$ACO_{\mathbb{R}}$ híbrido con múltiples colonias para problemas de optimización continua

Christian Blum[1], Pedro Cardoso[2], and Francisco Herrera[3]

Resumen—
Este trabajo se centra en algoritmos con colonias de hormigas para resolver problemas de optimización continua. En concreto, se propone una extensión de este tipo de algoritmos hacia un método multi-colonia para incrementar la capacidad de exploración del algoritmo. Asimismo, se estudia la introducción de diferentes optimizadores de búsqueda local para mejorar las soluciones generadas por las hormigas artificiales.

En este trabajo se presentan los resultados obtenidos por este modelo sobre el conjunto de funciones de prueba desarrolladas para la sesión de optimización continua del CEC´05.

Palabras clave—**Optimización con colonias de hormigas; algoritmos con multi-colonias; optimización continua; búsqueda local;**

I. Introducción

Los algoritmos de optimización con colonias de hormigas (Ant Colony Optimization – ACO) [1] son una meta-heurística que inicialmente se presentaron para resolver problemas de optimización combinatoria. Sin embargo, la adaptación de estos algoritmos de ACO a la optimización continua disfruta de una atención cada vez mayor. Entre las primeras aplicaciones de ACO al problema de optimización continua podemos citar los algoritmos *Continuous ACO* (CACO) [2], el algoritmo API [3], y el *Continuous Interacting Ant Colony* (CIAC) [4]. En [5] podemos encontrar otros trabajos más recientes; sin embargo, todos esos trabajos apenas siguen el marco original de ACO. Al contrario, el algoritmo $ACO_{\mathbb{R}}$, propuesto en [6], [7], y más tarde utilizado para el entrenamiento de redes neuronales *feedforward* [8], intenta seguir el espíritu de la meta-heurística ACO tan estrechamente como sea posible. En [7] se muestra que $ACO_{\mathbb{R}}$ obtiene ventajas notables con respecto a las variantes ACO existentes cuando se aplica sobre funciones continuas.

En este trabajo proponemos una extensión de $ACO_{\mathbb{R}}$ para trabajar con varias colonias de hormigas artificiales al mismo tiempo. Este tipo de algoritmo se conoce comúnmente como un método multi-colonia. El objetivo de la introducción de múltiples colonias es mejorar la capacidad de exploración del algoritmo tratando de evitar los óptimos locales de baja calidad al comienzo del proceso de búsqueda. Además, se estudia la introducción de diferentes optimizadores de Búsqueda Local (BL), de propósito general, para mejorar las soluciones generadas por las hormigas artificiales.

Este trabajo se organiza de la siguiente manera. En las Secciones II y III presentamos nuestra propuesta de multi-colonia, $ACO_{\mathbb{R}}$, y los métodos de BL utilizados, SW y Simplex. En la Sección IV se explica como se hace la integración de los métodos anteriores. En la Sección V mostramos el estudio experimental. Finalmente, la Sección VI presenta las conclusiones y el trabajo futuro.

[1] ALBCOM research group, Universitat Politècnica de Catalunya, E-mail: cblum@lsi.upc.edu.
[2] Universidad del Algarve, E-mail: pcardoso@ual.pt
[3] Dpto. Ciencias de la Computación e I.A., Universidad de Granada, E-mail: herrera@decsai.ugr.es.

II. Propuesta de algoritmo $ACO_{\mathbb{R}}$

En general, un algoritmo de ACO intenta resolver los problemas de optimización iterando en las dos etapas siguientes:
1. Construcción de soluciones candidatas de una forma probabilística sobre el espacio de búsqueda, mediante el uso de una distribución de probabilidad;
2. Las soluciones candidatas son utilizadas para modificar la distribución de probabilidad de manera a que se oriente las búsquedas futuras en la dirección de soluciones de alta calidad.

En este trabajo consideramos la minimización de una función continua con dominios definidos por intervalos en cada dimensión:

$$f : \mathbb{R}^n \mapsto \mathbb{R}$$

Una solución $\vec{\mathbf{x}} \in \mathbb{R}^n$ es una solución admisible, si y sólo si $l_i \leq x_i \leq u_i$ para $i = 1, \ldots, n$. Así, x_i es el valor de $\vec{\mathbf{x}}$ en la dimensión i, l_i denota el límite inferior del intervalo de la dimensión i, y u_i denota el correspondiente límite superior.

En la sección siguiente se describe el funcionamiento básico del $ACO_{\mathbb{R}}$. A continuación se explica la extensión del modelo a varias colonias y la introducción de BL.

A. Algoritmo $ACO_{\mathbb{R}}$

$ACO_{\mathbb{R}}$ es un algoritmo iterativo que trabaja con un archivo de soluciones A. Este archivo almacena en cualquier momento un número limitado de k de soluciones. El archivo mantiene las soluciones ordenadas por medio de su valor de función objetivo. Sea $\vec{\mathbf{x}}^l$ la l-ésima solución en A, con $f(\vec{\mathbf{x}}^1) \leq \cdots \leq f(\vec{\mathbf{x}}^n)$. Consideremos de ahora en adelante que el i-ésimo valor de la l-ésima solución en A es x_i^l. Al comienzo del algoritmo, el archivo se rellena con k soluciones generadas aleatoriamente por la selección para cada dimensión i de un valor al aleatorio, de manera

uniforme, sobre $[l_i, u_i]$. A continuación, en cada iteración, se genera probabilísticamente un conjunto de m soluciones y se añade a A. Con el fin de limitar el tamaño de A de nuevo a k, las soluciones de A se ordenan, y las peores m soluciones son eliminadas. Este proceso orienta la búsqueda hacia las mejores soluciones encontradas.

