretamente, muchos esfuerzos en investigación se centran en la optimización basada en Nubes de Partículas (Particle Swarm Optimzation, PSO) [7] y la Evolución Diferencial (ED) [8]. Ambas técnicas han sido ampliamente estudiadas en la literatura [9], [10], y lo siguen siendo en la actualidad [11],[12]. La primera se basa en el movimiento de los bancos de peces y las bandadas de aves mientras que la segunda realiza una evolución que otorga una gran importancia a la mutación.

En este trabajo se estudian en profundidad los métodos de PSO y ED, incluyendo las propuestas clásicas y las versiones más relevantes de la literatura. El estudio realizado nos permite analizar cada uno de los métodos en función de sus características, estableciendo una taxonomía de métodos de PSO y ED, respectivamente. Para contrastar el rendimiento de estas técnicas, se lleva a cabo un estudio experimental sobre el conjunto de funciones de prueba propuestas en la sesión especial del IEEE Congress of Evolutionary Computation de 2005 (CEC'2005) [13], usando las condiciones establecidas en éste para realizar una comparación justa de técnicas. Además, se realiza una comparación con el método ganador de esta competición, G-CMA-ES [14] propuesto en 2005, con el fin de comprobar el rendimiento de las técnicas estudiadas. Los resultados completos y el código fuente de los algoritmos analizados pueden encontrarse en el sitio web temático del grupo SCI^{2}S sobre optimización continua `http://sci2s.ugr.es/EAMHCO`.

El resto del trabajo se organiza como sigue: La Sección II presenta el problema y las técnicas empleadas. La Sección III describe la taxonomía realizada clasificando diferentes propuestas existentes en la literatura. La Sección IV muestra el estudio experimental y presenta los resultados. Por último, la Sección V destaca las conclusiones del trabajo.

II. Preliminares

Esta sección repasa algunos conceptos preliminares necesarios. La Sección II-A define formalmente el problema de la optimización continua. La Sección II-B define el algoritmo básico de PSO básico y sus variantes implementadas. Finalmente, la Sección II-C explica el algoritmo de ED y sus variantes.

P.D. Gutiérrez, I. Triguero y F. Herrera son miembros del Dept. de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR. Universidad de Granada. 18071, Granada, España, E-mails: pdgutipe@gmail.com, triguero@decsai.ugr.es, herrera@decsai.ugr.es.

A. La Optimización Continua

Como ya se ha comentado, los problemas de optimización continua son aquellos en los que la función que se quiere optimizar, función objetivo, es una función de variable real. Formalmente, se puede definir el problema como sigue: Dada una función $f : S \longrightarrow \Re$, encontrar un punto $x^* \in S$ tal que $f(x^*) \leq f(x)$ $\forall x \in S$, donde $S \subset \Re^D$. Esta definición responde a la minimización de la función, la definición para el problema de maximización es análoga.

B. Nubes de Partículas

Los métodos de PSO [7] mantienen una población de individuos, en este caso llamados partículas, que se desplazan por el espacio de búsqueda inspirándose en como se mueven las bandadas de aves y los bancos de peces. Inicialmente, las partículas se distribuyen aleatoriamente por el espacio de búsqueda. Las generaciones siguientes se denotan como: $G = 0, 1, \ldots, G_{max}$. Denotamos $p_{i,G}$ a la i-ésima particula en la generación G. Cada una de las partículas explora el espacio de búsqueda desplazándose en una dirección determinada por su velocidad:

$$p_{i,G+1} = p_{i,G} + v_{i,G} \tag{1}$$

donde $v_{i,G}$ representa la velocidad de la partícula i. El cálculo de la velocidad depende de la experiencia de la propia partícula, en base a su recorrido por el espacio de búsqueda (su mejor posición a lo largo de dicho recorrido) y del conocimiento que le aportan las partículas de su entorno:

$$\begin{aligned} v_{i,G+1} =& v_{i,G} + \varphi_1 \cdot rnd() \cdot (pBest_{i,G} - p_{i,G}) \\ &+ \varphi_2 \cdot rnd() \cdot (envBest_{i,G} - p_{i,G}) \end{aligned} \tag{2}$$

Donde $pBest_{i,G}$ es la mejor posición de la partícula, $envBest_{i,G}$ la mejor posición del entorno, $rnd()$ representa un número generado aleatoriamente en el intervalo [0,1] y φ_1 y φ_2 son los parámetros que controlan el uso de la propia experiencia y de del entorno, respectivamente.

