

Algoritmos Distribuidos Heterogéneos para problemas de Optimización Continua

S. Muelas, J.M.Peña, A.LaTorre y V. Robles

Resumen— Este trabajo analiza el comportamiento de dos algoritmos distribuidos heterogéneos que combinan las siguientes técnicas evolutivas: Algoritmos Genéticos (GA), Algoritmos de Estimación de Distribución (EDA) y Evolución Diferencial (DE) sobre el conjunto de funciones de pruebas desarrolladas para la sesión de optimización continua del CEC'05. Los resultados obtenidos demuestran que la interacción de estos algoritmos, por medio del modelo distribuido, obtiene unos resultados muy competitivos, mejorando o igualando en varias funciones, al mejor resultado obtenido por los tres mejores algoritmos que se propusieron para la sesión del CEC'05.

Palabras clave— algoritmos genéticos, algoritmos de estimación de distribución, evolución diferencial, algoritmos distribuidos

I. INTRODUCCIÓN

Los algoritmos evolutivos han demostrado ser una técnica de optimización muy potente para un gran conjunto de problemas de optimización. En la actualidad, existe un amplio conjunto de este tipo de algoritmos que han sido aplicados satisfactoriamente a funciones de muy diversas características [1]. Las diferentes variantes de algoritmos evolutivos siguen procesos de búsqueda muy distintos, siendo cada una de ellas más apropiadas para una determinada clase de problemas y no existiendo ninguna técnica ni selección de parámetros que sea sistemáticamente superior sobre todo el espacio posible de problemas de optimización [2]. Por lo tanto, es normal que se hayan desarrollado distintos mecanismos para poder combinar las propiedades beneficiosas de cada una de estas técnicas [3], [4], [5], [6], [7]. Uno de los más representativos son los llamados algoritmos evolutivos distribuidos heterogéneos, una subclase de los algoritmos evolutivos descentralizados [8] desarrollados con el fin de preservar la diversidad, evitar el problema de la convergencia prematura y facilitar la búsqueda de soluciones alternativas. La idea básica de estos algoritmos radica en el particionamiento de la población en varias subpoblaciones, cada una de ellas procesadas por un algoritmo independiente. El intercambio de información se produce por medio de un mecanismo de migración de individuos en base a una topología de conexión. Gracias a la combinación de distintas técnicas, estos algoritmos son considerados una herramienta eficaz para evitar el problema de la convergencia prematura e hibridación de técnicas de búsqueda para maximizar la exploración y la explotación en el espacio de búsqueda global [9], [7].

En este trabajo, se analiza el comportamiento de dos algoritmos distribuidos heterogéneos que hacen uso de algoritmos evolutivos que han obtenido buenos resultados en la literatura con problemas continuos (Algoritmos Genéticos, Algoritmos de Estimación de Distribu-

ción y Evolución Diferencial), sobre el conjunto de funciones seleccionadas para la sesión especial de optimización continua del VI Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB).

Este documento está estructurado de la siguiente forma: En la sección II se describen los algoritmos utilizados y algunos de los trabajos realizados con modelos distribuidos. La sección III presenta los dos algoritmos distribuidos propuestos, y en la sección IV se describen los resultados obtenidos. Finalmente, la sección V concluye el trabajo.

II. ESTADO DE LA CUESTIÓN

Los algoritmos genéticos (*Genetic Algorithms* – GAs) [10], [11] son una de las técnicas evolutivas que más se han utilizado para resolver problemas de optimización. El GA es un método de búsqueda basado en poblaciones que reproduce el ciclo de la vida: selección de individuos, reproducción de éstos por medio de operadores genéticos de cruce y mutación para generar a la descendencia y un mecanismo de reemplazo para seleccionar a los mejores individuos entre los progenitores y los generados.