Para la construcción probabilística de una solución, $\text{ACO}_{\mathbb{R}}$ usa una función de densidad de probabilidad llamada un *kernel Gaussiano* PDF. De esta manera, un *kernel Gaussiano* se define como la suma ponderada de varias funciones Gaussianas de una dimensión, $g^l(x)$:

$$G_i(x) = \sum_{l=1}^{k} \omega^l g^l(x) = \sum_{l=1}^{k} \omega^l \frac{1}{\sigma^l \sqrt{2\pi}} e^{-\frac{(x-\mu^l)^2}{2\sigma^{l2}}} \quad (1)$$

Debemos tener en cuenta que el índice i denota que $G_i(x)$ es el *kernel Gaussiano* PDF definido para la dimensión i del problema a ser considerado. Esto significa que, en principio, deben definirse n *kernels Gaussianos* PDF para la construcción de una solución. Con el fin de definir el *kernel Gaussiano* PDF, $G_i(x)$, los parámetros ω^l, μ^l, y σ^l deben tener un valor por cada $l = 1, \ldots, k$. El parámetro ω^l es el peso, el parámetro μ^l la media, y el parámetro σ^l la desviación estándar del *kernel Gaussiano* $g^l(x)$. Además, un *kernel Gaussiano* PDF $G_i(x)$ es fácil de muestrear y, sin embargo, proporciona una herramienta poderosa para describir distribuciones con distintas formas.

El *kernel Gaussiano* $g^l(x)$ corresponde a la solución $\vec{\mathbf{x}}^l$ del archivo A. Para la construcción de una nueva solución $\vec{\mathbf{x}}$, una hormiga artificial realiza n pasos de construcción. En el paso de construcción i-ésimo, la hormiga elige un valor para la dimensión i de $\vec{\mathbf{x}}$ $(i = 1, \ldots, n)$. Esto se realiza mediante el muestreo del *kernel Gaussiano* PDF $G_i(x)$, que se deriva de la siguiente manera a partir de las k soluciones del archivo almacenado en A. En primer lugar, se calcula el peso ω^l del *kernel Gaussiano* $g^l(x)$ de acuerdo con la siguiente fórmula:

$$\omega^l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}}, \quad (2)$$

El peso ω^l se define como el valor de la función de Gauss con argumento l, con una media de 1, y la desviación estándar qk, dónde q es un parámetro del algoritmo. Esto significa que cuanto menor sea l, mayor es el peso de $g^l(x)$. En otras palabras, esto significa que los *kernels Gaussianos* que corresponden a las mejores soluciones del archivo tienen un mayor peso que el resto. Además, cuando q es pequeña, se prefieren las mejores soluciones, y cuando es grande, la distribución del peso se vuelve más uniforme. La influencia de q en $\text{ACO}_{\mathbb{R}}$ es similar a ajustar el equilibrio de los métodos de actualización de feromona utilizados ACO para la optimización combinatoria entre *"mejor-iteración"* y *"mejor hasta ahora"*, (véase, por ejemplo, [9]).

El muestreo de $G_i(x)$ se lleva a cabo en dos fases. La primera fase consiste en elegir una de las funciones Gaussianas correspondientes al *kernel Gaussiano* PDF. La probabilidad, $\mathbf{p^l}$, de elegir la l-ésima función Gaussiana viene dada por:

$$\mathbf{p}^l = \frac{\omega^l}{\sum_{s=1}^{k} \omega^s} \quad (3)$$

Una vez que la función Gaussiana $g^r(x)$ se elige probabilísticamente, la segunda fase consiste en calcular una muestra para dicha función de Gauss escogida. Esto puede hacerse utilizando un generador de números aleatorios de acuerdo con una distribución normal definida por un conjunto de parámetros, o usando un generador aleatorio uniforme en conjunción con, por ejemplo, el método Box-Muller [10]. Este muestreo de dos fases es equivalente a la toma de muestras del kernel Gaussiano PDF $G^i(x)$, tal como se define en la Ecuación 1. En función de las limitaciones definidas por los dominios de intervalos, a veces sucede que los valores incluidos en la muestra no son admisibles. En este caso, se descarta el valor y se toma una nueva muestra.

Sin embargo, antes de generar la muestra de $g^r(x)$, se deben definir los parámetros μ^r y σ^r. En primer lugar, μ^r se establece simplemente como el valor x_i^r, es decir, el valor del i-ésima dimensión de la r-ésima solución en el archivo A. En segundo lugar, tenemos que especificar la desviación estándar de σ^r de $g^r(x)$. Para ello se calcula la distancia promedio de x_i^r a los valores de dimensión i del resto de soluciones del archivo. El resultado se multiplica por el parámetro ξ:

$$\sigma^r = \xi \sum_{l=1}^{k} \frac{|x_i^l - x_i^r|}{k-1} \quad (4)$$

El parámetro $\xi > 0$ tiene un efecto similar al de la tasa de evaporación de feromonas en los algoritmos ACO para la optimización combinatoria. Cuanto más alto sea el valor de ξ, menor es la velocidad de convergencia del algoritmo. Mientras que la tasa de evaporación de feromonas influye en la memoria a largo plazo en ACO – es decir, soluciones de calidad inferior se olvidan más rápido – el ξ en $\text{ACO}_{\mathbb{R}}$ influye en el modo en que se utiliza la memoria a largo plazo – es decir, soluciones de inferior calidad tienen en promedio una menor influencia en la construcción de nuevas soluciones.

