Algoritmo Basado en Cúmulos de Partículas y Evolución Diferencial para la Resolución de Problemas de Optimización Continua

Jose Manuel Garcia-Nieto[1], Javier Apolloni[2], Enrique Alba[1] y Guillermo Leguizamón[2]

Resumen— **Los algoritmos de optimización basados en Cúmulos de Partículas (Particle Swarm Optimization - PSO) y Evolución Diferencial (Differential Evolution - DE) vienen siendo utilizados satisfactoriamente, desde su creación en la pasada década, en la resolución de problemas complejos de optimización de naturaleza continua. En este trabajo analizamos el comportamiento de una nueva técnica metaheurística, combinando las estrategias de búsqueda y operadores presentes en PSO y DE, con la que pretendemos mejorar los resultados existentes en el estado del arte. Para ello, seguimos el marco experimental propuesto en la sesión especial de optimización continua de MAEB'09 y se realizan comparaciones estadísticas con tres algoritmos: G-CMA-ES, DE y K-PCX, tomados a su vez de la sesión especial de optimización continua de CEC'05. Los resultados obtenidos muestran un alto grado de competitividad de nuestra propuesta con respecto a los algoritmos comparados.**

Palabras clave—**Algoritmo de Cúmulos de Partículas, Evolución Diferencial, Benchmark de Funciones de Optimización Continua de CEC'05**

I. Introducción

Los algoritmos de optimización basados en Cúmulos de Partículas (Particle Swarm Optimization - PSO) [1] y Evolución Diferencial (Differential Evolution - DE) [2] vienen siendo utilizados satisfactoriamente, desde su creación en la pasada década, en la resolución de problemas complejos de optimización de naturaleza continua. Estos problemas se pueden encontrar tanto en el ámbito de la industria como en el académico y consisten básicamente en: *encontrar un* $\mathbf{x}^*$ *tal que* $\forall\ \mathbf{x}\ f(\mathbf{x}^*) \leq f(\mathbf{x})$. Donde $f(.)$ es una función de dominio en el espacio de los reales que modela un problema de optimización, $\mathbf{x} = \{x_1, x_2, \ldots, x_D\}$ es una posible solución a dicho problema, D es el número de variables de la función y $x_i \in [x_i^{inf}, x_i^{sup}]$ $(1 \leq i \leq D)$. Por último, x_i^{inf}, $x_i^{sup} \in \mathbb{R}$ corresponden a los límites inferior (inf) y superior (sup) del dominio de la variable, respectivamente.

En este trabajo estamos interesados en analizar el rendimiento de una nueva técnica metaheurística que llamaremos DEPSO (Differential Evolution Particle Swarm Optimization), consistente en un algoritmo híbrido que toma ideas tanto de PSO como de DE para la resolución de los problemas de optimización continua propuestos en la sesión especial de MAEB'09 (disponible en la URL `http://maeb09.lcc.uma.es/`) [3]. De esta forma, combinando las estrategias de búsqueda, adaptación de parámetros y operadores presentes en PSO y DE pretendemos mejorar los resultados existentes en el estado del arte. Para ello, seguimos el marco experimental propuesto en dicha sesión y se realizan comparaciones estadísticas con tres algoritmos: G-CMA-ES [4], DE [5] y K-PCX [6], tomados de la sesión especial de optimización continua del CEC'05 [7].

El resto de este artículo se organiza de la siguiente forma: en la Sección II se describen brevemente los algoritmos PSO y DE. La Sección III presenta el algoritmo DEPSO dando detalles de su estructura, implementación y funcionamiento. La Sección IV describe el estudio experimental realizado: los parámetros utilizados, el benchmark (CEC'05) de funciones a optimizar, así como los resultados obtenidos. De forma adicional, se realiza un análisis estadístico de los resultados en comparación con los algoritmos G-CMA-ES, DE y K-PCX. Por último, en la Sección V se incluyen las conclusiones y trabajo futuro a realizar continuando con esta línea de investigación.

II. Conceptos Básicos

En esta sección se describen brevemente las técnicas metaheurísticas de los algoritmos basados en Cúmulos de Partículas y Evolución Diferencial.

A. Algoritmos de Cúmulos de Partículas

Los algoritmos de optimización basados en cúmulos de partículas o Particle Swarm Optimization (PSO) [1] fueron desarrollados por Kennedy y Eberhart en 1995. Se trata de una técnica metaheurística basada en población e inspirada en el comportamiento social del movimiento de las bandadas de aves o de los bancos de peces.

En la búsqueda de una solución óptima o cuasi-óptima, PSO actualiza el cúmulo actual de partículas utilizando información acerca de la mejor solución obtenida por cada partícula $(\mathbf{p})$ y la mejor solución obtenida en el cúmulo entero $\mathbf{g}$. La posición de cada partícula $\mathbf{x}_i$ es un candidato a solución de un problema. Cada partícula tiene los siguientes atributos: la velocidad actual $\mathbf{v}_i$, la posición actual $\mathbf{x}_i$, la mejor posición obtenida por la partícula hasta el momento $\mathbf{p}_i$ y la mejor posición encontrada por los vecinos de la partícula hasta el momento $\mathbf{g}_i$. El vecindario de una partícula puede ser *global*, en el cual todas las partículas del cúmulo son consideradas vecinas entre sí, o *local*, en el que sólo son vecinas las partículas inmediatamente cercanas. En la primera fase del algoritmo, se inicializa aleatoriamente la velocidad y la posición de cada partícula del cúmulo.

1 Lenguajes y Ciencias de la Computación. Universidad de Málaga. Campus de Teatinos s/n, 29071, España. E-mail: {jnieto,eat}@lcc.uma.es.

2 LIDIC - Departamento de Informática. Universidad Nacional de San Luis. Ejército de los Andes 950, 5700, Argentina. E-mail: {javierma,legui}@unsl.edu.ar

En la segunda fase, para cada partícula del cúmulo se actualizan la velocidad ($\mathbf{v}_i$) y la posición ($\mathbf{x}_i$) mediante las siguientes ecuaciones:

$$\mathbf{v}_i \leftarrow \omega \cdot \mathbf{v}_i + \varphi_1 \cdot r_1 \cdot (\mathbf{p}_i - \mathbf{x}_i) + \varphi_2 \cdot r_2 \cdot (\mathbf{g}_i - \mathbf{x}_i) \quad (1)$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i \quad (2)$$

donde ω es el factor de inercia [8] mediante el que se controla el balance entre explotación y exploración en la búsqueda, φ_1 y φ_2 son factores de influencia de los coeficientes individual y social de cada partícula (normalmente $\varphi_1 = \varphi_2 = 2$). Por último, r_1 y r_2 son valores uniformemente aleatorios ($r_1 = r_2 = UN(0,1)$).

