Comportamiento de un Algoritmo Evolutivo Flexible para problemas de optimización continua

Silvia Alonso[1], Juan I. Jiménez[1], Himar Carmona[1], Blas Galván[1], Gabriel Winter[1], Begoña González[1]

Resumen--**Este trabajo describe una versión de un Algoritmo Evolutivo Flexible (AEF) diseñada con el fin de ser una nueva herramienta para resolver problemas de optimización del mundo real. En su desarrollo se han tenido en cuenta las lecciones aprendidas por otros autores en esta área. De esta forma, en vez de usar un único operador de cruce y/o mutación, el AEF cuenta con 60 operadores de distinta naturaleza. Esta implementación posee mecanismos de toma de decisiones basados en reglas que usan medidas fenotípicas de convergencia con el objetivo de prevenir y corregir situaciones de estancamiento, muestreando de nuevo las soluciones si es necesario. Si esto ocurre, en primer lugar se aplica un operador de cruce de un punto, después se cambian al azar los parámetros que controlan el muestreo, y finalmente se lanza un subconjunto de operadores de cruce para explorar el espacio de soluciones. Otra importante característica del algoritmo es que hace evolucionar individuos con un código genético extendido, que contiene información tanto sobre las variables, como sobre los operadores de muestreo y los parámetros que controlan el muestreo. Esta estructura da dos importantes ventajas. La primera es que el AEF ajusta sus parámetros *on the fly* ya que forman parte de los individuos que evolucionan. La segunda es que el método posee menos parámetros que otras implementaciones, siendo necesario definir sólo el tamaño de la población, el número de individuos elitistas y las evaluaciones a realizarse. Todas estas características son un paso eficiente hacia implementaciones sin parámetros en el área de la Computación Evolutiva. En este trabajo se presentan los resultados de aplicar el AEF al conjunto de funciones de prueba propuesto por los organizadores de la Sesión Especial en Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua del MAEB'09.**

Palabras clave—**Algoritmos Evolutivos, adaptación y auto-adaptación de operadores y parámetros genéticos.**

I. INTRODUCCIÓN

Los Algoritmos Evolutivos (AEs) son mecanismos versátiles capaces de abordar una gran variedad de problemas de optimización del mundo real. Aunque poseen muchas ventajas, como la capacidad de auto-adaptar la búsqueda de soluciones óptimas durante el proceso, también existen importantes desventajas en su aplicación. Entre estas últimas se debe señalar que los AEs dependen del problema al que se apliquen, y como consecuencia, los parámetros usados para su implementación deben ser ajustados para cada caso. Dado que esto representa un severo inconveniente, algunos investigadores usan parámetros fijos obtenidos en estudios exhaustivos realizados por otros autores [1] en vez de tratar de ajustar dichos parámetros en cada ocasión. Sin embargo, ya que esta cuestión es uno de los factores que los investigadores señalan como de mayor influencia en los resultados ([2],[3]), se han realizado múltiples estudios para adaptar los parámetros y/o los operadores que participan en la búsqueda de soluciones. Mientras algunos estudios se centran en adaptación de tasas de mutación y/o cruce [4], otros analizan valores de aptitud [5] o cualquier otro aspecto de la optimización [6]. Otras investigaciones alternativas consideran preferibles las implementaciones sin parámetros, ([7],[8]) a pesar de que en la mayoría de los casos son inviables debido al elevado coste computacional que implican. Un paso más sería considerar la posibilidad de que lo que se adapte sea la propia estructura del algoritmo dependiendo de las circunstancias de la optimización [9]. En este sentido, el uso e implementación de algún tipo de *memoria* adquiere una importancia primordial a la hora de guiar el proceso de optimización [10]. Algunos autores han realizado intentos exitosos en este campo, usando, por ejemplo, agentes [11]. En suma, se ha demostrado que es útil que las implementaciones sean flexibles, así como usar pocos parámetros o hacerlos evolucionar *on the fly*. Asimismo, el uso de memoria para dirigir la búsqueda es también esencial para obtener mejores resultados en los procesos de optimización.

El comienzo de este trabajo estará enfocado a describir un Algoritmo Evolutivo, denominado Algoritmo de Evolución Flexible (AEF), diseñado para aprovechar las características positivas enumeradas anteriormente, todas ellas de probada validez para mejorar procesos de optimización. Seguidamente, se incluirá un ejemplo de cómo actúa dicho algoritmo en un caso sencillo para ilustrar su comportamiento. A continuación, se comentarán los resultados obtenidos al aplicar esta implementación al conjunto de funciones de prueba propuestas por el MAEB'09. Para finalizar, se analizarán estos resultados y se establecerán unas conclusiones.

[1] Edificio Central del Parque Científico y Tecnológico, 2ª planta. División de Computación Evolutiva y Aplicaciones (CEANI), Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria : {salonso, ifranquiz, hcarmona,bgalvan,gwinter,bgonzalez}@iusiani.ulpgc.es

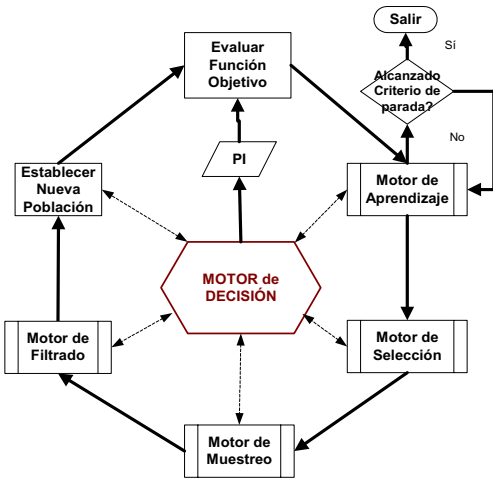
II. UNA INTRODUCIÓN AL AEF

Como ya se ha comentado en la sección precedente, varias son las lecciones aprendidas en este campo que pueden ser utilizadas para mejorar los procesos de optimización. Es por esta razón por lo que se ha desarrollado un Algoritmo Evolutivo que aproveche las ventajas de esas características. Versiones previas de esta implementación, llamada AEF, pueden encontrarse en trabajos previos ([12],[13],[14]). El método presenta varias características distintivas que serán descritas a continuación.

A. *La estructura general del AEF*

La estructura del AEF difiere en gran medida de otros AEs, como se puede ver en el diagrama incluido en la Fig. 1. El algoritmo ha sido dividido en funciones, llamadas *motores*, diseñadas para agrupar a las distintas acciones que van a ocurrir durante la optimización. Existe una función central denominada *Motor de Decisión* que se ocupa de gobernar todos los procesos y cuyas atribuciones se detallarán más adelante. Al comenzar la optimización, la población inicial (PI) de individuos candidatos a evolucionar y ser evaluados mediante la función de aptitud será establecida al azar. A continuación, otra función llamada *Motor de Aprendizaje* calculará medidas estadísticas, por ejemplo, de convergencia, que se almacenarán para ser usados después. El siguiente paso es pasar a través de un *Motor de Selección,* actualmente constituido por un mecanismo de selección por torneo (2:1).