No obstante, la implementación habitual del PSO incluye un factor de inercia [15], ω, que multiplica a la velocidad para evitar que crezca de forma excesiva. De este modo la Ecuación 2 queda como sigue:

$$\begin{aligned} v_{i,G+1} =& \omega \cdot v_{i,G} + \varphi_1 \cdot rnd() \cdot (pBest_{i,G} - p_{i,G}) \\ &+ \varphi_2 \cdot rnd() \cdot (envBest_{i,G} - p_{i,G}) \end{aligned} \tag{3}$$

El entorno de una partícula se puede definir de diversas formas. El más sencillo se denomina entorno global, es aquel en el que todas las partículas forman parte del entorno. De este modo a la hora de actualizar la velocidad con el entorno global se utiliza la mejor solución encontrada. Por otro lado, los entornos locales utilizan un número limitado de partículas en el entorno. Existen dos tipos de entorno local: social y geográfico. En los entornos de tipo social las partículas que forman parte del entorno de otra están definidas previamente. Por otra parte, en los entornos geográficos las partículas pertenecientes al entorno se determinan en función de la distancia que las separa de la partícula para la que se genera dicho entorno, seleccionando, habitualmente, las más cercanas. No obstante, este tipo de entorno no suele utilizarse debido a que produce que las partículas se estanquen en óptimos locales con facilidad. También se pueden definir entornos que modifican sus características a lo largo de la ejecución, pero este tipo suele estar ligado a variantes concretas de estos métodos.

B.1 Propuestas Avanzadas para Nubes de Partículas

En la literatura, se puede encontrar un gran número de modificaciones del PSO en distintos aspectos, desde cambios en el cálculo de la velocidad hasta hibridaciones con otras técnicas bioinspiradas. A continuación, se comentan las principales características de cada una de las variantes seleccionadas.

- PSO con Selección (SePSO) [16]: Incorpora el uso de un operador de selección, similar al que usan otras técnicas como los algoritmos genéticos, para mejorar el rendimiento explotando los resultados de las mejores partículas.
- PSO con Constricción (CPSO) [9]: Reemplaza el factor de inercia por un factor de constricción χ modificando el cálculo de la velocidad de la siguiente forma:

$$\begin{aligned} v_{i,G+1} =& \chi \cdot (v_{id,G} + \varphi_1 \cdot rnd() \cdot (pBest_{i,G} - p_{i,G}) \\ &+ \varphi_2 \cdot rnd() \cdot (envBest_{id,G} - p_{i,G})) \end{aligned} \tag{4}$$

- PSO con Stretching (StPSO) [17]: Utiliza una técnica de *stretching* que modifica la función objetivo cuando la búsqueda se estanca en un óptimo local. Esta modificación incrementa el valor de la función objetivo (fitness) de la zona donde se ha producido el estancamiento manteniendo la función original en las zonas donde esta mejora al óptimo local.
- PSO Compuesto (CoPSO) [17]: Se puede considerar una hibridación de PSO y ED, que utiliza a esta última para calcular los valores de φ_1, φ_2 y ω de forma adaptativa en cada iteración.
- PSO Completamente Informado (Fully Informed Particle Swarm, FIPSO) [18]: Consiste en una variación sobre el modelo de factor de constricción, Ecuación 4, que hace uso de todas las partículas del entorno para calcular la velocidad.
- PSO Jerárquico (Hierachical Particle Swarm Optimization, HPSO) [19]: Esta variante organiza las partículas en una jerarquía en forma de árbol. Cada partícula utiliza, como entorno para calcular su velocidad, a su ṕadre. De esta forma las partículas localizadas en la parte más alta del árbol influyen directa o indirectamente sobre el resto. La jerarquía se

actualiza durante la ejecución para llevar a la parte más alta a las mejores partículas.

- PSO Frankenstein (Frankenstein PSO, FPSO) [20]: Esta técnica combina distintas partes de otras variantes de PSO en un único algoritmo. Incluye un modelo en el que la inercia disminuye de forma lineal durante cierto número de iteraciones, un cálculo de la velocidad completamente informado y una topología adaptativa.
- PSO con Aprendizaje Ortogonal (Orthogonal Learning Particle Swarm Optimization, OLPSO) [11]: Esta variante crea una partícula artificial que usa para actualizar la velocidad. Esta partícula se genera combinando la partícula original y la seleccionada del entorno mediante una técnica de aprendizaje ortogonal, que da nombre al algoritmo.

La mayoría de estos métodos pueden utilizar entornos locales y globales. Las que no lo permiten suelen presentar diferentes topologías para los entornos, como los completamente informados, o variantes que siguen la misma filosofía, como los jerárquicos.