B. Evolución Diferencial

El algoritmo de Evolución Diferencial (DE) fue propuesto por Storm y Price [2], [9] en 1998. Se trata de una técnica no determinista basada en la evolución de una población de vectores (individuos) de valores reales que representan las soluciones en el espacio de búsqueda. La generación de nuevos individuos se lleva a cabo mediante operadores diferenciales de mutación y cruce.

Mediante la mutación diferencial se añade la diferencia proporcional de dos individuos elegidos aleatoriamente de la población a un tercer individuo (también elegido aleatoriamente). Formalmente, dados tres individuos $\mathbf{v}_{r1}$, $\mathbf{v}_{r2}$ y $\mathbf{v}_{r3}$ elegidos aleatoriamente de la población, donde $r1, r2, r3 \in \{1, 2, \ldots, N\}$ son números aleatorios diferentes entre sí y N es el tamaño de la población, un nuevo individuo mutado $\mathbf{w}_i$ se genera mediante la siguiente expresión:

$$\mathbf{w}_i \leftarrow \mathbf{v}_{r1} + \mu \cdot (\mathbf{v}_{r2} - \mathbf{v}_{r3}) \quad (3)$$

La constante de mutación $\mu > 0$ establece el rango de diferenciación entre los individuos $\mathbf{v}_{r2}$ y $\mathbf{v}_{r3}$ con el objetivo de evitar el estancamiento en el proceso de búsqueda.

Tras la mutación, se realiza una operación de recombinación sobre cada individuo $\mathbf{v}_i$ (*target*) para generar un individuo intermedio $\mathbf{u}_i$ (*trial*). Esta operación de cruce selecciona uniformemente una posición j del vector trial con la misma probabilidad que del vector target obtenido del individuo mutado.

$$u_i(j) = \begin{cases} w_i(j) & \text{if } r \leq Cr \text{ ó } j = j_r, \\ v_i(j) & \text{en otro caso.} \end{cases} \quad (4)$$

Como se puede observar en la Ecuación 4, el operador de cruce elige aleatoriamente un valor entero $j_r \in [1 \ldots D]$ y un valor real aleatorio $r \in (0, 1)$, también uniformemente distribuido para cada componente $j \in (1 \ldots D)$ del vector $trial$ $\mathbf{u}_i$. De este modo, con una probabilidad de recombinación Cr o bien en el caso en el que se cumpla la igualdad $j = j_r$, se selecciona el elemento $j_{ésimo}$ del individuo mutado $w_i(j)$ para ser colocado en el elemento $j_{ésimo}$ del individuo $trial$ $u_i(j)$. En otro caso, se selecciona el elemento $j_{ésimo}$ del individuo $target$ $v_i(j)$ para ser colocado en el elemento $j_{ésimo}$ del individuo $trial$. Finalmente, mediante un operador de selección se decide la aceptación del individuo $trial$ para la nueva generación si consigue alguna mejora sobre el individuo anterior, como muestra la Ecuación 5.

$$\mathbf{v}_i = \begin{cases} \mathbf{u}_i & \text{if } f(\mathbf{u}_i) \leq f(\mathbf{v}_i), \\ \mathbf{v}_i & \text{en otro caso.} \end{cases} \quad (5)$$

En la siguiente sección, se describe el algoritmo objeto de este trabajo en el que se utilizan las estructuras y operadores, tanto de PSO como de DE, con la intención de aprovechar las capacidades de búsqueda intrínsecas de cada uno de estos algoritmos.

III. El Algoritmo DEPSO

Basándonos en los estudios de Swagatam Das *et al.* [10], en los que proponen un acercamiento inicial a la hibridación de PSO y DE para la optimización continua, el algoritmo implementado para el presente trabajo, al que llamamos DEPSO, básicamente utiliza el esquema de variación diferencial que emplea DE para ajustar la velocidad de las partículas en PSO.

El mecanismo de actualización de partículas por variación diferencial proporciona un esquema para una rápida convergencia hacia un óptimo. De este modo, la actualización de la velocidad de las partículas utiliza dos vectores de posición de partículas seleccionados de manera aleatoria. Para cada partícula $\mathbf{x}_i$ de la población se obtiene el vector diferencia $\mathbf{w} = \mathbf{x}_{r1} - \mathbf{x}_{r2}$ donde las partículas $\mathbf{x}_{r1}$ y $\mathbf{x}_{r2}$ son seleccionadas aleatoriamente. La velocidad para la partícula i se calcula utilizando la siguiente ecuación:

$$\mathbf{v}'_i \leftarrow \omega \cdot \mathbf{v}_i + \mu \cdot \mathbf{w} + \varphi \cdot (\mathbf{g} - \mathbf{x}_i), \quad (6)$$

donde ω es el factor de inercia y μ es un factor de escala aplicado al vector diferencia ($\mu = UN(0,1)$). El tercer sumando corresponde al factor social influido por el mejor global de la población $\mathbf{g}$, proporcional al coeficiente social φ (en este caso $\varphi = UN(0,1)$). Así, en el cálculo del vector de la velocidad se reemplaza la experiencia personal de la partícula por el vector diferencial.

De la misma forma que en DE, la actualización de la $j_{ésima}$ componente de velocidad para la partícula i se realiza mediante la Ecuación 7 como sigue:

$$v'_i(j) = \begin{cases} v'_i(j) & \text{si } r \leq Cr, \\ v_i(j) & \text{en otro caso.} \end{cases} \quad (7)$$

Donde $r \in [0, 1]$ es un valor uniformemente distribuido que determina si se escoge la componente j desde la nueva velocidad o desde la velocidad actual en base a la probabilidad de recombinación $Cr \in [0, 1]$. Mediante este mecanismo se permite seleccionar algunas de las componentes del vector de velocidad aumentando la habilidad de explotación del algoritmo. Finalmente, la partícula i cambia de posición sólo si la nueva posición $\mathbf{x}'_i$ mejora respecto a la anterior en el proceso de evolución (asumiendo que se pretende minimizar), en otro caso permanece en la posición actual (ecuaciones 8 y 9).

$$\mathbf{x}''_i = \begin{cases} \mathbf{x}'_i & \text{si } f(\mathbf{x}'_i) \leq f(\mathbf{x}_i) \\ \mathbf{x}_i & \text{en otro caso,} \end{cases} \quad (8)$$

siendo

$$\mathbf{x}'_i \leftarrow \mathbf{x}_i + \mathbf{v'}_i \quad (9)$$

De manera adicional, con cierta probabilidad p_{mut}, se realiza una operación de mutación sobre cada partícula con la intención de evitar una rápida convergencia a un óptimo local. La nueva posición de la partícula $\mathbf{x}'$ se generada utilizando la Ecuación 10.

$$\mathbf{x}' \leftarrow \mathbf{x}^{inf} + UN(0,1) \cdot (\mathbf{x}^{sup} - \mathbf{x}^{inf}) \quad (10)$$

Los vectores $\mathbf{x}^{inf}$, $\mathbf{x}^{sup}$ corresponden a los limites inferiores y superiores respectivamente de cada dimensión de la función que se desea optimizar.