Fig. 1. Esquema general del AEF



A continuación, la subrutina más importante después de la del *Motor de Decisión* entra en acción: el *Motor de Muestreo*. Ésta está formada por un conjunto de 60 operadores diferentes tanto de mutación como de cruce. Algunos de los operadores incluidos han sido propuestos por otros autores o son variaciones de dichos operadores, tal y como se ha descrito en [14]. Entre los operadores de cruce considerados se pueden citar el Geométrico y el Aritmético [15], el cruce basado en Conectores Lógicos [17] o la mayoría de los incluidos en estudios como el realizado en [16]. Ejemplos de los operadores de mutación usados son el Gaussiano [17], el no uniforme [18] o el operador de mutación del BGA (*Breeder Genetic Algorithm*[19]).

La manera en que el *Motor de Decisión* gobierna el *Motor de Muestreo* es la principal causa de la alta flexibilidad del AEF y su capacidad de explotar y explorar a la vez el espacio de búsqueda de las soluciones. Existen dos parámetros de control que deciden cómo muestrear las soluciones, que pueden tener un valor entre 0 y 1. El primero rige cómo muestrear los individuos, es decir, si se muestrea una de sus variables escogida al azar o todas. El segundo decide a continuación cuál de los 60 operadores de muestreo existentes se usará, si el último utilizado u otro cualquiera. Esos dos parámetros se ajustan durante el proceso y no necesitan ser predeterminados. En la última implementación un cruce de un punto ha sido incluido para generar más diversidad en las muestras antes de entrar en el *Motor de Muestreo* propiamente dicho.

Al final del proceso, una futura versión del AEF eliminará las soluciones repetidas fenotípicamente. No obstante, en esta versión no se cuenta con dicho filtrado. Para terminar, se repite el proceso hasta que se cumple un determinado criterio de parada, generalmente alcanzar un número determinado de evaluaciones de la función objetivo.

B. *Definiendo algunos conceptos básicos*

En primer lugar, la flexibilidad del AEF es debida en gran medida a su *Estructura Dinámica de Operadores (EDO)* como ya se ha explicado anteriormente. En la mayoría de AEs existe un único operador de cruce o de mutación. Incluir más de un operador de cada tipo, sumado a poder utilizarlos en cualquier momento del proceso de optimización otorga al AEF una estructura única y una manera equilibrada de buscar nuevas soluciones. De esta manera, el par formado por los motores de decisión y de muestreo promueven el equilibrio entre la exploración y la explotación del espacio de búsqueda.

En segundo lugar, el AEF cuenta con un *Código Genético Extendido (CGE),* lo que significa que se hace evolucionar individuos compuestos no sólo de información referente a las variables en juego, sino también información sobre el propio proceso de optimización. El objetivo final es incorporar estadísticas, medidas y cualquier característica que pueda ser usada con el fin de guiar el proceso de optimización. Por el momento, la estructura de la población es $P(\vec{W}) = P(\vec{x}, f, \vec{y}, \vec{z})$, donde P es la

población de $\vec{W}$ individuos, $\vec{x}$ es el vector que contiene las variables, f es el valor de función objetivo, $\vec{y}$ el vector que contiene los métodos de muestreo, y $\vec{z}$ el que contiene los parámetros de control.

El punto clave de este CGE no es el cromosoma en sí mismo, sino el uso que se puede hacer de la información que guarda. En este sentido, es esencial poseer *Mecanismos de Control Central (MCC)* para dirigir la búsqueda, centrados en el *Motor de Decisión*. Actualmente, los MCC están basados en reglas, a pesar de que en versiones anteriores del AEF se han probado otros mecanismos de control central y de lógica difusa. Los mecanismos de toma de decisiones existentes gobiernan las acciones a llevarse a cabo en caso de que el proceso de optimización se estanque. Esta eventualidad es medida por medio de una tasa de convergencia tal como la propuesta en [20].

Si se produce estancamiento, los MCC desencadenan tres mecanismos para tratar de resolver el problema. Primero, se aplica de nuevo el operador de cruce de un punto ya ejecutado antes de entrar en el *Motor de Muestreo* con el fin de generar diversidad en la muestra. Esta medida se toma con el objeto de crear individuos diferentes de los ya obtenidos anteriormente, ya que el AEF trabaja usando elitismo, lo cual preserva a los mejores individuos que son continuamente cruzados con el resto de individuos y genera individuos similares entre sí. En consecuencia, la tendencia natural es generar súper-individuos que producen convergencia prematura a soluciones no deseadas. El paso siguiente es asignar un valor al azar a los parámetros de control. Durante el proceso, éstos tienden a parar su progreso si no se está convergiendo y, puesto que son esenciales para decidir cómo muestrear las soluciones es muy importante prevenir su estancamiento. Por último, y tomando ventaja de la existencia de grupos de operadores de distinta naturaleza, se ejecuta un *Motor de Muestreo* truncado denominado *Motor de Muestreo lejano*. Este motor truncado incluye principalmente operadores de cruce, así como algunos de mutación, escogidos debido a que su naturaleza es tal que ayudan la exploración lejos de las actuales soluciones.

Es importante señalar que el AEF trabaja con muy pocos parámetros, ya que la mayoría han sido incorporados a la estructura de los individuos y evolucionan durante el proceso de optimización. Esto significa que los parámetros de control descritos en el apartado anterior ya están implementados como parte de la soluciones y no necesitan ser ajustados para cada problema específico, como ocurría en versiones anteriores del AEF. Los parámetros que quedan son el tamaño de la población (generalmente fijado en un valor de 100 individuos), el número de evaluaciones de la función objetivo y el elitismo usado, que es de un 15%.

II. EJEMPLO DE FUNCIONAMIENTO DEL AEF

Con el objetivo de ilustrar cómo funciona el AEF se explica en detalle un ejemplo simple usando una función genérica, lo que permite entender la estructura del algoritmo y la manera que tienen de interactuar los motores que lo componen.

Se considera el problema de minimizar una función $f(\vec{x})$ de cuatro variables cuyos valores están en el intervalo [0,10]. Se establece un tamaño de población de 20 individuos, con un 15% de elitismo, con lo cual sólo 3 individuos pertenecerán a dicho grupo. Para esta función el óptimo es conocido e igual a cero. El criterio de parada será alcanzar un número máximo predeterminado de evaluaciones, fijado en 20000.