C. Evolución diferencial

El algoritmo de ED [8] comienza con una población de NP individuos. La población inicial distribuye a los individuos aleatoriamente por el espacio de búsqueda. Las generaciones siguientes se denotan como: $G = 0, 1, \ldots, G_{max}$. En ED es usual denominar a cada individuo como un vector D-dimensional $X_{i,G} = \{x^1_{i,G}, ..., x^D_{i,G}\}$, llamado vector objetivo.

Tras la inicialización se inicia un proceso iterativo en el que se aplican a cada individuo los operadores de mutación, cruce y selección. El algoritmo ED aplica, en primer lugar, un operador de mutación para generar un vector mutado $V_{i,G}$ respecto a cada individuo $X_{i,G}$, de la población actual. Para cada vector objetivo $X_{i,G}$, su vector mutado asociado es $V_{i,G}= \{V^1_{i,G}, ..., V^D_{i,G}\}$.

El operador de mutación conocido como *ED/Rand/1* es uno de los más sencillos y habituales. Genera un vector mutado de la siguiente forma:

$$V_{i,G} = X_{r^i_0,G} + F(X_{r^i_1,G} - X_{r^i_2,G}) \tag{5}$$

Donde los índices r^i_0, r^i_1 y r^i_2 son enteros mutuamente exclusivos y generados aleatoriamente en el rango $[1, NP]$, y son también diferentes del índice base i. El factor de escala F es un parámetro de control positivo para escalar la diferencia entre vectores. Existen otros operadores de mutación habituales, que introducen un mayor número de individuos en el cálculo, que parten del propio individuo o hacen uso del mejor individuo de la población.

Tras la fase de mutación, el operador de cruce se aplica para cada vector objetivo $X_{i,G}$ y su correspondiente vector mutado $V_{i,G}$, generando un nuevo vector, denominado vector de *prueba* $U_{i,G}$. Aunque existen otros, el operador de cruce binomial es uno de los más sencillos y empleados [21]. Inicializa su resultado con los valores del vector objetivo, y comprueba si cada componente, j, se modifica con los valores del vector mutado generando un número aleatorio, $rnd()$, en el intervalo [0, 1], si éste es menor o igual que un ratio de cruce, RC, se reemplaza dicha componente:

$$U^j_{i,G} = \begin{cases} V^j_{i,G} & \text{Si } rnd() \leq RC \\ X^j_{i,G} & \text{Si } rnd() > RC \end{cases} \tag{6}$$

Finalmente, se selecciona que individuo pasa a la siguiente generación $G + 1$. Si el vector de prueba obtiene una solución igual o mejor que el vector objetivo, este reemplaza a su correspondiente vector objetivo en la siguiente generación; en otro caso, el vector objetivo se mantiene en la población:

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{Si } f(U_{i,G}) \text{ es mejor que } f(X_{i,G}) \\ X_{i,G} & \text{En otro caso} \end{cases} \tag{7}$$

C.1 Propuestas Avanzadas para Evolución Diferencial

En la literatura se pueden encontrar un gran número de propuestas avanzadas del algoritmo de ED. Los parámetros de control (F y RC) determinan, en gran medida, la calidad del algoritmo por lo que muchas de estos métodos incluyan técnicas de adaptación de parámetros. A continuación se comentan las principales características de las variantes utilizadas en este estudio:

- ED Basada en Oposición (Opposition-Based Differential Evolution, OBDE) [10]: Esta técnica utiliza información de la posición opuesta de cada individuo, que tiene un 50 % de probabilidad de ser mejor que la del propio individuo. Cuando el opuesto es mejor se descarta al actual y se toma el opuesto, saltando así a una zona más prometedora.
- ED Auto Adaptativa (Self Adaptative Differential Evolution, SADE) [22]: Esta variante utiliza un mecanismo de auto-adaptación de los parámetros de control, así como de los operadores utilizados para generar el vector de prueba. Mantiene un conjunto de operadores de mutación y cruce que se aplican en distintos momentos de la evolución en base a los resultados obtenidos hasta el momento.
- ED con Vecinos Locales y Globales (Differential Evolution with Global and Local neighborhoods, DEGL) [23]: Este método introduce el uso de entornos en la mutación generando el vector mutado como combinación lineal de una mutación en un entorno global con otra de un entorno local. La mutación de entorno global se calcula en la forma habitual, con todas las partículas, mientras que la local se calcula con las partículas pertenecientes a un entorno social de topología en anillo.
- ED Adaptativa con Archivo Opcional Externo (Adaptative Differential Evolution with Optional

External Archive, JADE) [24]: Esta variante auto adapta los parámetros de control y hace uso de un archivo en el que se almacenan individuos que son reemplazados en la selección. Estos individuos se utilizan como candidatos a individuos aleatorios en el operador de mutación, introduciendo mayor diversidad en generación del vector mutado.