Los algoritmos de estimación de distribuciones (*Estimation of Distribution Algorithms* – EDAs) [12], [13] son una clase de algoritmos evolutivos que hacen uso de modelos gráficos probabilísticos en vez de operadores genéticos. En un EDA, la nueva población se obtiene por medio del muestreo de las distribuciones de probabilidad descritas por el modelo que se estima basándose en una selección de individuos prometedores de la población actual.

El algoritmo Evolución Diferencial (*Differential Evolution* – DE) es un algoritmo evolutivo que ha surgido en los últimos años y que está obteniendo cada vez más atención. Al igual que otros algoritmos evolutivos, es un optimizador estocástico basado en poblaciones de soluciones. En el proceso de mutación del DE, la diferencia ponderada de dos individuos seleccionados aleatoriamente, se añade a un tercer individuo para generar uno nuevo. Por cada individuo de la población se realiza este paso, generándose un individuo nuevo por medio del cruce entre el individuo seleccionado y el resultante de la mutación. El mejor de estos dos individuos (el seleccionado y el nuevo generado) formará parte de la población de la siguiente iteración. Este proceso se repite iterativamente hasta que se satisface el criterio de convergencia. El DE fue presentado para resolver el problema de ajuste de polinomios de Chebychev [14] y desde entonces se ha aplicado satisfactoriamente en una amplia colección de problemas [15], [16], [17], [18]

Un modelo distribuido evolutivo, también llamado

DATSI, Facultad de Informática, Universidad Politécnica de Madrid
E-mail: {smuelas,jmpena,atorre,vrobles}@fi.upm.es .

modelo de islas, es una manera de distribuir un algoritmo evolutivo por medio de la división de individuos en subpoblaciones y del ocasional intercambio entre éstas (migraciones). En el modelo más sencillo, el modelo homogéneo, las islas comparten el mismo algoritmo así como sus parámetros. Las migraciones quedan definidas por su topología de conexión, el número de individuos migrados, la frecuencia de éstas y la política de emigración/inmigración (cómo se seleccionan y se añaden los individuos). Se han publicado numerosos trabajos de estos modelos, algunos de ellos basados en modelos teóricos y otros en escenarios experimentales [19], [20], [21], [22]. Si los algoritmos que se ejecutan en cada isla utilizan distintas estrategias de búsqueda, el modelo se conoce como modelo heterogéneo distribuido. Los algoritmos distribuidos heterogéneos se han utilizado extensamente en la literatura con el fin de evitar la convergencia prematura y para maximizar la exploración y la explotación sobre el espacio de búsqueda. En función del grado de heterogeneidad, se distinguen los siguientes niveles [4]:

- **Parámetro:** La forma más sencilla de obtener heterogeneidad en las islas es modificando el valor de sus parámetros, bien de manera estática [23], aleatoria [24] o dinámica [25], [26].

- **Operador:** En este nivel la heterogeneidad se consigue utilizando distintos operadores evolutivos. Un claro exponente de este tipo de heterogeneidad es el algoritmo GD-RCGA [7]. Este algoritmo utiliza ocho islas distintas conectadas por medio de una topología de cubo en donde cuatro islas tienen un enfoque más explotativo y las otras cuatro uno más explorativo. Estos enfoques se consiguen por medio de la aplicación de distintos operadores de cruce y de distintos valores de presión selectiva. Gracias a esta combinación de islas, se obtiene simultáneamente un efecto de búsqueda diversificada (fiabilidad) y un eficaz ajuste local (precisión). Este algoritmo fue mejorado en [4] aumentando el número de islas a 16 y utilizando una topología de hipercubo de cuatro dimensiones.

- **Genotipo:** En este nivel, la codificación de las soluciones varía de una isla a otra. Un claro exponente de este nivel es el algoritmo "Injection Island" [21] en donde cada subpoblación codifica las soluciones con distintos niveles de resolución. Las subpoblaciones *inyectan* sus mejores individuos a islas de mayor resolución para una optimización con un ajuste más fino. Esto permite que la búsqueda se produzca simultáneamente en distintos niveles de resolución. También se han utilizado distintas codificaciones con el fin de escapar del conjunto de óptimos locales, ya que un óptimo local en una representación puede no serlo en otra distinta [27]. Por lo tanto, el uso combinado de distintas codificaciones permite aumentar la probabilidad de encontrar el óptimo global.