Para comenzar el proceso de optimización, se genera una primera población al azar obteniendo 20 individuos de 4 variables cada uno con valores entre 0 y 10. Un individuo tendrá la forma I= {$\vec{x}$, f, $\vec{y}$, $\vec{z}$}, siendo $\vec{x}$ el vector de las variables del individuo, f el valor de función objetivo correspondiente, $\vec{y}$ el vector que guarda los métodos de muestreo usados sobre cada variable del individuo y $\vec{z}$ el vector que contiene las dos variables de control: *control1* (usado para seleccionar las variables utilizadas al muestrear el individuo) y *control2* (usado para seleccionar qué operadores de muestreo se van emplear). Particularmente, los cinco primeros individuos pueden ser (los tres primeros son la élite):

$I^1 = \{x_1^1, x_2^1, x_3^1, x_4^1, f^1, y_1^1, y_2^1, y_3^1, y_4^1, c_1^1, c_2^1\} = \{3.3, 9.2, 6.3, 1.9, 0.5, 7, 32, 13, 35, 0.4, 0.6\}$

$I^2 = \{x_1^2, x_2^2, x_3^2, x_4^2, f^2, y_1^2, y_2^2, y_3^2, y_4^2, c_1^2, c_2^2\} = \{4.9, 0.3, 0.7, 9.6, 0.6, 2, 18, 40, 26, 0.1, 0.4\}$

$I^3 = \{x_1^3, x_2^3, x_3^3, x_4^3, f^3, y_1^3, y_2^3, y_3^3, y_4^3, c_1^3, c_2^3\} = \{1.0, 4.9, 2.2, 9.2, 0.7, 3, 56, 14, 24, 0.1, 0.2\}$

$I^4 = \{x_1^4, x_2^4, x_3^4, x_4^4, f^4, y_1^4, y_2^4, y_3^4, y_4^4, c_1^4, c_2^4\} = \{2.3, 8.9, 8.4, 0.7, 0.8, 45, 58, 60, 35, 0.7, 0.0\}$

$I^5 = \{x_1^5, x_2^5, x_3^5, x_4^5, f^5, y_1^5, y_2^5, y_3^5, y_4^5, c_1^5, c_2^5\} = \{2.7, 3.4, 5.3, 8.9, 0.9, 9, 36, 34, 3, 0.8, 0.0\}$

En ellos se presentan, por orden, los cuatro valores de las variables, el valor de la función objetivo del individuo, seguidos por cuatro valores correspondientes a los muestreadores que podrán ser utilizados posteriormente al muestrear cada variable, y por último los valores de *control1* y *control2*. Todos estos valores se establecen inicialmente al azar.

A continuación la función es evaluada y los valores son ordenados de mejor a peor. Los individuos especificados ya se han puesto por orden, siendo el 1º el que tiene el menor valor de función objetivo. Seguidamente el *Motor de Aprendizaje* calcula las medidas estadísticas de convergencia que podrán ser utilizadas posteriormente para corregir la búsqueda.

Seguidamente se cambian las variables de los individuos de toda la población salvo la élite

partiendo de dos individuos de la élite usando el cruce de un punto. No se modifican en esta repoblación los valores de control ni los de los operadores de muestreo guardados en los individuos originales. Por ejemplo, partiendo de los individuos 1º y 3º especificados antes se hallan los individuos 4 y 5 de la siguiente forma: Se escoge un punto de cruce, como puede ser después de la primera variable, y se intercambian las variables entre ellos. De esta forma el individuo 4 tendrá como primera variable la primera variable del 1º y las tres últimas del individuo 3, mientras que el individuo 5 tendrá la primera variable del tercer individuo y las tres últimas del 1º. El resto de valores son los originales del individuo 4. Así, los individuos 4 y 5 quedan de la forma siguiente, donde se han marcado las variables procedentes del individuo 1 con una línea continua, y las procedentes del individuo 3 con líneas discontinuas:

I^4= {$x_1^4, x_2^4, x_3^4, x_4^4, f^4, y_1^4, y_2^4, y_3^4, y_4^4, c_1^4, c_2^4$}= {3,3,4,9,2.2,9.2,0.8,45,58,60,35,0.7,0.0}

I^5= {$x_1^5, x_2^5, x_3^5, x_4^5, f^5, y_1^5, y_2^5, y_3^5, y_4^5, c_1^5, c_2^5$}= {1.0,9.2,6.3,1.9,0.9,9,36,34,3,0.8,0.0}

El resto de individuos de la población son modificados de la misma manera: se sustituyen sus variables por las pertenecientes a dos individuos de la élite tomados al azar intercambiadas según un punto de corte también escogido aleatoriamente.

Para continuar, se comprueban las medidas estadísticas para comprobar si hay estancamiento en esta nueva muestra y decidir con qué conjunto de muestreadores se va a muestrear, si con el motor completo original o con el truncado. Si hubiera estancamiento, se cambiarían los valores de control al azar, se volvería a repoblar y se especificaría que en el proceso de muestreo que va a comenzar se llame al *Motor de Muestreo* Lejano, todo ello con el fin de salir del estancamiento. En este ejemplo se considera que no hay estancamiento, por lo que cuando se llegue al muestreo se usará el *Motor de Muestreo* general.

Se procede seguidamente a tomar las decisiones que afectan al muestreo mediante el *Motor de Decisión*. Se ejecuta el *Motor de Selección*, escogiendo en primer lugar los individuos que podrán ser usados como padres (*selección de individuos)*. Cuatro son los individuos que el AEF selecciona: el mejor individuo (individuo 1), uno de un torneo 2:1 entre la élite, otro en el medio de la población (que podría ser, por ejemplo el 12) y uno entre los peores (como por ejemplo el 17). Entre los elitistas se hace un torneo 2:1 entre el individuo 2 y el 3 ganando dicho torneo por valor de función objetivo el individuo 2. Los individuos 12 y 17 se especifican a continuación:

I^{12}= {$x_1^{12}, x_2^{12}, x_3^{12}, x_4^{12}, f^{12}, y_1^{12}, y_2^{12}, y_3^{12}, y_4^{12}, c_1^{12}, c_2^{12}$}= {9.2,3.3,9.2,6.3,0.9,4,6,45,33,0.2,0.4}

I^{17}= {$x_1^{17}, x_2^{17}, x_3^{17}, x_4^{17}, f^{17}, y_1^{17}, y_2^{17}, y_3^{17}, y_4^{17}, c_1^{17}, c_2^{17}$}= {0.3,0.7,9.6,1.0,1.3,7,46,51,9,0.6,0.1}

Las cuatro primeras variables de dichos individuos proceden de la repoblación: en el caso del individuo 12, la primera variable procede del individuo 3 y el resto del individuo 2, mientras que para el individuo 17, las tres primeras variables proceden del individuo 2 y la última del individuo 3.

A continuación (y por medio de los MCC del *Motor de Decisión*) se escogen las variables que se van a muestrear (*selección de variables*). Los valores que pueden ser muestreados son tanto las cuatro primeras variables como cualquiera de los valores de control. En este caso, se muestrean dos variables, que coinciden con las dos primeras de los individuos. Esto implica que al escoger un operador de muestreo cualquiera las dos primeras variables del individuo muestreado se modificarán, siendo sustituidas por las dos primeras variables ó recombinación de las mismas de los padres utilizados por el operador de muestreo seleccionado. Para ilustrar esta operación se explica el proceso de muestreo de manera pormenorizada.