- ED con Búsqueda Local para el Factor de Escala (Scale Factor Local Search in Differential Evolution, SFLSDE) [12]: Este algoritmo puede considerarse un ED memético. Realiza una adaptación de los parámetros durante la evolución, haciendo hincapié en factor de escala F, para el que utiliza dos algoritmos de búsqueda local.

III. Taxonomía de métodos basados en Nubes de Partículas y Evolución Diferencial

En esta sección se propone una taxonomía para los algoritmos que se han estudiado en función de sus características. En la Sección III-A se presenta la clasificación de los métodos PSO. A continuación, la Sección III-B muestra la taxonomía de los métodos de ED.

A. Clasificación de los métodos de Nubes de Partículas

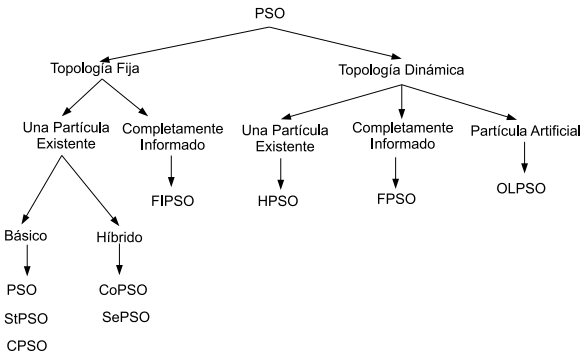

Fig. 1. Taxonomía de los métodos de PSO

La clasificación de las variantes de PSO, que se puede ver en la Figura 1, tiene en cuenta tres aspectos importantes de estos algoritmos:

- La topología del entorno.
- La partícula que utiliza a la hora de calcular la velocidad.
- La hibridación con otras técnicas bioinspiradas.

Dentro de la topología del entorno se pueden distinguir dos grupos de forma clara: los que mantienen la topología fija durante la ejecución y los que la modifican durante la misma. De este modo se observa como la mayor parte de los métodos utilizan una topología fija. No obstante, las variantes que optan por topologías dinámicas se corresponden con algunas de las más recientes.

En un segundo nivel, dentro de cada grupo, el tipo de partícula utilizada para el cálculo de la velocidad permite distinguir entre métodos que usan una única partícula, que pertenece al entorno, métodos que utilizan todas las partículas que pertenecen al entorno y métodos que crean una partícula que puede o no pertenecer al entorno o a la población. Se observa que la mayor parte de los métodos utilizan una partícula del entorno, mientras que solo dos métodos siguen el enfoque alternativo.

Por último, dentro del grupo más amplio (topología fija y una partícula del entorno) se puede distinguir entre métodos híbridos y los que no lo son (básicos).

B. Clasificación de los métodos de Evolución diferencial

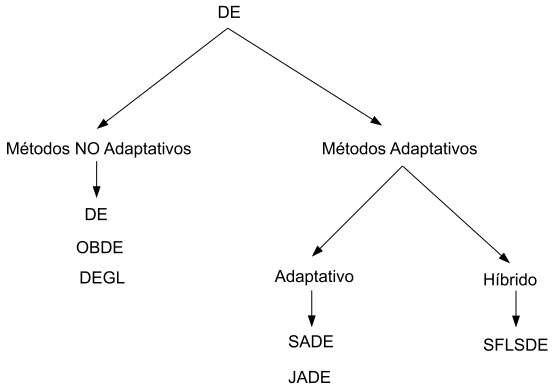

Fig. 2. Taxonomía de los métodos de ED

La clasificación de las variantes de ED, que se puede ver en la Figura 2, es sencilla pero permite identificar claramente los algoritmos. Tiene en cuenta dos aspectos de estos algoritmos:

- La adaptación de parámetros.
- La hibridación con otras técnicas.

El valor de los parámetros en los algoritmos de ED tiene mucha importancia, la clasificación entre métodos adaptativos y no adaptativos permite distinguir de forma muy clara el planteamiento de cada variante en este aspecto.

La distinción de la hibridación permite conocer que tipo de adaptación realiza cada uno de los métodos.

IV. Estudio experimental y análisis de resultados

Para determinar el rendimiento de los algoritmos, se ha realizado un estudio experimental sobre el conjunto de funciones diseñadas para el CEC'2005 [13]. La Sección IV-A describe brevemente estas funciones. La Sección IV-B muestra los parámetros empleados para las distintas técnicas. La Sección IV-C presenta y analiza los resultados obtenidos, junto con un estudio estadístico para contrastarlos.

A. Funciones de prueba

En este estudio, se han empleado las 25 funciones diseñadas para la sesión especial del CEC'2005. Este conjunto presenta una serie de funciones de dificultad creciente que engloba funciones unimodales conocidas, funciones multimodales también conocidas y funciones extendidas e híbridas generadas a partir de las anteriores.