En Algoritmo 1, se muestra el pseudocódigo del modelo híbrido DEPSO implementado para este trabajo.

Algoritmo 1 Pseudocódigo de DEPSO

```
1:  inicializa(C)
2:  mientras no alcance condición de final hacer
3:    para cada partícula x_i de la población hacer
4:      /* Variación diferencial */
5:      para cada dimensión j de la partícula x_i hacer
6:        w(j) ← x_r1(j) − x_r2(j)
7:        si r ≤ Cr entonces
8:          v'_i(j) ← ω · v_i(j) + μ · w(j) + φ · (g(j) − x_i(j))
9:        fin si
10:     fin para
11:     para cada dimensión j de la partícula x_i hacer
12:       x'_i(j) ← x_i(j) + v'_i(j)
13:     fin para
14:     si f(x'_i) ≤ f(x_i) entonces
15:       x''_i ← x'_i
16:     sino
17:       x''_i ← x_i
18:     fin si
19:     /* Mutatión */
20:     si UN(0,1) < p_mut entonces
21:       para cada dimensión j de la partícula x_i hacer
22:         x''_i(j) ← x^inf(j) + UN(0,1) · (x^sup(j) − x^inf(j))
23:       fin para
24:     fin si
25:   fin para
26: fin mientras
27: Salida: Mejor solución encontrada
```

Tras la inicialización previa de la población (cúmulo) C de partículas y su evaluación inicial (línea 1), en cada paso de la evolución se actualizan las posiciones de todas las partículas. Dentro del ciclo de evolución se realiza la variación diferencial (líneas 4 a 18) mediante las ecuaciones anteriormente explicadas y la operación de mutación, si procede (líneas 20 a 24). Además, se actualiza la mejor posición global encontrada hasta el momento para guiar el resto del cúmulo. Finalmente, el algoritmo devuelve la mejor solución encontrada.

IV. Experimentos

El algoritmo DEPSO se ha implementado en C++ utilizando la biblioteca de algoritmos de optimización MALLBA [11]. Para el benchmark de funciones que se desean optimizar se ha utilizado el código fuente, en lenguaje C, disponible en la página web de la sesión especial de optimización continua de CEC'05 [7]. En la realización de los experimentos, se ha seguido el marco experimental propuesto en la sesión especial de optimización continua de MAEB'09 [3].

El benchmark utilizado consta de un subconjunto de las 20 funciones multimodales (desde la función f_6 a la f_{25}) incluyendo funciones básicas en versiones rotadas y/o desplazadas además de funciones compuestas. El óptimo de las funciones está desplazado por un valor de *bias* que permite evitar que el algoritmo de búsqueda se beneficie de la simetría del espacio. Se ha considerado la optimización de estas funciones con espacios de variables de dimensiones 10 y 30.

Sobre cada función del benchmark y cada una de las dimensiones se han realizado 25 ejecuciones independientes. La condición de finalización de cada ejecución requiere que el número de evaluaciones de la función de optimización alcance $10^4 \cdot D$ (siendo D las dimensiones: 10 y 30) o bien cuando el error obtenido sea inferior a 10^{-8}. Las ejecuciones independientes se realizaron utilizando una plataforma de clusters CONDOR sobre máquinas Pentium IV 2.4 GHz con 1GB de RAM y sistema operativo Linux Fedora core 6.

A. Parámetros

El conjunto de parámetros establecido en estos experimentos ha sido el mismo para todas las ejecuciones independientes, funciones del benchmark y dimensiones. En la siguiente tabla se muestran los parámetros empleados.

TABLA I
PARÁMETROS UTILIZADOS PARA LAS EJECUCIONES DE DEPSO

Descripción	Parámetro	Valor
Tamaño de cúmulo	tc	50
Probabilidad de cruce	Cr	0,9
Inercia	ω	0,1 ... 0,5 (f_6 a f_{12}) 0,1 (f_{13} a f_{25})
Mutación diferencial	μ	$UN(0,1)$
Coeficiente social	φ	$1 \cdot UN(0,1)$
Probabilidad de mutación	p_{mut}	$\frac{1}{dimensión}$

Únicamente para las funciones simples (f_6 a f_{12}), se ha utilizado un factor de inercia adaptativo (Ecuación 11) cuyo valor decrece durante la ejecución del algoritmo desde $0{,}5$ (ω_{max}) hasta $0{,}1$ (ω_{min}), respecto al número actual de generaciones ($\#gen_{actual}$) y el número total de generaciones ($\#gen_{total}$).

$$\omega \leftarrow \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \cdot \#gen_{actual}}{\#gen_{total}} \quad (11)$$

B. Resultados

En esta sección se presentan los resultados obtenidos tras los experimentos llevados a cabo con DEPSO. Para facilitar su comparación con otros resultados encontrados en el estado del arte, se recogen en las siguientes tablas el error de los valores obtenidos de cada función del benchmark sobre el óptimo $f(\mathbf{x}^*)$ en el valor de bias establecido ($f(\mathbf{x}) - \mathbf{f}(\mathbf{x}^*)$). En las tablas II y III, se muestran los resultados obtenidos sobre las funciones f_6 a f_{15} y f_{16} a f_{25}, respectivamente con dimensión $D = 10$.