- **Algoritmo:** Este es el nivel más general de heterogeneidad en donde cada isla ejecuta un algoritmo completamente distinto [28], [29]. Este nivel de heterogeneidad ha sido el utilizado para los modelos distribuidos propuestos en este trabajo.

Además de estos niveles de heterogeneidad, se pueden distinguir otros más en función de la variación del número de individuos por isla a lo largo de la evolución. De

esta manera, si la cantidad de individuos varía en función de la calidad de los últimos generados, el modelo será competitivo [25], [6], [5]. Por contra, si todas las islas mantienen fijo su número de individuos, el modelo se considera colaborativo [23], [30].

III. CONTRIBUCIÓN

En este trabajo se analiza el comportamiento de dos modelos heterogéneos distribuidos que hacen uso de varios algoritmos evolutivos con distintas estrategias de búsqueda y que han obtenido en la literatura buenos resultados de forma aislada con problemas de espacios continuos. Como ya se ha mencionado, estos algoritmos son los siguientes: Algoritmos Genéticos, Algoritmos de Estimación de Distribución y Evolución Diferencial clásica. Por cada uno de estos algoritmos se realizó un estudio previo con el fin de optimizar la combinación de parámetros que mejor resultados obtuviese. Para ello se realizaron 25 ejecuciones sobre el conjunto de problemas fijado para múltiples combinaciones de parámetros y se evaluaron dichas combinaciones por medio de tests estadísticos no paramétricos utilizando como criterio un p -valor $< 0,01$. En base a estos resultados se seleccionaron las configuraciones detalladas en las tablas I, II y III.

TABLA I
PARÁMETROS UTILIZADOS PARA EL ALGORITMO GENÉTICO

Op. de Selección	Uniforme
Op. de Cruce	BLX- α [31] con $\alpha = 0,5$
Op. de Mutación	Gausiano
Prob. Cruce	100 %
Prob. Mutación	5 %
Reemplazo	50 % Elitismo

TABLA II
PARÁMETROS UTILIZADOS PARA EL ALGORITMO DE ESTIMACIÓN DE LA DISTRIBUCIÓN

Op. de Selección	Uniforme
Modelo de Aprendizaje	UMDA _c
Reemplazo	50 % Elitismo

TABLA III
PARÁMETROS UTILIZADOS PARA LA EVOLUCIÓN DIFERENCIAL

Op. de Selección	Uniforme
Op. de Cruce	Exponencial
F	0.9
CR	0.5

Una vez seleccionados los parámetros de cada algoritmo, se procedió a optimizar la configuración del modelo distribuido. Se probaron configuraciones de dos y

tres islas con distintos valores para los parámetros de migración. En la tabla IV se representan los valores que se escogieron en base a los resultados de las pruebas.

TABLA IV
PARÁMETROS UTILIZADOS PARA EL MODELO DISTRIBUIDO

Período de Migración	120 generaciones
Nº inds migrados	10
Topología	Anillo simple
Pol. de Selección	Uniforme
Pol. de Reemplazo	Peores individuos

Se analizaron también distintos tamaños de población teniendo siempre en cuenta la restricción de la sesión con respecto al número máximo de evaluaciones, como consecuencia de lo cual se procedió a ajustar lo máximo posible el tamaño de las islas con el fin de no decrementar considerablemente el número máximo permitido de generaciones. Los mejores resultados se obtuvieron con poblaciones de 50 individuos en configuraciones de dos islas. Para la configuración de tres islas, el mejor resultado se obtuvo con una configuración asimétrica de 50 individuos tanto para el GA como para el EDA y de 30 individuos para el DE. Tanto el GA como el EDA necesitan tener un mayor número de individuos que el DE debido a sus mayores problemas de convergencia prematura. Por contra, el DE necesita un mayor número de generaciones para obtener un buen rendimiento [32]. Por esta razón, con la configuración de tres islas se obtuvieron mejores resultados al reducir el número de individuos del DE.