A continuación se procede a realizar el muestreo. En el AEF se muestrean todos los individuos de la población que no pertenecen a la élite, en este caso 17, realizándose el proceso en las dos variables seleccionadas anteriormente de cada uno de los individuos. Para cada variable seleccionada para ser muestreada de cada individuo se decide qué operador de muestreo se va a utilizar (*selección de operadores*), que puede ser el usado anteriormente (en este caso el generado al azar y guardado en el cromosoma al ser la primera iteración) ó uno cualquiera. Por ejemplo, para el primer individuo muestreado, que es el cuarto de la población y el primero no perteneciente a la élite, se selecciona que la primera variable, de valor 3,3, se muestreará con un operador cualquiera, que seleccionado al azar es el 18 y la segunda variable, de valor 4,9 se decidirá después con qué operador se muestreará. Seguidamente, se muestrea con el operador 18 la primera variable. El nuevo valor de cada variable muestreada se calcula partir del valor de las variables correspondientes del individuo ó individuos especificados por el operador usado. En este caso, el método escogido parte del valor de dos padres: el mejor individuo (x_i^{old1}) y el ganador del torneo 2:1, es decir, el individuo 2 (x_i^{old2}). Por lo tanto se sustituirá la primera variable del individuo 4 modificando la primera variable del individuo 2, que es 4,9, según la expresión siguiente:

Método 18	$x_i^{new} = x_i^{old2} + sign \cdot (x_i^{old1} - x_i^{old2}) \cdot U(0,1)$

En este caso se considera que *sign*, variable que devuelve un signo positivo ó uno negativo, es negativa, mientras que U(0,1), que es un número generado entre 0 y 1 es 0,2. Como $x_i^{old1} = x_1^{old1} = 3{,}3$ y $x_i^{old2} = x_1^{old2} = 4{,}9$, la variable 1 del individuo 4 se calcula la siguiente forma:

$x_1^{new4} = x_1^{old2} - (x_1^{old1} - x_1^{old2}) \cdot U(0,1) = 4{,}9 - (3{,}3 - 4{,}9) \cdot 0{,}2 = 5{,}22$

A continuación se pasa a la siguiente variable, en este caso la variable 2 del individuo 4, de valor 4,9, y se decide qué muestreador se usará sobre ella. Se obtiene que se utilice el mismo operador antes usado, esto es, el 18. Al calcularla, U(0,1) toma ahora un valor de 0,5, mientras *sign* será positivo. Como $x_i^{old1} = x_2^{old1} = 9{,}2$ y $x_i^{old2} = x_2^{old2} = 0{,}3$, la variable 2 del individuo 4 se calculará como sigue:

$$x_2^{new4} = x_2^{old2} + (x_2^{old1} - x_2^{old2}) \cdot U(0,1) = 0.3 + (9.1-0.3) \cdot 0{,}5 = 4{,}7.$$

Si se sustituyen las variables 1 y 2 del individuo muestreado por los nuevos valores queda

$$I^{new4} = \{x_1^4, x_2^4, x_3^4, x_4^4, y^4, z_1^4, z_2^4, z_3^4, z_4^4, c_1^4, c_2^4\} = \{5.2, 4.7, 2.2, 9.4, 0.8, 18, 18, 60, 35, 0.7, 0.0\}$$

Donde se han marcado con líneas discontinuas los cambios en el individuo, que son los valores de las dos primeras variables y el valor del operador que se ha usado para muestrearlas. Con línea continua se señala el valor de función objetivo para las variables antiguas que cambiará en cuanto se vuelva a evaluar la función con los nuevos valores de sus variables al comienzo de la próxima generación.

A continuación se pasaría al individuo siguiente a ser muestreado, el 5, se ve qué operador de muestreo se debe usar para la primera de las de las dos variables del individuo escogidas para ser variadas y se muestrea. Se realizarían los cálculos para la primera variable, y después se comprobaría si se usa de nuevo el mismo muestreador u otro diferente para la segunda variable y se pasaría a muestrear dicha variable. Se sigue repitiendo la selección de muestreadores hasta recorrer todos los individuos y se muestrean todas las variables seleccionadas para ser muestreadas, esto es, las dos primeras del resto de individuos hasta el 20.

Finalmente se evalúan los individuos sin incluir a los tres de la élite, y se ordenan todos, generándose una nueva élite Por último, se vuelve a empezar el proceso, repitiéndose hasta que se alcanza el criterio de parada.

III. RESULTADOS

Esta sección incluye los resultados de aplicar el AEF al conjunto de 20 funciones de prueba propuestas para la sesión especial del MAEB'09 *Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua.* Las siete primeras (F6-F12) son básicas, las dos siguientes son expandidas (F13 y F14) y las últimas 11 son híbridas, mezclas entre algunas de las anteriores y otras funciones. El conjunto de funciones de prueba escogido es parte del propuesto para la Sesión Especial en Optimización con parámetros reales del CEC'05, y se eliminan las cinco primeras funciones.

A continuación se describen los resultados obtenidos, que serán discutidos en la siguiente sección. Todos los experimentos fueron programados en C++ y ejecutados en el mismo ordenador, un Pentium(R) 4 3.40 GHz, 1 GB RAM, Windows XP (SP2).

Dichos resultados se incluyen en tablas que contienen los valores de las medias del error obtenidos para los 20 problemas propuestos. Para cada función de prueba y cada dimensión (D= 10 y 100000 evaluaciones de función objetivo, D= 30 y 300000 evaluaciones) se realizaron 25 ejecuciones. Las tablas muestran los valores de las medias del error (f(x)-f(x*), donde x* es el vector de parámetros óptimo y x es nuestra solución) al finalizar el número de evaluaciones programado. También se muestran las desviaciones estándar de ese valor medio, así como el mejor y el peor valor obtenido de las 25 ejecuciones. Se incluyen esos valores en tres momentos de la ejecución: para 1000 evaluaciones, para 10000 y para 100000. En el caso de D=30, también se incluyen para 300000 evaluaciones.

El criterio de parada era alcanzar un número máximo de evaluaciones, dado por el valor 10000×D (desde ahora CP1 o criterio de parada 1) o alcanzar un valor de error menor que 10^{-8} (CP2 o criterio de parada 2). En la mayoría de los casos el algoritmo paró debido al criterio de parada CP1, es decir, el máximo número de evaluaciones fue alcanzado antes de que el valor de error fuera menor que 10^{-8}. Aquellos casos en los que el criterio de parada fue el CP2 están marcados con un asterisco en las tablas correspondientes (TABLA I y TABLA II).

A continuación se incluyen ocho tablas. En cada tabla están los valores obtenidos para cinco funciones test. Cuatro tablas corresponden a la dimensión D=10 y el resto a la dimensión D=30.

TABLA I

VALORES DE ERROR: FUNCIONES 6-10 (D=10)

10D		*6*	*7*	*8*	*9*	*10*
1E3	Mejor	8,4546E+04	2,2218E+01	2,0451E+01	1,0530E+01	4,0319E+01
	Peor	8,5626E+07	1,9861E+02	2,0920E+01	2,8343E+01	8,0104E+01
	Media	**1,2097E+07**	**9,7683E+01**	**2,0739E+01**	**1,7208E+01**	**6,3350E+01**
	D.E.	**1,9570E+07**	**5,0349E+01**	**1,3156E-01**	**4,9524E+00**	**1,0524E+01**
1E4	Mejor	2,6501E+01	3,2288E-01	2,0047E+01	7,2747E-06	7,9598E+00
	Peor	8,0133E+03	5,8431E+00	2,0525E+01	9,9531E-01	4,3780E+01
	Media	**1,5821E+03**	**2,2117E+00**	**2,0272E+01**	**2,4013E-01**	**2,3438E+01**
	D.E.	**2,2280E+03**	**1,4578E+00**	**1,2371E-01**	**4,3300E-01**	**9,4602E+00**
1E5	Mejor	3,5622E-02	4,9269E-02	2,0020E+01	2,786E-09*	7,9597E+00
	Peor	5,9414E+03	5,8098E+00	2,0315E+01	9,945E-09*	4,3778E+01
	Media	**3,2161E+02**	**1,8384E+00**	**2,0143E+01**	**7,918E-09***	**2,3401E+01**
	D.E.	**1,1848E+03**	**1,4372E+00**	**9,3362E-02**	**2,143E-09***	**9,4744E+00**