TABLA II

VALOR DE ERROR ALCANZADO POR DEPSO PARA LAS FUNCIONES f_6 A f_{15} CON DIMENSIÓN $D = 10$ Y 100.000 EVALUACIONES DE LA FUNCIÓN OBJETIVO

	Nº Ejecución	$\mathbf{f_6}$	$\mathbf{f_7}$	$\mathbf{f_8}$	$\mathbf{f_9}$	$\mathbf{f_{10}}$	$\mathbf{f_{11}}$	$\mathbf{f_{12}}$	$\mathbf{f_{13}}$	$\mathbf{f_{14}}$	$\mathbf{f_{15}}$
	1ª(Mejor)	4,71E+04	3,12E+00	2,05E+01	3,84E+01	3,53E+01	7,93E+00	2,66E+03	3,93E+00	3,73E+00	3,43E+02
	7ª	1,70E+05	7,11E+00	2,06E+01	4,59E+01	4,80E+01	1,05E+01	6,70E+03	5,31E+00	4,03E+00	4,81E+02
	13ª(Mediana)	2,33E+05	8,04E+00	2,06E+01	4,90E+01	5,58E+01	1,12E+01	9,50E+03	5,78E+00	4,14E+00	5,45E+02
1E+03	19ª	3,49E+05	1,11E+01	2,07E+01	5,27E+01	6,22E+01	1,19E+01	1,17E+04	6,21E+00	4,25E+00	5,99E+02
	25ª(Peor)	5,40E+05	1,21E+01	2,08E+01	5,95E+01	6,57E+01	1,22E+01	1,81E+04	6,50E+00	4,31E+00	6,16E+02
	Media	2,68E+05	8,26E+00	2,07E+01	4,93E+01	5,48E+01	1,11E+01	9,39E+03	5,60E+00	4,11E+00	5,35E+02
	Des. Tip.	1,41E+05	2,60E+00	9,05E-02	5,92E+00	8,09E+00	9,67E-01	4,42E+03	6,81E-01	1,74E-01	7,50E+01
	1ª(Mejor)	5,40E-02	1,00E-02	2,03E+01	9,63E+00	2,25E+01	2,22E+00	2,00E-02	1,83E+00	2,79E+00	1,26E+02
	7ª	4,18E+00	5,70E-02	2,04E+01	1,54E+01	2,76E+01	7,00E+00	1,03E+01	2,24E+00	3,43E+00	2,17E+02
	13ª(Mediana)	4,82E+00	1,41E-01	2,05E+01	1,82E+01	3,36E+01	8,08E+00	1,25E+01	3,07E+00	3,66E+00	2,65E+02
1E+04	19ª	6,21E+00	3,97E-01	2,05E+01	2,24E+01	3,62E+01	8,85E+00	5,52E+01	3,20E+00	3,76E+00	4,02E+02
	25ª(Peor)	9,25E+00	5,61E-01	2,06E+01	2,58E+01	3,84E+01	9,99E+00	7,27E+02	3,33E+00	3,81E+00	4,27E+02
	Media	4,97E+00	2,15E-01	2,05E+01	1,79E+01	3,19E+01	7,61E+00	8,93E+01	2,82E+00	3,56E+00	2,81E+02
	Des. Tip.	2,22E+00	1,90E-01	9,23E-02	4,46E+00	5,04E+00	1,84E+00	1,99E+02	5,01E-01	2,49E-01	9,91E+01
	1ª(Mejor)	0,00E+00	7,00E-03	2,02E+01	0,00E+00	3,01E+00	1,00E-04	0,00E+00	5,25E-01	1,02E+00	0,00E+00
	7ª	2,50E-02	4,40E-02	2,03E+01	9,95E-01	5,97E+00	2,02E-01	9,40E-02	9,13E-01	2,13E+00	6,30E+01
	13ª(Mediana)	4,50E-02	5,70E-02	2,03E+01	1,99E+00	7,97E+00	1,09E+00	1,00E+01	1,41E+00	2,42E+00	8,97E+01
1E+05	19ª	8,70E-02	9,60E-02	2,04E+01	2,99E+00	1,44E+01	1,75E+00	1,88E+01	1,72E+00	2,75E+00	1,39E+02
	25ª(Peor)	4,00E+00	1,30E-01	2,04E+01	3,98E+00	2,11E+01	2,01E+00	7,12E+02	1,87E+00	2,92E+00	4,20E+02
	Media	**4,75E-01**	**6,46E-02**	**2,03E+01**	**1,99E+00**	**1,02E+01**	**1,00E+00**	**3,70E+01**	**1,32E+00**	**2,31E+00**	**1,34E+02**
	Des. Tip.	1,12E+00	3,18E-02	5,01E-02	1,18E+00	5,12E+00	7,65E-01	1,41E+02	4,45E-01	5,42E-01	1,28E+02

TABLA III

VALOR DE ERROR ALCANZADO POR DEPSO PARA LAS FUNCIONES f_{16} A f_{25} CON DIMENSIÓN $D = 10$ Y 100.000 EVALUACIONES DE LA FUNCIÓN OBJETIVO

	Nº Ejecución	$\mathbf{f_{16}}$	$\mathbf{f_{17}}$	$\mathbf{f_{18}}$	$\mathbf{f_{19}}$	$\mathbf{f_{20}}$	$\mathbf{f_{21}}$	$\mathbf{f_{22}}$	$\mathbf{f_{23}}$	$\mathbf{f_{24}}$	$\mathbf{f_{25}}$
	1ª(Mejor)	2,00E+02	1,69E+02	8,43E+02	8,41E+02	4,08E+02	5,66E+02	6,26E+02	6,16E+02	3,02E+02	2,79E+02
	7ª	2,30E+02	2,63E+02	9,32E+02	9,37E+02	5,81E+02	9,85E+02	8,23E+02	1,02E+03	4,17E+02	4,13E+02
	13ª(Mediana)	2,50E+02	2,85E+02	1,01E+03	1,01E+03	7,12E+02	1,16E+03	8,38E+02	1,20E+03	5,98E+02	6,05E+02
1E+03	19ª	2,66E+02	2,93E+02	1,06E+03	1,06E+03	8,33E+02	1,22E+03	8,64E+02	1,22E+03	7,49E+02	7,74E+02
	25ª(Peor)	2,73E+02	3,17E+02	1,08E+03	1,08E+03	9,63E+02	1,24E+03	9,44E+02	1,25E+03	9,10E+02	1,02E+03
	Media	2,47E+02	2,71E+02	9,87E+02	9,98E+02	7,18E+02	1,05E+03	8,44E+02	1,11E+03	5,94E+02	5,96E+02
	Des. Tip.	2,08E+01	3,32E+01	7,10E+01	7,42E+01	1,61E+02	2,23E+02	5,98E+01	1,76E+02	1,96E+02	2,16E+02
	1ª(Mejor)	1,36E+02	1,58E+02	3,00E+02	3,00E+02	2,00E+02	3,00E+02	1,00E+02	3,00E+02	2,00E+02	2,00E+02
	7ª	1,64E+02	1,88E+02	8,00E+02	8,00E+02	2,00E+02	3,00E+02	7,74E+02	3,00E+02	2,00E+02	2,00E+02
	13ª(Mediana)	1,71E+02	1,97E+02	8,00E+02	8,00E+02	2,00E+02	3,00E+02	7,76E+02	3,00E+02	2,00E+02	2,00E+02
1E+04	19ª	1,78E+02	2,05E+02	8,00E+02	9,08E+02	2,00E+02	5,00E+02	8,00E+02	8,00E+02	2,00E+02	2,00E+02
	25ª(Peor)	1,88E+02	2,17E+02	9,33E+02	9,49E+02	2,00E+02	9,00E+02	8,00E+02	1,05E+03	5,00E+02	5,00E+02
	Media	1,68E+02	1,94E+02	7,05E+02	7,83E+02	2,00E+02	4,64E+02	7,37E+02	5,23E+02	2,68E+02	2,48E+02
	Des. Tip.	1,46E+01	1,59E+01	2,21E+02	1,79E+02	0,00E+00	2,12E+02	1,65E+02	2,76E+02	1,25E+02	1,12E+02
	1ª(Mejor)	8,21E+01	1,03E+02	3,00E+02	3,00E+02	2,00E+02	3,00E+02	1,00E+02	3,00E+02	2,00E+02	2,00E+02
	7ª	9,84E+01	1,17E+02	8,00E+02	8,00E+02	2,00E+02	3,00E+02	7,60E+02	3,00E+02	2,00E+02	2,00E+02
	13ª(Mediana)	1,06E+02	1,24E+02	8,00E+02	8,00E+02	2,00E+02	3,00E+02	7,64E+02	3,00E+02	2,00E+02	2,00E+02
1E+05	19ª	1,12E+02	1,29E+02	8,00E+02	9,08E+02	2,00E+02	5,00E+02	8,00E+02	8,00E+02	2,00E+02	2,00E+02
	25ª(Peor)	1,20E+02	1,52E+02	9,32E+02	9,46E+02	2,00E+02	9,00E+02	8,00E+02	1,05E+03	5,00E+02	5,00E+02
	Media	**1,05E+02**	**1,25E+02**	**7,05E+02**	**7,82E+02**	**2,00E+02**	**4,64E+02**	**7,29E+02**	**5,23E+02**	**2,68E+02**	**2,48E+02**
	Des. Tip.	9,49E+00	1,05E+01	2,21E+02	1,78E+02	0,00E+00	2,12E+02	1,63E+02	2,76E+02	1,25E+02	1,12E+02