Además se incluyen funciones con discontinuidades, rotadas, escalables, separables y no separables, representando de este modo un conjunto amplio de las características que pueden tener las funciones de problemas reales.

B. Parámetros de los algoritmos estudiados

Todos los algoritmos se han ejecutado con los parámetros estándar propuestos en los artículos que los presentan. No obstante, en el caso de los algoritmos de PSO, al permitir la mayor parte de los mismos el uso de entornos locales y globales, así como presentar distintas topologías y configuraciones se ha optado por seleccionar la mejor configuración dentro de cada algoritmo para su análisis. En la Tabla I se indica la configuración seleccionada para cada método.

TABLA I
PARÁMETROS EMPLEADOS

Método	Parámetros
PSO	$\varphi_1 = 2$, $\varphi_2 = 2$, ω desciende de 0,9 a 0,4 linealmente Población = 40, Entorno global
SePSO	$\varphi_1 = 2$, $\varphi_2 = 2$, ω desciende de 0,9 a 0,4 linealmente Población = 40, Entorno global
CPSO	$\chi = 0{,}7298$, Población = 40, Entorno local social con dos vecinos en topología de anillo
StPSO	$\varphi_1 = 0{,}5$, $\varphi_2 = 0{,}5$, ω desciende de 1 a 0,4 linealmente Población = 40, Entorno local social con dos vecinos en topología de anillo
CoPSO	$\varphi_1 = [4, 0{,}1]$, $\varphi_2 = [4, 0{,}1]$, $\omega = [1{,}2, 0{,}4]$ Población = 40, Entorno global $F = 0{,}5$, $RC = 0{,}9$, Población_ED = 5, IteracionesED = 15
FIPSO	$\chi = 0{,}7298$, Población = 20, Topología Square, Esquema de pesos lineal en función de la calidad de la solución
HPSO	$\varphi_1 = 1{,}7$, $\varphi_2 = 1{,}7$, ω desciende de 0,9 a 0,4 linealmente Población = 40, $d = 5$, $h = 3$
FPSO	$\varphi_1 = 2$, $\varphi_2 = 2$, ω desciende de 0,9 a 0,4 linealmente Población = 40, $top_ltype = 3$, $ine_ltype = 2$
OLPSO	$c = 2$, ω desciende de 0,9 a 0,4 linealmente Población = 40, Entorno global
ED	$F = 0{,}5$, $RC = 0{,}9$, Población = 50
OBDE	$F = 0{,}5$, $RC = 0{,}9$, TasaSalto = 0,4, Población = 50
SADE	Periodo de Aprendizaje = 50, Población = 50
DEGL	$F = 0{,}8$, $RC = 0{,}9$, Población = 50, Vecinos = 2, Esquema de peso exponencial
JADE	$p = 0{,}5$, $c = 0{,}1$, Población = 50
SFLSDE	$Fl = 0{,}1$, $Fu = 0{,}9$, Población = 50, $\tau_1 = 0.1$, $\tau_2 = 0.1$, $\tau_3 = 0.03$, $\tau_4 = 0.07$, iterSFGSS = 8, iterSFHC = 20

Los criterios de evaluación establecidos en el CEC'2005 son los siguientes. Los métodos deben ejecutarse en dimensiones 10, 30 y 50, realizando 25 ejecuciones de cada función, tomando su media y desviación típica. Como criterio de parada, se establece el límite de evaluaciones de la función objetivo como $1000 \cdot Dimension$, o alcanzar un error inferior a 10^{-8}.

C. Resultados obtenidos y estudio estadístico

En este estudio experimental se analizan los resultados en términos de error medio obtenido en las 25 ejecuciones por cada función. Debido a limitaciones de espacio, las tablas con los resultados completos se encuentran en el sitio web `http://sci2s.ugr.es/EAMHCO/srcnew/cec2005results_DEs.zip`. Como resumen, la Tabla II presenta los resultados medios y desviación típica obtenidas en las 25 funciones por cada uno de los algoritmos, en dimensiones 10, 30 y 50. Además, se ha remarcado en negrita el mejor resultado obtenido por cada dimensión y familia.

Para contrastar los resultados experimentales se van a emplear tests no paramétricos de comparaciones múltiples. Su empleo en minería de datos está recomendado en los casos en que se intenten comparar los resultados de un nuevo algoritmo con respecto a varios métodos simultáneamente [25].