TABLA II

VALORES DE ERROR: FUNCIONES 11-15 (D=10)

10D		*11*	*12*	*13*	*14*	*15*
1E3	Mejor	7,4671E+00	3,0509E+03	3,0000E+00	3,7249E+00	1,7298E+02
	Peor	1,2631E+01	3,1742E+04	1,8715E+01	4,3480E+00	6,3973E+02
	Media	**1,0084E+01**	**1,2051E+04**	**8,4983E+00**	**4,1308E+00**	**3,9323E+02**
	D.E.	**1,2441E+00**	**6,2278E+03**	**3,5697E+00**	**1,7007E-01**	**1,7099E+02**
1E4	Mejor	5,0114E+00	3,7195E+01	2,2790E-01	3,1930E+00	5,1720E-05
	Peor	9,6915E+00	9,7226E+03	2,8159E+00	4,0566E+00	4,5007E+02
	Media	**7,2483E+00**	**1,9536E+03**	**7,4663E-01**	**3,7377E+00**	**1,3624E+02**
	D.E.	**1,2529E+00**	**2,5043E+03**	**5,2482E-01**	**2,3794E-01**	**2,0285E+02**
1E5	Mejor	4,4540E+00	4,0976E-01	4,1831E-02	2,9699E+00	3,687E-09*
	Peor	9,3238E+00	1,7663E+03	5,1657E-01	3,9894E+00	4,3638E+02
	Media	**6,9486E+00**	**2,7096E+02**	**3,1859E-01**	**3,5277E+00**	**1,3465E+02**
	D.E.	**1,2750E+00**	**4,6296E+02**	**1,3469E-01**	**2,1418E-01**	**2,0046E+02**

El AEF ha sido incapaz de llegar al nivel de precisión pedido excepto para las funciones 9 y 15,

para D=10. En esas dos funciones el número medio de evaluaciones necesario para llegar al valor de error pedido fue menor que el límite, en este caso 100000 evaluaciones, y en consecuencia, el algoritmo ha parado por el CP2. En el caso de la función 9 se necesitaron sólo 17640 evaluaciones, mientras que para la función 15 se necesitaron 45799. En el caso de esta última función, se paró por el CP2 18 veces y 12 por el CP1, siendo la única función donde cualquiera de los dos criterios de parada podía ocurrir aproximadamente con igual probabilidad. En cualquier caso, nótese que el AEF alcanza un valor muy próximo al valor final ya desde las 1000 evaluaciones en la práctica totalidad de los casos.

TABLA III

VALORES DE ERROR: FUNCIONES 16-20 (D=10)

10D		*16*	*17*	*18*	*19*	*20*
1E3	Mejor	2,0382E+02	2,1686E+02	9,2280E+02	7,9234E+02	9,1279E+02
	Peor	3,6804E+02	4,5075E+02	1,1308E+03	1,1397E+03	1,1135E+03
	Media	**2,5042E+02**	**2,9509E+02**	**1,0478E+03**	**1,0215E+03**	**1,0566E+03**
	D.E.	**3,8707E+01**	**5,3193E+01**	**5,4668E+01**	**8,4797E+01**	**4,5231E+01**
1E4	Mejor	1,1442E+02	1,1832E+02	5,9188E+02	5,1744E+02	8,0026E+02
	Peor	2,1966E+02	2,5528E+02	1,0632E+03	1,0934E+03	1,0812E+03
	Media	**1,5839E+02**	**1,9095E+02**	**9,1594E+02**	**9,0443E+02**	**9,6143E+02**
	D.E.	**2,4200E+01**	**3,4455E+01**	**1,3496E+02**	**1,4479E+02**	**9,1943E+01**
1E5	Mejor	1,1430E+02	1,0790E+02	5,9125E+02	5,1679E+02	6,4656E+02
	Peor	2,1966E+02	2,1976E+02	1,0622E+03	1,0932E+03	1,0811E+03
	Media	**1,5808E+02**	**1,6066E+02**	**9,0891E+02**	**8,9982E+02**	**9,5086E+02**
	D.E.	**2,4083E+01**	**3,2962E+01**	**1,3428E+02**	**1,5088E+02**	**1,1023E+02**

TABLA IV

VALORES DE ERROR: FUNCIONES 21-25 (D=10)

10D		*21*	*22*	*23*	*24*	*25*
1E3	Mejor	8,6795E+02	8,1991E+02	1,0919E+03	5,3390E+02	6,3689E+02
	Peor	1,2975E+03	1,0840E+03	1,3111E+03	1,3109E+03	1,3034E+03
	Media	**1,2178E+03**	**9,4496E+02**	**1,2669E+03**	**9,3588E+02**	**1,1018E+03**
	D.E.	**1,0912E+02**	**7,8677E+01**	**4,8240E+01**	**2,3227E+02**	**1,5530E+02**
1E4	Mejor	5,0000E+02	7,8009E+02	5,5947E+02	2,0000E+02	2,0000E+02
	Peor	1,2597E+03	1,0041E+03	1,2922E+03	9,0000E+02	9,0000E+02
	Media	**1,0223E+03**	**8,6521E+02**	**1,1992E+03**	**2,8861E+02**	**2,5610E+02**
	D.E.	**2,4499E+02**	**7,3972E+01**	**1,4187E+02**	**1,7704E+02**	**1,9379E+02**
1E5	Mejor	5,0000E+02	7,7751E+02	5,5947E+02	2,0000E+02	2,0000E+02
	Peor	1,2594E+03	9,7881E+02	1,2769E+03	9,0000E+02	9,0000E+02
	Media	**1,0205E+03**	**8,4555E+02**	**1,1251E+03**	**2,8850E+02**	**2,5600E+02**
	D.E.	**2,4431E+02**	**6,4045E+01**	**2,0624E+02**	**1,7699E+02**	**1,9382E+02**

TABLA V

VALORES DE ERROR: FUNCIONES 6-10 (D=30)