En dichas tablas se muestran los valores finales de las ejecuciones independientes ordenados de mejor a peor: 1ª (Mejor), 7ª, 13ª (Mediana), 19ª y 25ª (Peor). Estos valores se recogen a las 1.000, 10.000 y 100.000 evaluaciones de función objetivo. Además, se presentan las medias y desviaciones típicas (Des. Tip.). En las tablas IV y V, se muestran los resultados obtenidos sobre las funciones f_6 a f_{15} y f_{16} a f_{25}, respectivamente con dimensión $D = 30$. De esta forma seguimos el formato de tablas establecido en CEC'05 y recomendado en MAEB'09.

Continuando con este protocolo, en la figura Fig. 1 se muestran las gráficas generadas mediante las trazas de la ejecución número 13 (Mediana) de DEPSO sobre las funciones del benchmark con dimensión $D = 30$, en las que se recoge el mejor valor de error obtenido en cada evaluación de función objetivo de las 300.000 evaluaciones totales.

C. Discusión y Análisis

Para el análisis de los resultados, se comparan las medias de los valores de error obtenidos por DEPSO en las 25 ejecuciones independientes con las medias obtenidas por tres algoritmos de referencia presentados en la sesión especial de optimización continua de CEC'05. Tales algoritmos son los siguientes:

- G-CMA-ES [4]: Estrategia Evolutiva adaptando una matriz de Covarianza.
- K-PCX [6]: Algoritmo Genético de Optimización de Estado Estacionario.
- DE [5]: Modelo clásico de Evolución Diferencial para Optimización de parámetros reales.

En esta comparativa hemos hecho uso de los métodos de comparación no paramétricos detallados en [12] ya que, como se muestra en dicho artículo, para las funciones de tests consideradas, no pueden emplearse las funciones paramétricas como t-test al no cumplir las condicio-

TABLA IV

VALOR DE ERROR ALCANZADO POR DEPSO PARA LAS FUNCIONES f_6 A f_{15} CON DIMENSIÓN $D = 30$ Y 300.000 EVALUACIONES DE LA FUNCIÓN OBJETIVO

	Nº Ejecución	$\mathbf{f_6}$	$\mathbf{f_7}$	$\mathbf{f_8}$	$\mathbf{f_9}$	$\mathbf{f_{10}}$	$\mathbf{f_{11}}$	$\mathbf{f_{12}}$	$\mathbf{f_{13}}$	$\mathbf{f_{14}}$	$\mathbf{f_{15}}$
	1ª(Mejor)	7,26E+08	2,92E+01	2,10E+01	1,88E+02	2,13E+02	3,97E+01	1,09E+05	1,92E+01	1,35E+01	4,86E+02
	7ª	2,15E+09	3,99E+01	2,11E+01	2,23E+02	2,68E+02	4,17E+01	1,50E+05	2,36E+01	1,37E+01	5,56E+02
	13ª(Mediana)	3,18E+09	4,88E+01	2,11E+01	2,43E+02	2,79E+02	4,31E+01	1,79E+05	2,46E+01	1,38E+01	6,10E+02
3E+03	19ª	4,77E+09	5,89E+01	2,12E+01	2,60E+02	2,87E+02	4,35E+01	2,10E+05	2,50E+01	1,39E+01	6,49E+02
	25ª(Peor)	9,68E+09	6,39E+01	2,12E+01	2,66E+02	2,94E+02	4,50E+01	2,96E+05	2,59E+01	1,41E+01	7,32E+02
	Media	3,68E+09	4,88E+01	2,11E+01	2,38E+02	2,73E+02	4,27E+01	1,85E+05	2,42E+01	1,38E+01	6,04E+02
	Des. Tip.	2,24E+09	1,05E+01	5,28E-02	2,49E+01	2,04E+01	1,45E+00	5,13E+04	1,55E+00	1,49E-01	7,09E+01
	1ª(Mejor)	4,10E+00	0,00E+00	2,09E+01	5,59E+01	1,50E+02	3,85E+01	1,04E+03	1,56E+01	1,28E+01	2,06E+02
	7ª	2,32E+01	1,70E-02	2,10E+01	1,15E+02	1,97E+02	4,03E+01	4,50E+03	1,70E+01	1,32E+01	2,20E+02
	13ª(Mediana)	2,59E+01	2,50E-02	2,10E+01	1,41E+02	2,08E+02	4,10E+01	8,58E+03	1,80E+01	1,34E+01	3,00E+02
3E+04	19ª	7,36E+01	3,80E-02	2,11E+01	1,52E+02	2,24E+02	4,14E+01	1,17E+04	1,85E+01	1,36E+01	3,30E+02
	25ª(Peor)	2,00E+03	5,40E-02	2,11E+01	1,60E+02	2,33E+02	4,20E+01	2,00E+04	1,92E+01	1,36E+01	4,06E+02
	Media	1,43E+02	2,76E-02	2,10E+01	1,30E+02	2,06E+02	4,08E+01	8,59E+03	1,77E+01	1,34E+01	2,95E+02
	Des. Tip.	3,99E+02	1,36E-02	3,96E-02	3,09E+01	2,19E+01	9,25E-01	5,19E+03	1,06E+00	2,23E-01	7,34E+01
	1ª(Mejor)	0,00E+00	0,00E+00	2,08E+01	1,49E+01	5,38E+01	1,05E+01	4,20E+02	2,42E+00	1,20E+01	2,02E+02
	7ª	1,00E-03	1,00E-02	2,09E+01	2,29E+01	1,50E+02	1,32E+01	1,47E+03	3,97E+00	1,27E+01	2,17E+02
	13ª(Mediana)	5,90E-02	1,00E-02	2,09E+01	2,49E+01	1,74E+02	1,58E+01	2,60E+03	1,10E+01	1,28E+01	3,00E+02
3E+05	19ª	3,99E+00	2,00E-02	2,10E+01	2,69E+01	1,79E+02	3,18E+01	5,08E+03	1,30E+01	1,30E+01	3,29E+02
	25ª(Peor)	8,50E+00	3,20E-02	2,10E+01	3,38E+01	1,91E+02	3,81E+01	8,22E+03	1,58E+01	1,31E+01	4,04E+02
	Media	**1,75E+00**	**1,34E-02**	**2,09E+01**	**2,49E+01**	**1,64E+02**	**2,06E+01**	**3,30E+03**	**9,65E+00**	**1,28E+01**	**2,90E+02**
	Des. Tip.	2,51E+00	7,95E-03	4,63E-02	4,84E+00	2,86E+01	1,06E+01	2,43E+03	4,80E+00	2,76E-01	7,64E+01