Por último, señalar que la función de identificador de Gauss (r) se determina sólo una vez por iteración y hormiga, utilizando las ponderaciones ω^l como se indicó antes. Esto significa que r se determina sólo una vez a la hora de elegir un valor de dimensión $i = 1$. Para todas las dimensiones restantes, se utiliza el r elegido para $i = 1$. Por supuesto, la función de

Gauss muestreada difiere en cada paso de construcción, porque los parámetros μ^r y σ^r son diferentes para cada dimensión.

B. Extensión a multi-colonias

Hemos ampliado el algoritmo $\text{ACO}_{\mathbb{R}}$ original como se ha señalado anteriormente, con el objetivo de trabajar con múltiples colonias. Concretamente, hemos introducido una lista ordenada C que contiene n_c colonias. De ahora en adelante la i-ésima colonia en C es denotada por $C[i]$. Cada colonia $C[i]$ funciona exactamente de la misma manera que el algoritmo $\text{ACO}_{\mathbb{R}}$ original. Sin embargo, de vez en cuando las diferentes colonias hacen un intercambio de información. A continuación diferenciamos entre iteraciones del algoritmo, e iteraciones de una colonia. De hecho, una iteración del algoritmo consiste en una iteración de cada colonia. El intercambio de información funciona de la siguiente manera: cada vez que han transcurrido t iteraciones del algoritmo, la colonia $C[i]$ envía la mejor solución en su archivo hasta el archivo de la colonia $C[i+1]$ $(i = 1, \ldots, n_c - 1)$. La última colonia $C[n_c]$ envía la mejor solución de su archivo a $C[1]$. Esto significa que la vecindad que se define entre las colonias para el intercambio de información tiene la forma de un anillo. Después, cada colonia elimina la peor solución en su archivo a fin de mantener el tamaño del archivo constante (con k soluciones). Los parámetros n_c y t son, de este modo, parámetros importante del algoritmo.

III. Búsqueda Local

En esta sección incluimos una descripción de los dos métodos de BL empleados para el algoritmos: el algoritmos Solis y Wets (SW) [11] y el método Simplex de Nelder y Mead [12]. Ambos son métodos aleatorios con estructura de ascensión de colinas: siguen un proceso iterativo, en el cual en cada paso se genera una nueva solución a partir de la solución actual, que reemplazará a la anterior en caso de que la mejore.

A. Método Solis y West

Este método de BL aleatorio [11]sigue un esquema de ascensión de colinas con un tamaño de salto adaptativo. Se caracteriza en varios aspectos:

- Las nuevas soluciones se obtienen mediante la suma de una variable de incremento (generada mediante una distribución normal $N(0, \rho)$) y una variable *bias*. Esta variable *bias* permite mantener una inercia, orientando la búsqueda hacia direcciones exitosas en iteraciones anteriores.
- En cada paso considera una dirección, y si el resultado no se mejora, entonces toma la contraria.
- Se cuenta el número de éxitos (soluciones que mejoran a la actual) o fallos (soluciones peores que la actual) consecutivos. En función de unos valores máximos, el parámetro ρ se incrementa o decrementa para aumentar o disminuir el espacio de búsqueda sobre la solución actual.

B. Método Simplex Nelder-Mead

El método Simplex [12] se basa en un esquema descendente. Se denomina Simplex a una figura geométrica en n dimensiones compuesta de $n+1$ puntos $s_0 \cdots s_n$. Uno de los puntos del Simplex es la solución actual, y los otros n puntos se utilizan para definir un vector de direcciones a lo largo de las n dimensiones. Sea s_0 el punto inicial, los n puntos restantes son generados mediante la expresión $s_i = s_0 + \lambda e_j$, donde los e_j son n vectores unitarios y λ es una constante generalmente igual a la unidad.

Sobre dicha estructura simplex se definen una serie de transformaciones geométricas elementales (reflexión, contracción, expansión y multi-contracción), que permitan mover, expandir o contraer el simplex. En cada paso, para seleccionar la operación a aplicar, el método utiliza los valores de la función a optimizar para los vértices del simplex considerado. Después de aplicar cada transformación, el peor vértice considerado será sustituido por un nuevo vértice con mejor valor para la función objetivo.

IV. Introducción de la BL en el algoritmo $\text{ACO}_{\mathbb{R}}$

Es comúnmente aceptado que los procesos de las meta-heurísticas pueden ser mejorados por la inclusión de operadores de BL [13], conduciendo a la obtención de un algoritmo memético o híbrido. En la literatura especializada se han estudiado diversos tipos de hibridaciones, como por ejemplo el uso de soluciones iniciales obtenidas a través de otro método aparte del principal (usado en los algoritmos genéticos para obtener la población inicial), o la combinación de dos o más métodos coordinados para mejorar las soluciones [14]. Mientras que en el primer caso los métodos no están considerados en general como métodos híbridos, para el segundo caso si estamos tratando con métodos híbridos. En esta clase de algoritmos tenemos, por ejemplo, combinaciones de métodos en el siguiente sentido: (a) un conjunto de métodos principales generan aproximaciones a las soluciones con mayor o menor calidad, (b), y son seguidos por otro(s) método(s) que van a perfeccionar las soluciones anteriores.