30D		*6*	*7*	*8*	*9*	*10*
1E3	Mejor	3,2002E+09	1,7270E+03	2,1075E+01	2,5992E+02	4,0483E+02
	Peor	1,5202E+10	5,6119E+03	2,1340E+01	3,4611E+02	5,9481E+02
	Media	**9,4172E+09**	**2,7326E+03**	**2,1245E+01**	**3,0643E+02**	**4,9048E+02**
	D.E.	**3,1979E+09**	**9,0978E+02**	**7,7201E-02**	**2,3042E+01**	**4,2965E+01**
1E4	Mejor	8,1554E+06	3,9465E+01	2,0658E+01	5,0173E+01	2,0625E+02
	Peor	6,4648E+07	4,5422E+02	2,1076E+01	2,0543E+02	3,2925E+02
	Media	**2,7408E+07**	**1,9229E+02**	**2,0839E+01**	**1,1408E+02**	**2,8516E+02**
	D.E.	**1,6170E+07**	**8,6441E+01**	**1,0032E-01**	**3,5014E+01**	**4,0355E+01**
1E5	Mejor	1,3726E+02	1,4899E+00	2,0256E+01	1,2290E+01	1,2120E+02
	Peor	2,4790E+03	4,1339E+01	2,0690E+01	1,4123E+02	3,2184E+02
	Media	**5,1415E+02**	**1,1137E+01**	**2,0464E+01**	**4,2879E+01**	**2,1737E+02**
	D.E.	**5,3840E+02**	**1,0641E+01**	**1,0941E-01**	**3,3630E+01**	**5,5751E+01**
3E5	Mejor	5,1867E+01	1,0574E+00	2,0123E+01	3,4265E+00	1,0730E+02
	Peor	1,4199E+03	1,7126E+01	2,0512E+01	1,2577E+02	3,1734E+02
	Media	**2,5807E+02**	**3,5902E+00**	**2,0320E+01**	**2,4077E+01**	**1,8635E+02**
	D.E.	**3,0568E+02**	**4,4264E+00**	**1,1317E-01**	**3,3114E+01**	**5,3470E+01**

TABLA VI

VALORES DE ERROR: FUNCIONES 11-15 (D=30)

30D		*11*	*12*	*13*	*14*	*15*
1E3	Mejor	4,3459E+01	6,4205E+05	1,1870E+03	1,3852E+01	7,2531E+02
	Peor	4,8134E+01	1,2195E+06	7,3197E+04	1,4482E+01	1,0277E+03
	Media	**4,6145E+01**	**9,5497E+05**	**1,5277E+04**	**1,4194E+01**	**8,8349E+02**
	D.E.	**1,1838E+00**	**1,3637E+05**	**1,7241E+04**	**1,2353E-01**	**9,0124E+01**
1E4	Mejor	2,9828E+01	6,2742E+04	7,3373E+00	1,3043E+01	1,8459E+02
	Peor	4,1195E+01	2,3453E+05	3,8635E+01	1,3876E+01	5,9906E+02
	Media	**3,3987E+01**	**1,3522E+05**	**2,1632E+01**	**1,3485E+01**	**4,6290E+02**
	D.E.	**2,6126E+00**	**4,1631E+04**	**8,0757E+00**	**2,3712E-01**	**7,6688E+01**
1E5	Mejor	2,7384E+01	1,4931E+03	3,1568E+00	1,1941E+01	1,5143E-02
	Peor	4,0134E+01	1,0791E+05	1,6617E+01	1,3561E+01	5,0356E+02
	Media	**3,1486E+01**	**3,1652E+04**	**7,3701E+00**	**1,3057E+01**	**3,3100E+02**
	D.E.	**3,1594E+00**	**2,1738E+04**	**3,1977E+00**	**3,6929E-01**	**1,1256E+02**
3E5	Mejor	2,7366E+01	4,5697E+02	2,8788E+00	1,1936E+01	2,0005E+02
	Peor	4,0129E+01	5,9457E+04	1,6617E+01	1,3550E+01	5,0356E+02
	Media	**3,1433E+01**	**1,4819E+04**	**5,5672E+00**	**1,2908E+01**	**3,4055E+02**
	D.E.	**3,1502E+00**	**1,3042E+04**	**2,8015E+00**	**3,9249E-01**	**9,4916E+01**

TABLA VII

VALORES DE ERROR: FUNCIONES 16-20 (D=30)

30D		*16*	*17*	*18*	*19*	*20*
1E3	Mejor	4,7341E+02	4,9440E+02	1,0696E+03	1,1012E+03	1,1461E+03
	Peor	8,4573E+02	9,2175E+02	1,3014E+03	1,2331E+03	1,2930E+03
	Media	**6,2104E+02**	**7,3472E+02**	**1,1758E+03**	**1,1737E+03**	**1,1971E+03**
	D.E.	**9,4367E+01**	**9,7981E+01**	**4,7328E+01**	**3,4281E+01**	**3,3264E+01**
1E4	Mejor	2,1397E+02	3,0549E+02	9,3758E+02	8,5384E+02	8,8418E+02
	Peor	5,6661E+02	6,3320E+02	1,0222E+03	1,0667E+03	1,0568E+03
	Media	**3,3194E+02**	**4,1984E+02**	**9,8054E+02**	**9,7729E+02**	**9,9622E+02**
	D.E.	**1,0204E+02**	**9,8310E+01**	**2,1659E+01**	**3,9819E+01**	**3,4693E+01**
1E5	Mejor	1,0015E+02	2,0150E+02	9,4172E+02	8,0001E+02	8,0002E+02
	Peor	5,0241E+02	5,7612E+02	1,0124E+03	1,0572E+03	1,0362E+03
	Media	**2,6995E+02**	**3,3836E+02**	**9,6717E+02**	**9,4355E+02**	**9,7360E+02**
	D.E.	**1,2622E+02**	**1,1778E+02**	**2,1167E+01**	**6,5277E+01**	**4,5955E+01**
3E5	Mejor	1,0006E+02	1,6899E+02	9,4159E+02	8,0000E+02	8,0000E+02
	Peor	5,0225E+02	5,6013E+02	1,0124E+03	1,0572E+03	1,0213E+03
	Media	**2,6965E+02**	**3,1484E+02**	**9,6384E+02**	**9,4272E+02**	**9,6712E+02**
	D.E.	**1,2592E+02**	**1,2488E+02**	**1,8943E+01**	**6,5528E+01**	**4,2093E+01**

TABLA VIII

VALORES DE ERROR: FUNCIONES 21-25 (D=30)

30D		*21*	*22*	*23*	*24*	*25*
1E3	Mejor	1,2326E+03	1,2400E+03	1,2515E+03	1,2754E+03	1,3111E+03
	Peor	1,3472E+03	1,6079E+03	1,3679E+03	1,4077E+03	1,4838E+03
	Media	**1,2922E+03**	**1,3795E+03**	**1,3202E+03**	**1,3556E+03**	**1,4007E+03**
	D.E.	**2,8530E+01**	**7,7417E+01**	**2,7009E+01**	**3,5807E+01**	**3,6333E+01**
1E4	Mejor	6,0313E+02	1,0394E+03	7,6790E+02	4,0485E+02	4,5106E+02
	Peor	1,2699E+03	1,1962E+03	1,2517E+03	1,2855E+03	1,3184E+03
	Media	**9,1470E+02**	**1,1060E+03**	**1,1537E+03**	**7,5546E+02**	**8,7348E+02**
	D.E.	**2,3696E+02**	**4,0694E+01**	**1,0951E+02**	**2,6245E+02**	**3,3561E+02**
1E5	Mejor	5,0000E+02	9,7751E+02	5,3417E+02	2,0000E+02	2,0000E+02
	Peor	1,2674E+03	1,1652E+03	1,2301E+03	1,2808E+03	1,3157E+03
	Media	**6,9548E+02**	**1,0825E+03**	**9,6589E+02**	**3,2816E+02**	**5,1786E+02**
	D.E.	**3,2017E+02**	**4,3045E+01**	**3,1702E+02**	**3,5426E+02**	**5,0547E+02**
3E5	Mejor	5,0000E+02	9,7749E+02	5,3417E+02	2,0000E+02	2,0000E+02
	Peor	1,2673E+03	1,1652E+03	1,2253E+03	1,2806E+03	1,3089E+03
	Media	**6,9528E+02**	**1,0799E+03**	**9,0961E+02**	**3,2815E+02**	**5,1656E+02**
	D.E.	**3,2002E+02**	**4,3104E+01**	**3,3213E+02**	**3,5423E+02**	**5,0413E+02**

IV. DISCUSIÓN

En esta sección se analizarán los valores presentados en las tablas anteriores. Para discutir los resultados se tendrán en cuenta la distinta naturaleza de las funciones propuestas. También se realizarán comparaciones entre pares de funciones de distintas características. Dada la cantidad de datos a considerar, las medias de los valores de error han sido escogidas como valores representativos para mostrar el comportamiento general de los datos. Por esto, todas las consideraciones realizadas han sido hechas principalmente, pero no de manera exclusiva, sobre los valores medios hallados.