TABLA V

VALOR DE ERROR ALCANZADO POR DEPSO PARA LAS FUNCIONES f_{16} A f_{25} CON DIMENSIÓN $D = 30$ Y 300.000 EVALUACIONES DE LA FUNCIÓN OBJETIVO

	Nº Ejecución	$\mathbf{f_{16}}$	$\mathbf{f_{17}}$	$\mathbf{f_{18}}$	$\mathbf{f_{19}}$	$\mathbf{f_{20}}$	$\mathbf{f_{21}}$	$\mathbf{f_{22}}$	$\mathbf{f_{23}}$	$\mathbf{f_{24}}$	$\mathbf{f_{25}}$
	1ª(Mejor)	2,66E+02	3,01E+02	9,07E+02	9,05E+02	9,92E+02	5,20E+02	6,36E+02	5,22E+02	4,38E+02	4,14E+02
	7ª	2,94E+02	3,41E+02	9,18E+02	9,20E+02	1,01E+03	5,34E+02	6,57E+02	5,30E+02	4,94E+02	5,40E+02
	13ª(Mediana)	3,18E+02	3,72E+02	9,27E+02	9,24E+02	1,05E+03	5,51E+02	7,30E+02	5,41E+02	5,56E+02	5,88E+02
3E+03	19ª	3,30E+02	4,44E+02	9,35E+02	9,29E+02	1,16E+03	6,11E+02	8,79E+02	5,86E+02	5,77E+02	6,19E+02
	25ª(Peor)	4,29E+02	4,86E+02	9,37E+02	9,48E+02	1,27E+03	1,00E+03	9,05E+02	8,44E+02	6,92E+02	7,00E+02
	Media	3,27E+02	3,84E+02	9,26E+02	9,25E+02	1,09E+03	6,12E+02	7,67E+02	5,91E+02	5,46E+02	5,78E+02
	Des. Tip.	4,56E+01	5,79E+01	9,44E+00	9,37E+00	1,00E+02	1,43E+02	1,09E+02	1,03E+02	6,67E+01	7,44E+01
	1ª(Mejor)	2,08E+02	2,43E+02	8,58E+02	8,59E+02	9,00E+02	5,09E+02	5,02E+02	5,09E+02	2,38E+02	2,39E+02
	7ª	2,21E+02	2,61E+02	8,63E+02	8,62E+02	9,00E+02	5,09E+02	5,04E+02	5,10E+02	2,42E+02	2,43E+02
	13ª(Mediana)	2,46E+02	2,74E+02	8,64E+02	8,64E+02	9,00E+02	5,10E+02	5,05E+02	5,10E+02	2,44E+02	2,45E+02
3E+04	19ª	2,59E+02	3,01E+02	8,65E+02	8,66E+02	9,00E+02	5,10E+02	5,50E+02	5,10E+02	2,47E+02	2,47E+02
	25ª(Peor)	3,37E+02	3,76E+02	8,69E+02	8,71E+02	1,18E+03	8,00E+02	5,50E+02	5,10E+02	2,49E+02	2,48E+02
	Media	2,52E+02	2,91E+02	8,64E+02	8,64E+02	9,40E+02	5,33E+02	5,21E+02	5,10E+02	2,44E+02	2,45E+02
	Des. Tip.	3,66E+01	4,21E+01	2,75E+00	3,08E+00	9,31E+01	8,04E+01	2,25E+01	2,25E-01	3,14E+00	2,49E+00
	1ª(Mejor)	3,82E+01	6,40E+01	8,35E+02	8,27E+02	9,00E+02	5,09E+02	5,00E+02	5,09E+02	2,32E+02	2,32E+02
	7ª	1,73E+02	2,21E+02	8,55E+02	8,56E+02	9,00E+02	5,09E+02	5,00E+02	5,10E+02	2,34E+02	2,33E+02
	13ª(Mediana)	2,01E+02	2,27E+02	8,59E+02	8,58E+02	9,00E+02	5,10E+02	5,01E+02	5,10E+02	2,34E+02	2,34E+02
3E+05	19ª	2,17E+02	2,48E+02	8,62E+02	8,60E+02	9,00E+02	5,10E+02	5,50E+02	5,10E+02	2,35E+02	2,35E+02
	25ª(Peor)	2,93E+02	3,25E+02	8,65E+02	8,62E+02	1,18E+03	8,00E+02	5,50E+02	5,10E+02	2,35E+02	2,37E+02
	Media	**1,87E+02**	**2,22E+02**	**8,57E+02**	**8,55E+02**	**9,40E+02**	**5,33E+02**	**5,18E+02**	**5,10E+02**	**2,34E+02**	**2,34E+02**
	Des. Tip.	7,11E+01	6,59E+01	8,08E+00	8,57E+00	9,31E+01	8,04E+01	2,43E+01	2,24E-01	7,55E-01	1,34E+00

nes de independencia, normalidad y heterocedasticidad requeridas para tal fin.