En nuestro caso particular, tenemos en cuenta dos variantes híbridas al $\text{ACO}_{\mathbb{R}}$: el $\text{ACO}_{\mathbb{R}}^{\text{sw}}$ y el $\text{ACO}_{\mathbb{R}}^{\text{simplex}}$. Como el nombre sugiere, la primero utiliza el operador de BL SW para intentar mejorar las soluciones obtenidas por el $\text{ACO}_{\mathbb{R}}$, mientras que el segundo utiliza el operador de BL Simplex (Sección III). Sin embargo, el uso de estos operadores está restringido en el sentido de que se aplican sólo si las soluciones construidas por el $\text{ACO}_{\mathbb{R}}$, x, son consideradas de calidad. En otras palabras, en un problema de minimización el operador de BL se aplica cuando

$$\frac{f(x) - f\left(x^{best}\right)}{\left|f\left(x^{best}\right)\right|} < \epsilon, \tag{5}$$

Parámetros de BL		
ϵ	-0.5	Ecuación 5
nsteps	50	Número de pasos de la BL
$ACO_{\mathbb{R}}$		
n_c	3	Número de colonias
m	10	Soluciones generadas por iteración/colonia
k	20	Tamaño del archivo
ξ	1.0	Ecuación 4
q	0.1	Ecuación 2

TABLA I
PARÁMETROS USADOS EN LOS EXPERIMENTOS

dónde x^{best} es la mejor solución conocida por la colonia y $\epsilon \in \mathbb{R}$ es un parámetro que influye en el número de veces que se aplica la BL. Mayores valores de ϵ implican que el operador de BL se aplica a la mayoría de las soluciones generadas por el $ACO_{\mathbb{R}}$. Valores más pequeños harán lo contrario. En particular, para $\epsilon < 0$ solamente las soluciones que mejoren la mejor solución de la colonia entrarán dentro de la BL.

V. Estudio Experimental

Para este trabajo, nuestros resultados experimentales siguen las directrices descritas para la *Sesión Especial en Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua* del MAEB'09. Esto incluye el uso del *benchmark* de 20 funciones ($F6 - F25$) que se definen en [15].

En el estudio experimental se han considerado tres métodos: el $ACO_{\mathbb{R}}$ multi-colonia sin BL ($ACO_{\mathbb{R}}$), un $ACO_{\mathbb{R}}$ multi-colonia combinado con la BL SW ($ACO_{\mathbb{R}}^{sw}$), y un $ACO_{\mathbb{R}}$ multi-colonia combinado con la BL Simplex ($ACO_{\mathbb{R}}^{simplex}$). Todos estos métodos están implementados en C++ y han sido ejecutados 25 veces para cada función de prueba, sobre un sistema operativo Linux en un procesador Intel Pentium IV ^{TM} a $3GHz$ y $1GB$ de RAM. Se han considerado las dimensiones $D = 10$ y $D = 30$ para cada problema, con un límite de $10{,}000 \times D$ evaluaciones de la función objetivo. Los restantes parámetros de los algoritmos, establecidos por un conjunto de estudios preliminares, están resumidos en la Tabla I.

A. Resultados para $D = 10$

Los resultados para el mejor de los casos, peor de los casos, promedio y desviación estándar de errores ($E = f(x) - f(x^*)$, donde x^* es la solución del problema), se resume en las Tablas II, III, IV, y V. Del análisis de dichas tablas podemos concluir que:

- El $ACO_{\mathbb{R}}$ tiene el menor error medio para 11 de las 20 funciones de prueba, mientras que $ACO_{\mathbb{R}}^{sw}$ tiene el menor error medio para 8 de esas funciones[1]
- $ACO_{\mathbb{R}}^{simplex}$ tiene el menor error medio en un único caso, pero tiene el segundo mejor error medio en 11 de las 20 funciones de prueba.
- Considerando solamente las funciones $F5 - F14$ podemos observar que el $ACO_{\mathbb{R}}$ tiene el mejor error medio en 8 de los casos. Al contrario, si consideramos las funciones de prueba $F15 - F25$ entonces es el $ACO_{\mathbb{R}}^{sw}$ que tiene el menor error medio en 8 casos contra 3 del $ACO_{\mathbb{R}}$.

B. Resultados para $D = 30$

Para la dimension $D = 30$, los resultados para el mejor de los casos, peor de los casos, promedio y desviación estándar de los errores, se resumen en las Tablas VI, VII, VIII, y IX.

Del análisis de estas tablas algunas conclusiones se pueden deducir:

- Al igual que para D = 10, el $ACO_{\mathbb{R}}$ tiene el menor error promedio para 11 de las 20 funciones de prueba. Además, se clasifica en segundo en 8 casos.
- El $ACO_{\mathbb{R}}^{sw}$ tiene el menor error promedio para 8 de las 20 funciones.
- $ACO_{\mathbb{R}}^{simplex}$ tiene el menor error promedio en 4 casos, y el peor error en 10 de las 20 funciones de prueba.
- Considerando de nuevo las funciones $F5 - F14$ podemos observar que el $ACO_{\mathbb{R}}$ tiene el mejor error medio en 6 de los casos (mas un caso en que tiene el mismo resultado que el $ACO_{\mathbb{R}}^{simplex}$). Si consideramos las funciones de prueba $F15 - F25$ entonces el $ACO_{\mathbb{R}}^{sw}$ obtiene el menor error medio en 8 casos, estando en igualdad con el $ACO_{\mathbb{R}}$ en 6 de eses casos. Para eses últimos casos también podemos verificar que para todos ellos por lo menos uno de los métodos con BL ha logrado obtener el menor erro.