Para determinar las mejores propuestas en cada una de las familias, se ha seleccionado el test de Friedman como método para detectar la existencia de diferencias significativas entre los resultados, y el método de Holm como test *post-hoc* para caracterizar las diferencias encontradas [26] [1]. Las Tablas III y IV presentan una comparativa entre los métodos de PSO y ED respectivamente. En cada una de las tablas se han realizado tres test estadísticos, uno por dimensión, anotando el rango obtenido por el test de Friedman, y el p-valor ajustado con el test de Holm. Además se ha señalado en negrita el rango del método que se toma como control (aquel que obtiene el rango más bajo), y aquellos p-valores $< 0{,}05$ indicando que el método es estadísticamente superado por el método de control. Por otro lado, el p-valor asociado al test de Friedman se presenta en la última fila de cada tabla.

A partir de las Tablas II, III y IV se pueden extraer las siguientes conclusiones:

- Atendiendo a la familia de algoritmos PSO, destacamos SePSO y FIPSO como aquellos algoritmos que obtienen los mejores resultados en media. No obstante, los resultados medios pueden llevarnos a una conclusión errónea. En la Tabla III podemos observar como el test de Friedman destaca a FPSO, HPSO y SePSO como los mejores algoritmos en dimensiones 10, 30 y 50 respectivamente.
- En referencia a los algoritmos de ED, podemos señalar a SADE, JADE y SFLSDE como los más destacados. A partir del test estadístico obtenemos esta misma conclusión, sin encontrar diferencias significativas entre estos tres.
- La dimensionalidad de la función a optimizar tiene una gran influencia en el comportamiento de todos

[1] Más información en el sitio web temático de SCI2S sobre *Statistical Inference in Computational Intelligence and Data Mining* `http://sci2s.ugr.es/sicidm/`

TABLA II

TABLA RESUMEN DE LOS RESULTADOS OBTENIDOS

	10		**30**		**50**	
Algoritmo	Media	Desv. Típica	Media	Desv. Típica	Media	Desv. Típica
PSO	7,87E+003	3,13E+004	2,15E+005	9,24E+005	2,33E+006	8,69E+006
SePSO	4,70E+003	2,19E+004	**1,56E+005**	6,22E+005	**6,55E+005**	2,25E+006
CPSO	8,80E+003	4,28E+004	7,09E+005	3,52E+006	3,20E+006	1,59E+007
StPSO	7,34E+003	3,53E+004	3,54E+005	1,74E+006	1,73E+006	8,52E+006
CoPSO	1,60E+004	7,79E+004	8,77E+005	3,08E+006	2,28E+006	8,03E+006
FIPSO	**4,83E+002**	7,64E+002	7,95E+005	2,78E+006	1,34E+007	4,95E+007
HPSO	2,11E+004	1,05E+005	5,81E+005	2,90E+006	1,02E+006	5,08E+006
FPSO	6,62E+002	2,01E+003	2,68E+005	1,18E+006	1,84E+006	7,12E+006
OLPSO	1,22E+005	5,49E+005	1,03E+007	3,74E+007	1,98E+007	6,86E+007
ED	4,74E+002	8,95E+002	1,12E+004	5,01E+004	2,30E+004	1,04E+005
ODBE	3,81E+002	6,86E+002	2,16E+006	1,08E+007	8,44E+006	4,22E+007
SADE	**3,33E+002**	4,74E+002	**7,88E+003**	3,59E+004	2,16E+004	9,79E+004
DEGL	7,28E+002	1,60E+003	4,32E+004	2,06E+005	1,22E+005	5,83E+005
JADE	1,01E+004	4,89E+004	1,35E+004	6,18E+004	**4,24E+003**	1,04E+004
SFLSDE	3,82E+002	6,62E+002	2,86E+004	1,38E+005	6,38E+004	3,10E+005

TABLA III

TEST DE FRIEDMAN+TEST DE HOLM PARA LOS MÉTODOS DE PSO

	10		**30**		**50**	
Algoritmo	Rango	p-valor	Rango	p-valor	Rango	p-valor
PSO	5,48	**0,0098**	4,70	0,8162	4,08	0,3934
SePSO	4,04	0,3934	3,72	0,9177	**3,08***	–
CPSO	3,96	0,3934	4,28	0,8173	4,82	0,1007
StPSO	5,30	**0,0176**	4,72	0,8162	4,88	0,1007
CoPSO	6,44	**0,0000**	5,32	0,1806	5,04	0,0683
FIPSO	4,96	0,0527	6,36	**0,0031**	7,24	**0,0000**
HPSO	4,94	0,0527	**3,64***	–	3,64	0,4697
FPSO	**3,04***	–	4,54	0,8162	4,64	0,1320
OLPSO	6,84	**0,0000**	7,72	**0,0000**	7,58	**0,0000**
	p-valor de Friedman Dimensión 10 = 0.000005					
	p-valor de Friedman Dimensión 30 = 0					
	p-valor de Friedman Dimensión 50 = 0					

* Método de control.