A pesar de que todas las funciones de prueba son multimodales, los valores que puede alcanzar el AEF son muy diferentes. Como criterio a seguir, consideramos que mostrar un buen comportamiento implica que los valores de error no crecen más que uno o dos órdenes de magnitud cuando D aumenta. Por lo tanto, una función se comportará peor si sus valores medios de error no consiguen mantenerse en órdenes similares de magnitud.

En general, los menores valores medios se obtienen para las funciones 13 y 14, es decir las multimodales expandidas, seguidos por el grupo de las multimodales sencillas, y por último, las híbridas. Al aumentar la dimensión el grupo más afectado es el de las funciones multimodales sencillas, cuyos valores medios de error varían ostensiblemente en algunos casos. El grupo menos afectado es el de las híbridas, cuyos valores apenas cambian de orden en un par de casos. Las multimodales expandidas, por el contrario pierden un orden de magnitud al aumentar la dimensión.

Entre todas las ejecuciones realizadas, los mejores resultados se han obtenido para la función 9, función multimodal sencilla donde se ha obtenido el nivel de precisión deseado en una media de menos de 18000 evaluaciones para D=10. Las siguientes dos funciones con mejores resultados se encuentran en el grupo de las expandidas (F13 y F14), seguidas por la F11, también multimodal sencilla. Esto implica que se pueden obtener resultados relativamente buenos en dos de los tres grupos considerados, con la excepción del grupo de funciones híbridas. En este último grupo, se observa que en la práctica totalidad de los casos, el AEF llega a óptimos locales en muy pocas evaluaciones (unas 10000). En estos casos, se observa una clara tendencia al estancamiento en respuesta a la gran dificultad de resolución que presentan dichas funciones, pese a los esfuerzos realizados implementando mecanismos para evitar que se cayera en dicha convergencia prematura. La F8 muestra un comportamiento similar. Sus valores de error son aproximadamente constantes y similares a 20 sea cual sea la dimensión estudiada.

Se puede también realizar comparaciones entre grupos de funciones de distinta naturaleza. Por ejemplo, se puede estudiar el comportamiento de las funciones cuyo óptimo global está fuera del rango de inicialización (F7 y F25). La función sencilla F7 obtiene resultados bastantes buenos en el límite superior de evaluaciones para las distintas dimensiones (1,8 en 1000000 evaluaciones y 3,5 a las 300000), mientras que para la híbrida no se obtienen tan buenos resultados. La explicación de esta diferencia de comportamiento es la fuerte dependencia del rango de inicialización para encontrar el óptimo en cada ejecución, resultando favorecido el caso predeterminado expandido.

Se han considerado asimismo dos funciones de distinta naturaleza cuyo óptimo estaba en la frontera: F8 y F20. El comportamiento del AEF es similar y bastante bueno en ambas, no variando mucho el valor medio al aumentar D.

Es también interesante comparar las funciones con ruido y sin ruido. En este caso comparamos dos del grupo de las híbridas, F16 y F17, que no parecen demasiado afectadas por la presencia del ruido en sus resultados, ya que la función 17 aumenta su valor en 100 unidades solamente. Si se tomaran funciones más sencillas el ruido les afectaría más. Si tenemos en cuenta funciones rotadas frente a sus homónimas no rotadas se sigue la misma tendencia. Por ejemplo al comparara funciones multimodales simples (F9 y F10) frente a multimodales híbridas (F15 y F16) se observa que las sencillas reaccionan peor a la presencia de ruido.

Al tomar funciones cuyo óptimo global no está situado dentro de la frontera (F18) y compararlas con funciones con el óptimo en la frontera (F20), los valores alcanzados no varían. Ocurre lo mismo al considerar funciones cuyo óptimo global está situado en una cuenca o zona estrecha (F19) o en una más ancha (F18).

El AEF funciona peor cuando se implementa una matriz con una alta condición numérica (F22) frente a funciones con matrices ortogonales (F21). Mientras que para el primer caso los valores de error medios aumentan con D, dichos valores disminuyen claramente para el segundo caso.

Por el contrario, y en claro contraste con el resto de casos estudiados anteriormente, si se compara la función continua (F21) frente a su versión no continua (F23), los valores de error disminuyen para ambas funciones al aumentar D. La versión no continua presenta mejores resultados que la continua. Lo mismo pasa al comparar una función con el óptimo global dentro de su rango de inicialización (F24) con la misma función pero con su óptimo fuera del mismo (F25). No obstante, todos ellos muestran convergencia prematura.

V. CONCLUSIONES

En este trabajo se ha descrito una versión de un algoritmo evolutivo llamado **Algoritmo Evolutivo Flexible (AEF)**. El AEF posee varias características que le otorgan una gran flexibilidad y una amplia aplicabilidad. La manera en que el conjunto de 60 operadores de cruce y mutación incluidos en el *Motor de Muestreo* es usado es lo que se denomina *Estructura Dinámica de Operadores (EDO)*. Esta estructura ayuda a explorar y explotar el espacio de búsqueda de una manera más eficiente, obteniendo generalmente mejores resultados que cuando se usa un único operador. El *Código Genético Extendido (CGE)* que contiene no sólo las variables de los individuos sino también información sobre los métodos de muestreo usados y los parámetros que controlan dicho muestreo, ha dotado al AEF con un comportamiento más robusto ya que mantiene cierta memoria sobre lo que está pasando en el proceso de optimización. Dicha información puede ser usada para redirigir el proceso de optimización. Además, la inclusión de reglas basadas en estadísticas a modo de *Mecanismos de Control Central (MCC)* para prevenir el estancamiento ha probado ser de utilidad al tratar de superar óptimos locales. El método ha sido probado en un conjunto de 20 funciones propuesto para la sesión especial del MAEB'09 *Metaheurísticas, Algoritmos Evolutivos y Bioinspirados para Problemas de Optimización Continua.*

En cuanto a los resultados obtenidos, los mejores valores se han conseguido para las funciones multimodales expandidas y las multimodales sencillas, mientras que las híbridas han alcanzado peores valores medios de error. Es interesante señalar que al considerar funciones más complicadas, como las híbridas, el comportamiento del algoritmo parece verse afectado en menor

medida si hay rotación, ruido o cualquier otro efecto. También se ha de mencionar que en casi todos los casos se ha llegado de manera muy rápida al óptimo, bien sea local o global, necesitando un número de evaluaciones en torno a 30000, y luego las variaciones han resultado ser mínimas. Sin embargo, en general el método ha encontrado algunas dificultades tales como presentar convergencia prematura a óptimos locales especialmente al ser ejecutado en funciones multimodales híbridas. En consecuencia nuevas reglas o procedimientos deben ser implementadas para resolver dichos problemas.