IV. RESULTADOS

En esta sección se realiza un análisis de los resultados obtenidos por dos de las mejores configuraciones distribuidas heterogéneas analizadas: la mejor configuración de dos islas (GA-DE) y la de tres islas que hace uso de todos los algoritmos. Se realizaron 25 ejecuciones sobre el subconjunto de funciones $f6 - f25$ de la sesión especial de optimización continua del CEC'05 [33] tanto con 10 como con 30 dimensiones. El número máximo de evaluaciones se fijó en 100.000 evaluaciones para las funciones de 10 dimensiones y 300.000 para las de 30. Se registró la media del error obtenido con respecto al óptimo global, así como la de su desviación típica. En las tablas V y VI se muestran los resultados obtenidos para las funciones de 10 y 30 dimensiones, incluyendo tanto al error medio como la desviación típica de los tres algoritmos que mejores resultados obtuvieron en la sesión especial del CEC'05¹. Además, en las columnas correspondientes a los algoritmos heterogéneos, se ha indicado si el error medio del algoritmo distribuido obtiene mejor o igual resultado que cada uno de los algoritmos de referencia.

En base a los resultados representados en la tabla V se puede observar que, con 10 dimensiones, ambos al-

¹Debido a que no se ha podido obtener los valores de la desviación típica del DE propuesto para la sesión, en las tablas sólo se representa su error medio

goritmos obtienen unos resultados competitivos. Sus resultados igualan o mejoran el de alguna de las funciones de referencia en 17 de las 20 funciones propuestas y en 6 de ellas, los resultados son mejores al de todas ellas. Además, hay dos funciones, $f9$ y $f15$, que son resueltas en todas las ejecuciones por el algoritmo distribuido de dos islas. Es conveniente destacar que estas valoraciones están hechas teniendo en cuenta únicamente el error medio y que para poder realizar una afirmación más contundente, habría que realizar algún test estadístico que confirmase estas diferencias en las medias ya que en una gran parte de las funciones, se aprecia una dispersión alta de los resultados.

También se aprecia que en la mayoría de las funciones compuestas, el error medio es muy considerable y, aunque habría que realizar un estudio del espacio del fitness, da la impresión que los valores obtenidos por todos los algoritmos se encuentran muy distanciados con respecto al óptimo global. De hecho, y al igual que sucedió con la sesión especial del CEC'05, 13 de las 20 funciones no han sido resueltas por ninguna ejecución (siendo compuestas la mayor parte de estas funciones).

De las tres funciones en donde los modelos distribuidos han obtenido peores resultados, $f6$ es en donde se han obtenido los peores resultados con respecto a las tres funciones de referencia. En la figura 1 se ha representado la evolución de la media del valor de la función objetivo a lo largo del número de evaluaciones, tanto para los algoritmos distribuidos como para la versiones secuenciales de cada uno de los tipos de algoritmos. Se puede observar que, en este caso, el DE es el que obtiene el mejor resultado y que las gráficas de los algoritmos distribuidos son muy similares a la del DE, estando éstas desplazadas con respecto al número de evaluaciones. Esta función se caracteriza por tener un estrecho y largo "pasillo" desde el óptimo local al óptimo global por lo que el DE necesita un número alto de generaciones para poder alcanzarlo. Esta característica hace que tanto el GA como el EDA tengan serias dificultades para mejorar el rendimiento del DE, por lo que su inclusión en el modelo distribuido lo único que acarrea es una perjudicial reducción del número de generaciones del DE. Otro ejemplo de influencia negativa por parte de una isla lo encontramos en las funciones $f7$ y $f13$. En este caso, el algoritmo EDA perjudica notablemente el rendimiento del algoritmo de tres islas, haciéndole converger prematuramente en un óptimo local (figura 2). Estos casos pueden considerarse como casos particulares de los cuales una de las técnicas es claramente mejor/peor que las otras. En esos casos la combinación de ellas implica reducir el número de evaluaciones que la técnica más apropiada puede realizar al consumirse evaluaciones en los individuos generados por técnicas menos apropiadas.