C. Resultados globales

Teniendo en cuenta todas las funciones de prueba y las dimensiones consideradas, podemos concluir que los métodos logran resultados análogos, en el sentido de que calculan soluciones con media y desviación estándar de errores con órdenes de magnitud similar para cada una de las funciones de prueba.

Sin embargo, utilizando la media de error como valor de comparación, $ACO_{\mathbb{R}}$ obtiene los mejores resultados para la "primera serie de funciones" ($F6 - F14$). Para estas funciones, $ACO_{\mathbb{R}}$ tiene la mejor media de error en 15 de los 18 casos.

Por otra parte, el $ACO_{\mathbb{R}}^{sw}$ tiene mejores resultados en comparación con los otros métodos para las funciones más difíciles ($F15 - F25$). Para estas funciones, $ACO_{\mathbb{R}}^{sw}$ obtiene la mejor media de error en 15

[1]Tenga en cuenta que en caso de empate de los métodos, a ambos se le otorga la mejor clasificación, es decir, si los métodos A y B tienen error medio igual a x y el método C error medio igual a $y > x$ entonces A y B se clasifican en primer lugar y C en el tercer lugar.

TABLA II

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F6$ A $F10$ CON $D = 10$

		F6	F7	F8	F9	F10
$ACO_{\mathbb{R}}^{sw}$	Best	5,16E-03	3,45E-02	2,02E+01	3,98E+00	2,98E+00
	Worst	7,17E+01	8,95E-01	2,05E+01	1,99E+01	3,58E+01
	Mean	**4,22E+00**	**2,77E-01**	**2,04E+01**	**1,13E+01**	**1,55E+01**
	s.d.	**1,43E+01**	**1,98E-01**	**6,74E-02**	**4,54E+00**	**8,25E+00**
$ACO_{\mathbb{R}}^{simplex}$	Best	1,20E-03	1,03E-01	0,00E+00	3,98E+00	4,97E+00
	Worst	8,50E+01	7,85E-01	2,05E+01	2,89E+01	3,98E+01
	Mean	**7,84E+00**	**3,57E-01**	**1,96E+01**	**1,33E+01**	**1,66E+01**
	s.d.	**2,28E+01**	**1,77E-01**	**4,08E+00**	**6,92E+00**	**8,29E+00**
$ACO_{\mathbb{R}}$	Best	7,02E-03	3,20E-02	2,00E+01	9,95E-01	2,98E+00
	Worst	4,19E+00	3,47E-01	2,05E+01	1,59E+01	3,78E+01
	Mean	**8,31E-01**	**1,48E-01**	**2,04E+01**	**6,61E+00**	**1,22E+01**
	s.d.	**1,51E+00**	**8,61E-02**	**1,10E-01**	**2,94E+00**	**7,90E+00**

TABLA III

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F11$ A $F15$ CON $D = 10$

		F11	F12	F13	F14	F15
$ACO_{\mathbb{R}}^{sw}$	Best	2,25E+00	5,91E-05	4,78E-01	2,41E+00	7,52E+01
	Worst	8,03E+00	2,28E+03	2,15E+00	3,87E+00	3,35E+02
	Mean	**5,54E+00**	**2,69E+02**	**1,12E+00**	**3,24E+00**	**2,05E+02**
	s.d.	**1,47E+00**	**5,80E+02**	**4,33E-01**	**3,48E-01**	**7,28E+01**
$ACO_{\mathbb{R}}^{simplex}$	Best	2,71E+00	1,34E-03	4,06E-01	2,43E+00	6,32E+01
	Worst	7,90E+00	1,16E+04	2,77E+00	3,75E+00	4,32E+02
	Mean	**5,22E+00**	**9,86E+02**	**1,11E+00**	**3,29E+00**	**2,12E+02**
	s.d.	**1,34E+00**	**2,40E+03**	**5,39E-01**	**3,49E-01**	**9,54E+01**
$ACO_{\mathbb{R}}$	Best	1,04E+00	2,94E-08	3,41E-01	2,37E+00	1,19E+02
	Worst	5,73E+00	1,69E+03	3,09E+00	4,01E+00	4,29E+02
	Mean	**3,55E+00**	**1,77E+02**	**1,10E+00**	**3,15E+00**	**2,65E+02**
	s.d.	**1,41E+00**	**4,60E+02**	**6,17E-01**	**3,91E-01**	**1,19E+02**

TABLA IV

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F16$ A $F20$ CON $D = 10$

		F16	F17	F18	F19	F20
$ACO_{\mathbb{R}}^{sw}$	Best	1,02E+02	9,57E+01	3,00E+02	3,00E+02	3,00E+02
	Worst	1,83E+02	1,83E+02	8,00E+02	8,00E+02	8,00E+02
	Mean	**1,29E+02**	**1,39E+02**	**4,45E+02**	**4,42E+02**	**4,45E+02**
	s.d.	**2,09E+01**	**2,32E+01**	**1,54E+02**	**1,96E+02**	**1,75E+02**
$ACO_{\mathbb{R}}^{simplex}$	Best	9,27E+01	1,09E+02	3,00E+02	3,00E+02	3,00E+02
	Worst	1,81E+02	2,61E+02	1,02E+03	1,04E+03	1,01E+03
	Mean	**1,25E+02**	**1,41E+02**	**6,32E+02**	**7,22E+02**	**7,24E+02**
	s.d.	**2,38E+01**	**3,84E+01**	**2,51E+02**	**2,36E+02**	**2,37E+02**
$ACO_{\mathbb{R}}$	Best	9,76E+01	1,01E+02	3,00E+02	3,00E+02	3,00E+02
	Worst	1,49E+02	1,87E+02	9,53E+02	9,96E+02	1,00E+03
	Mean	**1,19E+02**	**1,32E+02**	**7,47E+02**	**8,53E+02**	**8,04E+02**
	s.d.	**1,49E+01**	**2,24E+01**	**1,93E+02**	**1,44E+02**	**2,01E+02**