Para terminar, es importante comentar que el AEF es una implementación prácticamente sin parámetros, donde sólo hay que especificar el tamaño de la población (generalmente fijado en un valor de 100 individuos), el número de evaluaciones de la función objetivo y el elitismo usado, que es de un 15%. Los parámetros restantes son los de control sobre el proceso de la optimización y son ajustados durante la ejecución del proceso de optimización. Esta circunstancia elimina el ajuste preliminar de parámetros necesario al aplicar otros métodos a cada problema particular. Así se eliminan tiempo de computación y de evaluaciones, ya que no es necesario realizar dichos ajustes.

REFERENCIAS

[1] J.D. Schaffer, R. Caruana, L. Eshelman y R. Das (1989).“A Study Of Control Parameters Affecting Online Performance Of genetic Algorithms For Function Optimization”, En Proceedings of the 3rd International Conference on Genetic Algorithms, editado por J.D. Schaffer, Morgan Kaufmann Publishers, San Mateo, CA, págs.51-60.

[2] L. Davis (1989). “Adapting Operator Probabilities In Genetic Algorithms”. En Proceedings of the 3rd International Conference on Genetic Algorithms, editado por J.D. Schaffer, Morgan Kaufmann Publishers, San Mateo, CA , págs. 61-69.

[3] B.A. Julstrom (1995). “What have you done for me lately” Adapting Operator Probabilities in a Steady-State Genetic Algorithm”. En Proceedings of the Sixth International Conference on Genetic Algorithms, (Larry J. Eshelman, Ed.). Morgan Kaufmann Publishers, San Francisco, California, págs.81-87.

[4] M. Srinivas y L. Patnaik (1994). “Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms” . IEEE Transactions on Systems, Man, and Cybernetics 24(4). Páginas 656-667.

[5] A.H. Wright y A. Agapie (2001). “Cyclic y Chaotic Behavior in Genetic Algorithms”. En Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, San Francisco, California. GECCO 2001, págs.718-724.

[6] N.N. Schraudolph y R.K. Belew (1992). “Dynamic Parameter Encoding for Genetic Algorithms”. Machine Learning, Vol. 9:1, pp 9-21.

[7] P.J. Darwen (2000). “Black Magic: Interdependence Prevents Principled Parameter Setting, self-adapting costs too much computation”. En Applied Complexity: From Neural Nets to Managed Lyscapes, págs 227-237. Institute for Food and Crop Research, Christchurch, New Zealand.

[8] B. Edmonds (1998). “Meta-Genetic Programming: Co-evolving the Operators of Variation”. CPM Informe Nº: 98-32. Centre for Policy Modelling, Manchester Metropolitan University.

[9] T.P. Hong, H.S. Wang, W.Y. Lin y W.Y. Lee (2002). “Evolution of Appropriate Crossover and Mutation Operators in a Genetic Process”. En Applied Intelligence Journal 16(1): Páginas 7-17**.** Publicado por Springer Science+Business Media B.V., anteriormente Kluwer Academic Publishers B.V. ISSN: 0924-669X (Artículo) 1573-7497 (Online) DOI: 10.1023/A:1012815625611 .

[10] G. Cervone, K.A. Kaufman y R.S. Michalski (2002). “Recent Results from the Experimental Evaluation of the Learnable Evolution Model”. En Late-Breaking Papers of the Genetic and Evolutionary Computation Conference, Erik Cantú-Paz (Lawrence Livermore National Laboratory) Editor, New York. GECCO 2002, pp 47-54.

[11] M.C.J. Bot, N Urquhart y K Chisholm (2001). “Agent Motion Planning with GAs Enhanced by Memory Models”. En Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, San Francisco, California. GECCO 2001, págs..227-234.

[12] G. Winter, B. Galván, P. Cuesta ,y S. Alonso (2002).“Flexible Evolution”.En K.C. Giannakoglou, D. T. Tsahalis, J. Périaux, K.D. Papaliouy T. Fogarty (editores), Evolutionary Methods for Design, Optimization y Control with Applications to Industrial Problems. EUROGEN 2001, Athens. ED. CIMNE, Barcelona, España, págs. 89-94.ISBN: 84-89925-97-6.

[13] G. Winter, B. Galván, S. Alonso y B. González (2002). “Evolving From Genetic Algorithms to Flexible Evolution Agents”. En Late-Breaking Papers of the Genetic and Evolutionary Computation Conference, Erik Cantú-Paz (Lawrence Livermore National Laboratory) Editor, New York. GECCO 2002, págs. 466-473.

[14] G. Winter, B. Galván, S. Alonso, B. González, D. Greiner y J.I. Jiménez (2004) . “A Flexible Evolutionary Agent: cooperation and Competition among real-coded evolutionary operators”. Incluido en Soft Computing - A Fusion of Foundations, Methodologies and Applications, Vol: 9 (4) pp, 299-323. Springer-Verlag Gmbh. ISSN: 1432-7643.(Paper) 1433-7479. (Online)DOI: 10.1007/s00500-004-0381-8.

[15] Z. Michalewicz, G. Nazhiyath, y M. Michalewicz (1996).“A Note on Usefulness of Geometrical Crossover for Numerical Optimization Problems”. En L.J. Fogel, P.J. Angeline, y Th. Bäck, editores, Proceedings of the fifth Annual Conference on Evolutionary programming. The MIT Press, Cambridge, MA, págs. 305-312.

[16] F. Herrera , M. Lozano y J. L. Verdegay (1996)**.** *"Dynamic and heuristic fuzzy connectives based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms“***.** International Journal of Intelligent Systems 11(12):páginas 1013--1040.

[17] F. Herrera, M. Lozano y A.M. Sánchez (2003). “*A Taxonomy for the Crossover Operador for Real-Coded Genetic Algorithms: an Experimental Study*”. International Journal of intelligents systems, vol. 18, páginas 309-338. Wiley Periodicals, Inc. Publisehd online in Wiley InterScience. DOI:10.1002/int.10091. ISSN: 0884-8173.

[18] T. Bäck y B. Naujoks (1998). “*Innovative Methodologies in Evolution Strategies”.* Informe D2.2 de INGENET Networks. H. Mühlenbein y D. Schlierkamp-Voosen (1993).“Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization”. En Evolutionary Computation, 1(1): páginas 25-49.

[19] H. Mühlenbein and D. Schierkamp-Voosen (1993). *“Predictive Models for the Breeder Genetic algorithm I. Continuous Parameter Optimization”* Evolutionary computation, Vol. 1 Issue 1, págs. 25-49.

[20] F. Herrera , M. Lozano (2003)**.** *"Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions”.*Soft Computing 7, págs 545562. Springer Verlag 2003 DOI 10.1007 .