En el extremo opuesto se encuentran las funciones $f12$ y $f17$ donde cada algoritmo del modelo distribuido, realiza su aportación en distintos momentos de la evolución, obteniendo el mejor resultado con respecto a la versión secuencial de los mismos (figura 3). Para entender mejor la aportación de cada isla, se ha generado la figura 4.

TABLA V
RESULTADOS OBTENIDOS CON DIEZ DIMENSIONES

	GA-DE	GA-EDA-DE	G-CMA-ES	DE	K-PCX
f6	1.10e+00/1.30e+00	2.55e+00/2.22e+00	4.87e-09/1.66e-09	1.59e-01	4.78e-01/1.32e+00
f7	1.92e-01/1.10e-01	2.73e-01/1.28e-01	3.31e-09/2.02e-09	1.46e-01	2.31e-01/2.12e-01
f8	2.01e+01/1.01e-01 D	2.02e+01/7.00e-02 D	2.00e+01/3.89e-03	2.04e+01	2.00e+01/3.25e-07
f9	0.00e+00/0.00e+00 GDK	0.00e+00/0.00e+00 GDK	2.39e-01/4.34e-01	9.55e-01	1.19e-01/5.97e-01
f10	1.10e+01/4.31e+00 D	4.7e+00/1.49e+00 D	7.96e-02/2.75e-01	1.25e+01	2.39e-01/7.20e-01
f11	4.77e+00/1.59e+00 K	3.95e+00/1.51e+00 K	9.34e-01/9.00e-01	8.47e-01	6.65e+00/1.66e+00
f12	6.54e+01/1.09e+02 K	2.85e+01/3.69e+00 GDK	2.93e+01/1.42e+02	3.17e+01	1.49e+02/4.18e+02
f13	2.77e-01/7.41e-02 GDK	4.10e-01/9.50e-02 DK	6.96e-01/1.50e-01	9.77e-01	6.53e-01/2.06e-01
f14	2.97e+00/5.12e-01 GD	2.80e+00/2.74e-01 GD	3.01e+00/3.49e-01	3.45e+00	2.35e+00/4.82e-01
f15	0.00e+00/0.00e+00 GDK	1.96e+00/9.82e+00 GDK	2.28e+02/6.80e+01	2.59e+02	5.10e+02/1.27e+02
f16	1.12e+02/1.46e+01 D	9.74e+01/3.92e+00 D	9.13e+01/3.49e+00	1.13e+02	9.59e+01/6.85e+00
f17	1.14e+02/1.40e+01 GD	1.06e+02/8.15e+00 GD	1.23e+02/2.09e+01	1.15e+02	9.73e+01/5.55e+00
f18	4.71e+02/1.92e+02 K	5.31e+02/1.98e+02 K	3.32e+02/1.12e+02	4.00e+02	7.52e+02/2.07e+02
f19	5.77e+02/2.24e+02 K	5.77e+02/2.54e+02 K	3.26e+02/9.93e+01	4.20e+02	7.51e+02/2.04e+02
f20	6.12e+02/2.10e+02 K	6.39e+02/2.36e+02 K	3.00e+02/0.00e+00	4.60e+02	8.13e+02/1.63e+02
f21	4.91e+02/4.00e+01 GDK	5.34e+02/1.36e+02 K	5.00e+02/3.48e-13	4.92e+02	1.05e+03/1.16e+02
f22	7.46e+02/9.51e+01	7.51e+02/9.57e+01	7.29e+02/6.86e+00	7.18e+02	6.59e+02/1.33e+02
f23	5.92e+02/1.13e+02 K	6.25e+02/1.36e+02 K	5.59e+02/3.24e-11	5.72e+02	1.06e+03/1.50e+02
f24	1.99e+02/0.00e+00 GDK	2.35e+02/9.9e+01 K	2.00e+02/2.29e-06	2.00e+02	4.06e+02/5.13e-01
f25	3.93e+02/5.82e+01 DK	4.12e+02/2.04e+00 D	3.74e+02/3.22e+00	9.23e+02	4.06e+02/4.88e-01