Debido al bajo número de algoritmos que comparamos se ha utilizado el test no paramétrico de Ranking por Signos de Wilcoxon [13], mediante el que comparamos DEPSO con cada uno de los algoritmos anteriormente citados. Este test es de una alternativa no paramétrica al t-test por parejas. Su funcionamiento se basa en calcular el valor absoluto de la diferencia entre cada par de valores de la muestra, los resultados son ordenados de mayor a menor y se computa un *ranking* para determinar la posición de cada diferencia, luego se determina el signo de cada elemento del *ranking* de acuerdo al signo de la diferencia anterior y finalmente se computa la suma promedio de los *ranking* divididos en valores positivos $R+$ y negativos $R-$. Si el p-valor calculado por el test es menor que el nivel de confianza adoptado (p-valor=0,05) se rechaza la hipótesis nula y el algoritmo asociado al mayor de los valores es el mejor.

TABLA VI

COMPARACIÓN DE DEPSO CONTRA G-CMA-ES, DE, K-PCX POR LA PRUEBA NO PARAMÉTRICA DE RANKING POR SIGNOS CONSIDEUNO LOS VALORES DE ERROR MEDIO PARA 10 Y 30 DIMENSIONES Y 95 % DE NIVEL DE CONFIANZA (P-VALOR=0,05)

Algoritmo	Dimensión	$R+$	$R-$	p-valor
G-CMA-ES	10	**111**	79	0,520
	30	103	**107**	0,940
DE	10	102	**108**	0,911
	30	82	**128**	0,391
K-PCX	10	66	**144**	0,145
	30	**111**	79	0,520

En la Tabla VI, se muestran los resultados de aplicar el test no paramétrico de Ranking por Signos en la comparación de los resultados de nuestra propuesta (DEPSO) con los tres algoritmos a comparar: G-CMA-ES, DE y K-PCX, en las dimensiones 10 y 30. Al comparar dos algoritmos, por ejemplo DEPSO con G-CMA-ES, el valor

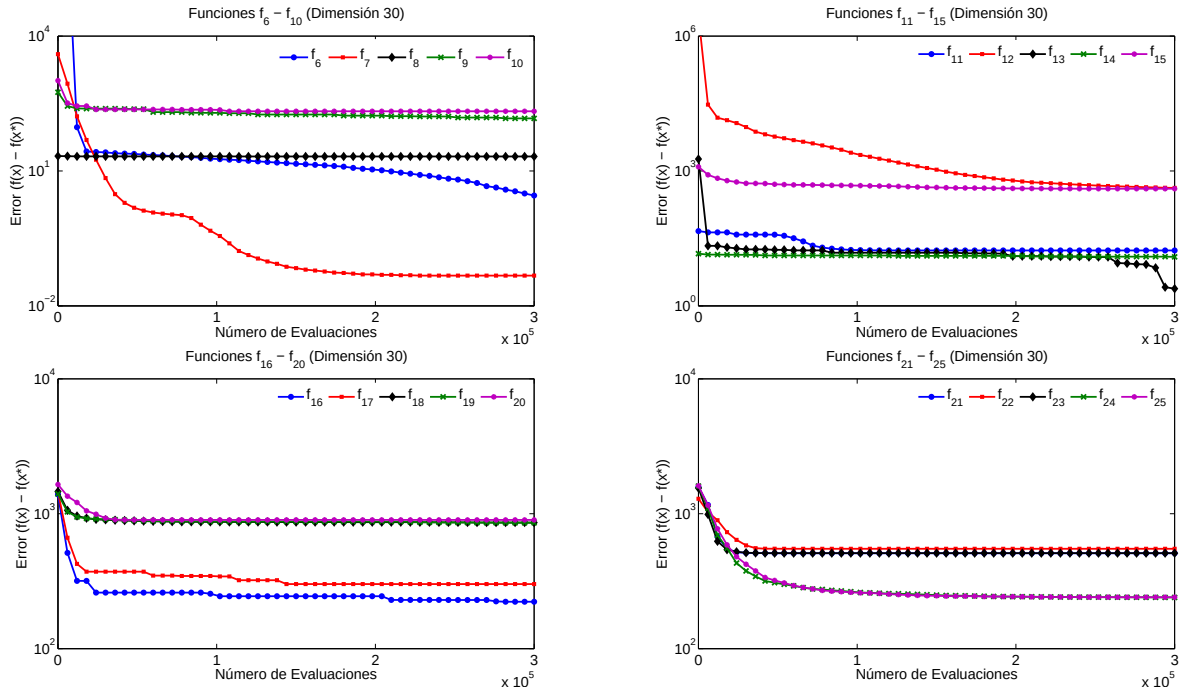

Fig. 1

TRAZAS DE LA EJECUCIÓN NÚMERO 13 (MEDIANA) DE DEPSO SOBRE LAS FUNCIONES DEL BENCHMARK CON DIMENSIÓN $D = 30$ EN LAS QUE SE MUESTRA EL MEJOR ERROR EN ESCALA LOGARÍTMICA EN CADA EVALUACIÓN DE FUNCIÓN

$R+$ indica el ranking promedio donde el algoritmo G-CMA-ES obtiene valores de error medio ($f(\mathbf{x}) - \mathbf{f}(\mathbf{x}^*)$) inferiores al algoritmo DEPSO. Por otro lado, $R-$ muestra el ranking promedio donde el algoritmo DEPSO obtiene valores de error medio inferiores a G-CMA-ES. Como se puede observar en la Tabla VI, para dimensión 10, DEPSO obtiene mejor ranking promedio que DE y K-PCX. Para dimensión 30, DEPSO obtiene mejor ranking promedio que G-CMA-ES y DE. No obstante, en ningún caso se rechaza la hipótesis nula por lo que estadísticamente no se puede asegurar que existan diferencias entre los resultados.

A continuación, pasamos a comparar directamente las medias de los resultados finales de DEPSO (tablas II, III, IV y V) con las medias finales de los algoritmos de CEC'05. En la Tabla VII se resume un listado de los algoritmos a los que DEPSO consigue batir respecto a cada función del benchmark, indicando en las columnas 3 y 5 las posiciones p en las que resulta DEPSO respecto a los algoritmos comparados (Alg. CEC'05). Así, por ejemplo, para la función f_{15}, DEPSO consigue mejores resultados que DE, K-PCX y G-CMA-ES en dimensión 10 ($p = 1$) y mejores resultados que DE y K-PCX en dimensión 30 ($p = 2$). De este modo, se puede observar cómo para dimensión 10, DEPSO obtiene los mejores resultados para 6 funciones y los peores resultados sólo para 4 funciones (símbolo -). Para dimensión 30, DEPSO consigue los mejores resultados para 3 funciones, consigue batir a al menos 2 algoritmos en 6 funciones y no consigue mejorar los resultados tan sólo en 3.