de los 22 casos de prueba. Considerando los resultados del $ACO_{\mathbb{R}}^{simplex}$ y del $ACO_{\mathbb{R}}^{sw}$ en conjunto, los mejores valores de promedia de error son obtenido para 19 de los 22 casos. Estos valores indican que la BL ha mejorado las soluciones de la $ACO_{\mathbb{R}}$ de manera eficaz.

D. Comparativa con los modelos G-CMA-ES, DE, y K-PCX

La Tabla X compara el promedio del error entre las tres variantes del $ACO_{\mathbb{R}}$ y los métodos G-CMA-ES, DE, y K-PCX. Según esta tabla, $ACO_{\mathbb{R}}^{sw}$, $ACO_{\mathbb{R}}^{simplex}$, y$ACO_{\mathbb{R}}$ no son peores que cualquiera de los métodos de referencia para 7, 5, y 4 de las 20 funciones de prueba, respectivamente. Si restringimos nuestro análisis a las funciones de prueba $F15$ a $F25$ podemos observar que:

- $ACO_{\mathbb{R}}^{sw}$ tiene un error promedio similar a G-CMA-ES, DE, y K-PCX en 6, 11 y 13 de los 22 casos, respectivamente.
- $ACO_{\mathbb{R}}^{sw}$ tiene un error promedio similar a todos los métodos (G-CMA-ES, DE, y K-PCX) en 6 de los 22 casos.
- $ACO_{\mathbb{R}}^{simplex}$ tiene un error promedio similar a K-

TABLA V

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F21$ A $F25$ CON $D = 10$

		F21	F22	F23	F24	F25
$ACO_{\mathbb{R}}^{sw}$	Best	2,00E+02	7,26E+02	5,59E+02	2,00E+02	3,97E+02
	Worst	5,00E+02	8,08E+02	5,59E+02	2,00E+02	5,00E+02
	Mean	**4,66E+02**	**7,63E+02**	**5,59E+02**	**2,00E+02**	**4,12E+02**
	s.d.	**9,42E+01**	**1,82E+01**	**3,48E-13**	**0,00E+00**	**1,89E+01**
$ACO_{\mathbb{R}}^{simplex}$	Best	3,00E+02	7,44E+02	5,48E+02	2,00E+02	3,94E+02
	Worst	1,17E+03	8,00E+02	1,15E+03	5,00E+02	4,14E+02
	Mean	**6,77E+02**	**7,64E+02**	**6,33E+02**	**2,12E+02**	**4,08E+02**
	s.d.	**3,01E+02**	**1,18E+01**	**1,75E+02**	**6,00E+01**	**4,32E+00**
$ACO_{\mathbb{R}}$	Best	3,00E+02	7,38E+02	5,59E+02	2,00E+02	3,96E+02
	Worst	1,16E+03	8,73E+02	1,08E+03	9,00E+02	4,19E+02
	Mean	**6,53E+02**	**7,65E+02**	**6,81E+02**	**2,52E+02**	**4,06E+02**
	s.d.	**2,84E+02**	**2,70E+01**	**1,81E+02**	**1,58E+02**	**6,36E+00**

TABLA VI

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F6$ A $F10$ CON $D = 30$

		F6	F7	F8	F9	F10
$ACO_{\mathbb{R}}^{sw}$	Best	2,61E-03	2,84E-13	2,09E+01	3,18E+01	4,88E+01
	Worst	9,31E+01	4,43E-02	2,11E+01	2,62E+02	2,56E+02
	Mean	**1,09E+01**	**2,00E-02**	**2,10E+01**	**1,86E+02**	**1,44E+02**
	s.d.	**1,82E+01**	**1,17E-02**	**4,70E-02**	**6,91E+01**	**6,84E+01**
$ACO_{\mathbb{R}}^{simplex}$	Best	7,29E-03	2,84E-13	2,08E+01	4,38E+01	4,78E+01
	Worst	9,35E+01	6,89E-02	2,10E+01	2,31E+02	2,27E+02
	Mean	**1,31E+01**	**2,41E-02**	**2,10E+01**	**9,60E+01**	**1,64E+02**
	s.d.	**2,42E+01**	**1,73E-02**	**5,92E-02**	**6,12E+01**	**5,99E+01**
$ACO_{\mathbb{R}}$	Best	2,06E-04	2,56E-13	2,00E+01	2,49E+01	6,07E+01
	Worst	1,85E+01	5,16E-02	2,10E+01	9,05E+01	1,27E+02
	Mean	**6,96E+00**	**2,28E-02**	**2,07E+01**	**5,28E+01**	**8,20E+01**
	s.d.	**6,36E+00**	**1,49E-02**	**3,95E-01**	**1,68E+01**	**1,41E+01**