TABLA IV

TEST DE FRIEDMAN+TEST DE HOLM PARA LOS MÉTODOS DE ED

	10		**30**		**50**	
Algoritmo	Rango	p-valor	Rango	p-valor	Rango	p-valor
ED	3,64	0,0690	3,30	0,7240	3,30	0,6794
ODBE	3,18	0,3019	3,74	0,1806	3,82	0,1135
SADE	**2,42***	–	2,92	1,0000	2,82	0,8547
DEGL	5,32	**0,0000**	5,42	**0,0000**	5,32	**0,0000**
JADE	3,68	0,0690	2,94	1,0000	3,08	0,8547
SFLSDE	2,76	0,5205	**2,68***	–	**2,66***	–
	p-valor de Friedman Dimensión 10 = 0.000001					
	p-valor de Friedman Dimensión 30 = 0.000001					
	p-valor de Friedman Dimensión 50 = 0.000002					

* Método de control.

los algoritmos. En general, el test de Holm detecta más diferencias significativas cuando la dimensionalidad es baja. Sin embargo, cuando tratamos problemas con alta dimensionalidad la mayoría de algoritmos obtienen resultados similares.

- Comparando ambas familias de métodos, pode-

TABLA V
COMPARATIVA CON G-CMA-ES: TEST DE WILCOXON

	10			30			50		
G-CMA-ES Vs	R+	R-	p-valor	R+	R-	p-valor	R+	R-	p-valor
SePSO	311,0	14,0	**6,55E-6**	258,0	67,0	**0,0088**	226,0	99,0	0,0903
HPSO	304,0	21,0	**2,66E-5**	247,0	78,0	**0,0219**	198,0	127,0	0,3311
FPSO	278,0	22,0	**6,39E-5**	286,5	38,5	**4,03E-4**	276,0	24,0	**9,08E-5**
SADE	263,0	37,0	**6,49E-4**	217,0	83,0	0,0564	205,0	120,0	0,2457
JADE	298,0	27,0	**7,49E-5**	216,5	108,5	0,1524	217,0	108,0	0,1485
SFLSDE	286,0	39,0	**4,29E-4**	235,0	90,0	0,0516	231,5	68,5	**0,0187**

mos decir que en general los algoritmos de ED son más robustos que los algoritmos de PSO. Los resultados medios obtenidos por los mejores algoritmos de ED difieren en gran medida de los mejores obtenidos por los PSO, especialmente cuando la dimensionalidad del problema es alta.

- Analizando globalmente estas tres tablas, podemos decir que no hay ningún algoritmo que destace sobre el resto para todas las dimensiones.

Una vez realizado el estudio por familias, seleccionamos los tres mejores métodos de PSO y ED, con el fin de compararlos con el algoritmo con mejor comportamiento de la sesión del CEC'2005, G-CMA-ES [14]. Para ello, usamos el test de Wilcoxon [25], comparando G-CMA-ES con los mejores métodos analizados en este trabajo. La Tabla V recoge los resultados de aplicar el test de Wilcoxon en dimensiones 10, 30 y 50. Esta tabla muestra para cada dimensión los rangos positivos (R+) y negativos (R-) conseguidos y su p-valor asociado. Aquellos p-valores remarcados en negrita señalan los métodos mejorados por G-CMA-ES, con un nivel de significancia $\alpha = 0{,}05$.

Los resultados de esta tabla reflejan como en baja dimensionalidad, G-CMA-ES es el mejor algoritmo con diferencias significativas. Conforme aumenta la dimensionalidad G-CMA-ES no consigue que sus diferencias sean estadísticamente significativas respecto del resto de técnicas. Sin embargo, con los rangos obtenidos, podemos afirmar que este algoritmo supera en gran medida a propuestas posteriores a su publicación.

V. CONCLUSIONES

En este trabajo se ha realizado un estudio de los principales algoritmos de PSO y ED propuestos en la literatura. En base a ese estudio se ha propuesto una taxonomía para cada familia que destaca sus características principales. En el estudio experimental se ha observado como, por lo general, las propuestas avanzadas mejoran los resultados de los métodos básicos. Sin embargo, como indica en el estudio estadístico, estos métodos están, en general, lejos respecto a G-CMA-ES. Estas propuestas avanzadas aportan ideas interesantes, pero limitan sus comparaciones a los métodos de una misma familia. En este trabajo se muestra que es necesario comparar con el mejor algoritmo conocido en la temática.