Respecto a DEPSO en particular, a partir de la función f_{14} hasta la f_{25} (exceptuando tres casos) consigue un mejor comportamiento, lo que lleva a pensar que sobre las funciones compuestas el rendimiento es mayor que sobre las funciones simples. Este comportamiento no se observa sin embargo en G-CMA-ES, considerado el mejor algoritmo hasta el momento. Otra interesante observación reside en la mejora que aporta la hibridación de DE con PSO sobre el algoritmo DE básico, ya que para dimensión 10, DEPSO es mejor que DE en 11 de las 20 funciones consideradas y para dimensión 30 DEPSO es mejor que DE en 13 de las 20 funciones consideradas.

TABLA VII

FUNCIONES PARA LAS QUE DEPSO OBTIENE MEJORES RESULTADOS QUE LOS ALGORITMOS MENCIONADOS

Func.	Dimensión 10		Dimensión 30	
	Alg. CEC'05	p	Alg. CEC'05	p
f_6	DE	3	DE, K-PCX	2
f_7	DE, K-PCX	2	K-PCX	3
f_8	DE	3	DE	3
f_9	-	4	-	4
f_{10}	DE	3	-	4
f_{11}	K-PCX	3	DE, K-PC	2
f_{12}	K-PCX	3	C-MA-ES	3
f_{13}	-	4	DE	3
f_{14}	**DE, K-PCX, C-MA-ES**	**1**	**DE, K-PCX, C-MA-ES**	**1**
f_{15}	**DE, K-PCX, C-MA-ES**	**1**	DE, K-PCX	2
f_{16}	DE	3	DE	3
f_{17}	-	4	DE, C-MA-ES	2
f_{18}	K-PCX	3	DE, C-MA-ES	2
f_{19}	-	4	DE, C-MA-ES	2
f_{20}	**DE, K-PCX, C-MA-ES**	**1**	-	4
f_{21}	**DE, K-PCX, C-MA-ES**	**1**	K-PCX	3
f_{22}	C-MA-ES	3	**DE, K-PCX, C-MA-ES**	**1**
f_{23}	**DE, K-PCX, C-MA-ES**	**1**	**DE, K-PCX, C-MA-ES**	**1**
f_{24}	K-PCX	3	C-MA-ES	3
f_{25}	**DE, K-PCX, C-MA-ES**	**1**	DE	3

V. CONCLUSIONES

En este trabajo se propone el uso de una nueva técnica metaheurística llamada DEPSO, mediante la que se combinan las estrategias de búsqueda y operadores presentes en los algoritmos PSO y DE. Tras su evaluación se consiguen mejoras sobre los resultados existentes en el estado del arte. Para ello, seguimos el marco experimental propuesto en la sesión especial de optimización continua de MAEB'09 y se realizan comparaciones estadísticas con otros tres algoritmos: G-CMA-ES, DE y K-PCX, tomados a su vez de la sesión especial de optimización continua de CEC'05. Los resultados obtenidos muestran un alto grado de competitividad de nuestra propuesta, superando para un buen número de las instancias los resultados de algoritmos como CMA-ES que en la actualidad constituyen la referencia en el estado del arte.

Como trabajo futuro se pretende evaluar DEPSO y nuevas hibridaciones de metaheurísticas bioinspiradas con el mismo y nuevos benchmark de funciones utilizando espacios de variables de mayores dimensiones: 50, 100, 500 y 1.000 variables, así como realizar comparativas con nuevos algoritmos encontrados en el estado del arte. Parte de este trabajo ya se está realizando para el benchmark de funciones continuas de CEC'08 [14], estableciendo comparativas con los algoritmos presentados en dicha sesión especial.

AGRADECIMIENTOS

Enrique Alba y José Manuel García-Nieto están parcialmente financiados por el CICE, Junta de Andalucía bajo contrato P07-TIC-03044 (Proyecto DIRICOM, http://diricom.lcc.uma.es). J. Apolloni agradece a la Universidad de Málaga, España y a la Agencia Española de Cooperación Internacional y Desarrollo por la beca de investigación MAEC (BOE 157/2008).

REFERENCIAS

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, Piscataway, NJ., Proceedings of IEEE International Conference on*, pp. 1942–1948, 1995.

[2] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A practical Approach to Global Optimization*, Springer-Verlag, London, UK, 2005.

[3] F. Herrera and M. Lozano, "Sesión Especial en Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua," [online] http://decsai.ugr.es/˜lozano/AEBs-Continuo/AEBs.htm, Feb 2009.

[4] A. Auger and N. Hansen, "A restart cma evolution strategy with increasing population size," *IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1769–1776, 2005.

[5] J. Ronkkonen, S. Kukkonen, and K.V. Price, "Real-parameter optimization with differential evolution," *IEEE Congress on Evolutionary Computation*, vol. 1, pp. 506–513, 2005.

[6] Ankur Sinha, Santosh Tiwari, and Kalyanmoy Deb, "A population-based, steady-state procedure for real-parameter optimization," *IEEE Congress on Evolutionary Computation*, vol. 1, pp. 514–521, 2005.

[7] P.Ñ. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," Tech. Rep., Nanyang Technological University, 2005.

[8] R. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," in *Proceedings of the International Congress on Evolutionary Computation*, July 2000, vol. 1, pp. 84–88.

[9] R. Storn and K. V. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep., TR-95012-ICSI, 1995.

[10] Swagatam Das, Ajith Abraham, and Amit Konar, "Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives," *In Advances of Computational Intelligence in Industrial Systems*, 2008.

[11] E. Alba and MALLBA Group, "Mallba: A Library of Skeletons for Combinatorial Optimisation," in *Proceedings of the Euro-Par*, B. Monien and R. Feldmann, Eds., 2002, vol. LNCS 2400, pp. 927–932.

[12] S. García, D. Molina, M. Lozano, and F. Herrera, "An experimental study about the use of non-parametric tests for analysing the behaviour of evolutionary algorithms in optimization problems," in *MAEB 2007 (V Congreso español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados)*, 2007, pp. 275–285.

[13] R. Wilcox, *New statistical procedures for the social sciences*, Hillsdale, 1987.

[14] K. Tang, X. Yao, P.Ñ. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, , and Z. Yang, "Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization," Tech. Rep., Nature Inspired Computation and Applications Laboratory, USTC, China, November 2007.