TABLA VII

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F11$ A $F15$ CON $D = 30$

		F11	F12	F13	F14	F15
$ACO_{\mathbb{R}}^{sw}$	Best	1,80E+01	4,47E+02	4,29E+00	1,20E+01	4,00E+02
	Worst	3,46E+01	1,12E+05	1,77E+01	1,35E+01	4,04E+02
	Mean	**2,53E+01**	**2,00E+04**	**7,64E+00**	**1,28E+01**	**4,01E+02**
	s.d.	**4,62E+00**	**2,79E+04**	**3,07E+00**	**3,93E-01**	**1,34E+00**
$ACO_{\mathbb{R}}^{simplex}$	Best	1,15E+01	3,31E+02	3,30E+00	1,17E+01	1,12E+02
	Worst	3,11E+01	2,87E+04	9,89E+00	1,36E+01	5,03E+02
	Mean	**2,01E+01**	**7,97E+03**	**6,12E+00**	**1,29E+01**	**3,04E+02**
	s.d.	**4,60E+00**	**8,11E+03**	**2,08E+00**	**3,75E-01**	**8,26E+01**
$ACO_{\mathbb{R}}$	Best	9,50E+00	2,22E+02	2,50E+00	1,08E+01	1,45E+02
	Worst	3,25E+01	2,25E+04	1,68E+01	1,35E+01	5,07E+02
	Mean	**2,01E+01**	**5,86E+03**	**6,91E+00**	**1,27E+01**	**3,10E+02**
	s.d.	**4,44E+00**	**5,97E+03**	**3,45E+00**	**5,43E-01**	**1,04E+02**

TABLA VIII

ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F16$ A $F20$ CON $D = 30$

		F16	F17	F18	F19	F20
$ACO_{\mathbb{R}}^{sw}$	Best	5,25E+01	6,85E+01	9,05E+02	9,06E+02	9,06E+02
	Worst	1,62E+02	2,21E+02	9,19E+02	9,22E+02	9,18E+02
	Mean	**1,06E+02**	**1,26E+02**	**9,09E+02**	**9,12E+02**	**9,11E+02**
	s.d.	**3,03E+01**	**4,09E+01**	**3,42E+00**	**3,92E+00**	**3,03E+00**
$ACO_{\mathbb{R}}^{simplex}$	Best	6,42E+01	6,64E+01	8,00E+02	9,06E+02	9,05E+02
	Worst	2,29E+02	3,10E+02	9,16E+02	9,20E+02	9,23E+02
	Mean	**1,18E+02**	**1,55E+02**	**9,07E+02**	**9,11E+02**	**9,12E+02**
	s.d.	**3,48E+01**	**7,06E+01**	**2,24E+01**	**3,65E+00**	**4,88E+00**
$ACO_{\mathbb{R}}$	Best	7,64E+01	8,04E+01	9,06E+02	9,07E+02	9,06E+02
	Worst	2,62E+02	4,10E+02	9,21E+02	9,18E+02	9,18E+02
	Mean	**1,17E+02**	**1,36E+02**	**9,13E+02**	**9,11E+02**	**9,11E+02**
	s.d.	**4,46E+01**	**8,60E+01**	**3,92E+00**	**3,08E+00**	**3,24E+00**

TABLA IX
ERRORES: MEJOR, PEOR, PROMEDIO Y DESVIACIÓN ESTÁNDAR PARA LAS FUNCIONES $F20$ A $F25$ CON $D = 30$

		F21	F22	F23	F24	F25
$ACO_{\mathbb{R}}^{sw}$	Best	5,00E+02	8,97E+02	5,34E+02	2,00E+02	2,11E+02
	Worst	5,00E+02	9,98E+02	5,34E+02	2,00E+02	2,16E+02
	Mean	**5,00E+02**	**9,36E+02**	**5,34E+02**	**2,00E+02**	**2,13E+02**
	s.d.	**0,00E+00**	**2,43E+01**	**5,00E-04**	**0,00E+00**	**9,51E-01**
$ACO_{\mathbb{R}}^{simplex}$	Best	5,00E+02	8,82E+02	5,34E+02	2,00E+02	2,12E+02
	Worst	8,18E+02	9,80E+02	9,58E+02	2,00E+02	2,15E+02
	Mean	**5,13E+02**	**9,40E+02**	**5,84E+02**	**2,00E+02**	**2,13E+02**
	s.d.	**6,36E+01**	**2,61E+01**	**1,37E+02**	**0,00E+00**	**8,47E-01**
$ACO_{\mathbb{R}}$	Best	5,00E+02	8,61E+02	5,34E+02	2,00E+02	2,12E+02
	Worst	5,00E+02	1,03E+03	5,44E+02	2,00E+02	2,14E+02
	Mean	**5,00E+02**	**9,36E+02**	**5,35E+02**	**2,00E+02**	**2,13E+02**
	s.d.	**0,00E+00**	**3,93E+01**	**2,02E+00**	**0,00E+00**	**6,93E-01**

PCX en 13 de los 22 casos

- $ACO_{\mathbb{R}}$ tiene un error promedio similar a K-PCX en 12 de los 22 casos.

Algunas de las conclusiones indican que la introducción de las búsquedas locales, como el caso del operador SW, es un valor añadido ya que en estos casos, el $ACO_{\mathbb{R}}^{sw}$ obtiene mejores resultados que el $ACO_{\mathbb{R}}$ simple.