Como trabajo futuro, se plantea llevar un estudio empírico con funciones de alta dimensionalidad, con el fin de comprobar si estos métodos son o no competitivos respecto de G-CMA-ES cuando la dimensión del problema crece.

AGRADECIMIENTOS

Subvencionado por los proyectos TIN2011-28488 y TIC-2010-6858. I. Triguero dispone de una beca FPU del Ministerio de Educación y Ciencia.

REFERENCIAS

[1] E.R. Hansen, *Global Optimization Using Interval Analysis*. Pure and Applied Mathematics Series, M. Dekker, 1992.

[2] S.S. Rao, *Engieering optimization: theory and practice*. Wiley-Interscience publication, Wiley, 1996.

[3] H-G. Beyer, *The Theory of Evolution Strategies*. Natural computing series, Springer, 2001.

[4] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Artificial Intelligence, Addison-Wesley Pub. Co., 1989.

[5] W. Banzhaf, P. Nordin, RE Keller, FD Francone, *Genetic Programming - An introduction*. The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann Publishers, 1998.

[6] D. Fogel, *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. Wiley-IEEE Press, 2005.

[7] J. Kennedy, R. Eberhart *Particle Swarm Optimization*. Procedings IEEE International Conference on Neural Networks 1942–1948, 1995.

[8] Storn R., Price K. V, *A simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization 11 (10), 341–359, 1997.

[9] M. Clerc, J. Kennedy , *The particle swarm-explosion, stability and convergence in a multidimensional complex space*. IEEE Transations on Evolutionary Computation 6 (1) 58–73, 2002.

[10] S. Rahnamayan, H.R. Tizhoosh, M.A. Salama, *Opposition-Based Differential Evolution*. IEEE Transactions on Evolutionary Compututation 12 (1) 64–79, 2008.

[11] Z-H Zhan, J. Zhang, Y. Li, Y-H. Shi, *Orthogonal Learning Particle Swarm Optimization*. IEEE Transactions on Evolutionary Compututation, in press, 2011.

[12] Ferrante N., Tirronen V, *Scale factor local search in differential evolution*. Memetic Computing 1 (2) 153–171, 2009.

[13] P.N. Sganthan, N.Hansen, J.J. Liang, K. Deb, Y-P. Chen, A. Auger, S. Tiwari, *Problems Definitions and Evaluation Criteria for the CEC 2005, Special Session on Real-Parameter Optimization*. IEEE Congress of Evolutionary Computation, 2005.

[14] Hansen, N., S.D. Müller, P. Koumoutsakos, *Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)*. Evolutionary Computation, 11(1) 1–18, 2003.

[15] YH. Shi, R. Eberhart, *A modified particle swarm optimizer* Procedings IEEE World Congress on Computational Intelligence 69–73, 1998.
[16] P. Angelinne, *Using Selection to Improve Particle Swarm Optimization*. Procedings IEEE International Congress on Evolutionary Computation 84–90, 1998.
[17] K.E. Parsopoulos, M.N. Vrahatis, *Recent approaches to global optimization problems through Particle Swarm Optimization*. Natural Computing 1 235–306, 2002.
[18] R. Mendes, J. Kennedy, J. Neves, *The Fully Informed Particle Swarm: Simpler, Maybe Better*. IEEE Transactions on Evolutionary Compututation 8 (3) 204–210, 2004.
[19] S. Janson, M. Middendorf, *A Hierarchical Particle Swarm Optimizer and Its Adaptative Variant*. IEEE Trasnctions on Systems, Man and Cybernetics 35 (6) 1272–1282, 2005.
[20] M.A. Montes de Oca, T. Stützle, M. Birattari, M. Dorigo, *Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm*. IEEE Transactions on Evolutionary Compututation 13 (5) 1120–1132, 2009.
[21] I. Triguero, S. García, F. Herrera, *Differential Evolution for Optimizing the Positioning of Prototypes in Nearest Neighbor Classification*. Pattern Recognition 44 (4), 901–916, 2011.
[22] AK. Qin, VL. Huang, PN. Suganthan, *Differntial Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization*. IEEE Transactions on Evolutionary Compututation 13 (2) 398–417, 2009.
[23] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, *Differential evolution using a neighborhood-based mutation operator*. IEEE Transactions on Evolutionary Computation 13 (3) 526–553, 2009.
[24] J. Zhang, A.C. Sanderson, *JADE: adaptive differential evolution with optional external archive*. IEEE Transactions on Evolutionary Computation 13 (5) 945–958, 2009.
[25] García S., Herrera F, *An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets"for all Pairwise Comparisons*. Journal of Machine Learning Research 9, 2677–2694, 2008.
[26] García S., Fernández A., Luengo J., Herrera F, *Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power*. Information Sciences 180, 2044–2064, 2010.