VI. Conclusiones

En este trabajo hemos realizado un análisis de la utilización del $ACO_{\mathbb{R}}$ simple y combinado con los operadores de BL SW ($ACO_{\mathbb{R}}^{sw}$) y Simplex ($ACO_{\mathbb{R}}^{simplex}$). Como punto de referencia hemos considerado una serie de funciones con diferentes características que incluyen varios tipos de mínimos, dominios y dimensiones.

En primer lugar hemos comparado nuestros métodos entre sí, observando que el uso de los operadores de BL mejora los resultados obtenidos, principalmente cuando aumenta la dificultad de las funciones de prueba. Además, estos resultados se ha llevado a cabo un estudio comparativo con algunos de los métodos de referencia del CEC'05. En este caso, los resultados son positivos ya que se mejoran los errores de algunas de las funciones más difíciles, como el caso de las funciones $F15$ y $F21$ con $D = 10$ y $F17$ con $D = 30$. Además fueran obtenidos los mismos resultados para los casos de $F21$ con $D = 30$ y $F24$ con $D = 10$ y 30.

Como trabajo futuro derivado de este estudio podemos citar una mejor sintonización de los parámetros y la inclusión de otras búsquedas locales y su posible combinación en función de las características de los problemas.

Agradecimientos

Este trabajo ha sido subvencionado en parte por el proyecto TIN2007-66523 del Gobierno Español. Por otra parte, Christian Blum reconoce el apoyo del programa *Ramón y Cajal* del Ministerio Español de la Ciencia y Innovación.

Referencias

[1] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.

[2] B. Bilchev and I. C. Parmee, "The ant colony metaphor for searching continuous design spaces," in *Proceedings of the AISB Workshop on Evolutionary Computation*, 1995, vol. 993 of *Lecture Notes in Computer Science*, pp. 25–39.

[3] N. Monmarché, G. Venturini, and M. Slimane, "On how pachycondyla apicalis ants suggest a new search algorithm," *Future Generation Computer Systems*, vol. 16, pp. 937–946, 2000.

[4] J. Dréo and P. Siarry, "A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions," in *Proceedings of ANTS 2002 – From Ant Colonies to Artificial Ants: Third International Workshop on Ant Algorithms*, M. Dorigo, G. Di Caro, and M. Sampels, Eds. 2002, vol. 2463 of *Lecture Notes in Computer Science*, pp. 216–221, Springer Verlag, Berlin, Germany.

[5] P. Korosec and J. Silc, "The differential ant-stigmergy algorithm for large scale real-parameter optimization," in *Proceedings of ANTS 2008 – 6th International Conference on Ant Colony Optimization and Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, and A. Winfield, Eds. 2008, Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany.

[6] K. Socha, "Extended ACO for continuous and mixed-variable optimization," in *Proceedings of ANTS 2004 – Fourth International Workshop on Ant Algorithms and Swarm Intelligence*, M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, Eds. 2004, Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany.

[7] K. Socha and M. Dorigo, "Ant Colony Optimization for Continuous Domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[8] K. Socha and C. Blum, "An ant colony optimization algorithm for continuous optimization: Application to feed-forward neural network training," *Neural Computing & Applications*, vol. 16, no. 3, pp. 235–248, 2007.

[9] T. Stützle and H. H. Hoos, "$\mathcal{MAX}$-$\mathcal{MIN}$ Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[10] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 1958.

[11] F.J. Solis and R.J. Wets, "Minimization by random search techniques," *Mathematical Operations Research*, vol. 6, pp. 19–30, 1981.

[12] J.Ñelder and R. Mead, "A simplex method for functions minimizations," *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[13] Pablo Moscato, *New ideas in optimization*, chapter Memetic algorithms: a short introduction, pp. 219–234, McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.

[14] D. Molina, F. Herrera, and M. Lozano, "Adaptive local

TABLA X

Comparativa entre las variantes del $ACO_{\mathbb{R}}$ y los métodos G-CMA-ES, DE, y K-PCX (G: error promedio de la variante del $ACO_{\mathbb{R}}$ no superior a la de G-CMA-ES; D: error promedio de la variante del $ACO_{\mathbb{R}}$ no superior a la de DE; K: error promedio de la variante del $ACO_{\mathbb{R}}$ no superior a la del K-PCX))

$D = 10$	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
$ACO_{\mathbb{R}}^{sw}$						K			D	GDK
$ACO_{\mathbb{R}}^{simplex}$			GDK			K			D	GDK
$ACO_{\mathbb{R}}$		K	D		D	K			D	K
$D = 30$										
$ACO_{\mathbb{R}}^{sw}$	D		D			DK	G	K	GDK	K
$ACO_{\mathbb{R}}^{simplex}$	D		D			DK	G	K	GDK	DK
$ACO_{\mathbb{R}}$	D		D		D	DK	G	K	GDK	DK

$D = 10$	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25
$ACO_{\mathbb{R}}^{sw}$			K	K	DK	GDK		DK	GDK	D
$ACO_{\mathbb{R}}^{simplex}$			K	K	K	K		K	K	D
$ACO_{\mathbb{R}}$			K		K	K		K	K	D
$D = 30$										
$ACO_{\mathbb{R}}^{sw}$	D	GDK				GDK	K	K	GDK	D
$ACO_{\mathbb{R}}^{simplex}$	D	GDK				K	K	K	GDK	D
$ACO_{\mathbb{R}}$	D	GDK				GDK	K	K	GDK	D

search parameters for real-coded memetic algorithms," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 2005, vol. 1.

[15] P.N. 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.