G representa que el error medio del algoritmo propuesto es igual o inferior al del algoritmo G-CMA-ES de la sesión CEC'05 [34]
D representa que el error medio del algoritmo propuesto es igual o inferior al del algoritmo DE de la sesión CEC'05
K representa que el error medio del algoritmo propuesto es igual o inferior al del algoritmo K-PCX de la sesión CEC'05 [35]

TABLA VI
RESULTADOS OBTENIDOS CON TREINTA DIMENSIONES

	GA-DE	GA-EDA-DE	G-CMA-ES	DE	K-PCX
f6	2.86e+01/2.45e+01	4.31e+01/3.60e+01	5.90e-09/1.61e-09	2.51e+01	1.75e+00/2.02e+00
f7	3.95e-01/8.82e-02	4.06e-01/1.52e-01	5.31e-09/1.41e-09	2.96e-03	1.50e-02/1.15e-02
f8	2.09e+01/6.55e-02 D	2.09e+01/4.47e-02 D	2.01e+01/2.79e-01	2.10e+01	2.00e+01/4.97e-07
f9	0.00e+00/0.00e+00 GDK	0.00e+00/0.00e+00 GDK	9.38e-01/1.18e+00	1.85e+01	2.79e-01/4.56e-01
f10	4.40e+01/1.04e+01 D	3.76e+00/5.78e+00 D	1.65e+00/1.35e+00	9.69e+01	5.17e-01/7.11e-01
f11	1.90e+01/3.33e+00 DK	1.95e+01/5.89e+00 DK	5.48e+00/3.13e+00	3.42e+01	2.95e+01/3.65e+00
f12	1.30e+04/7.10e+03 G	1.12e+04/9.05e+03 G	4.43e+04/2.19e+05	2.75e+03	1.68e+03/3.44e+03
f13	1.53e+00/1.97e-01 GDK	1.85e+00/1.45e-01 GDK	2.49e+00/5.13e-01	3.23e+00	1.19e+01/3.80e+00
f14	1.21e+01/4.39e-01 GDK	1.23e+01/3.42e-01 GD	1.29e+01/4.19e-01	1.34e+01	1.38e+01/1.90e-01
f15	3.69e+01/8.84e+01 GDK	3.04e+02/3.31e+01 DK	2.08e+02/2.75e+01	3.60e+02	8.76e+02/1.95e+01
f16	7.83e+01/3.35e+01 D	7.42e+01/2.38e+01 D	3.50e+01/2.04e+01	2.12e+02	7.15e+01/8.10e+01
f17	1.12e+02/3.74e+01 GDK	1.10e+02/2.96e+01 GDK	2.91e+02/1.93e+02	2.37e+02	1.56e+02/1.58e+02
f18	9.09e+02/1.77e+00	8.81e+02/5.29e+01 GD	9.04e+02/2.88e-01	9.04e+02	8.30e+02/1.61e+00
f19	9.00e+02/3.03e+01 GD	8.93e+02/4.15e+01 GD	9.04e+02/2.71e-01	9.04e+02	8.31e+02/1.47e+00
f20	8.87e+02/4.46e+01 GD	8.93e+02/4.28e+01 GD	9.04e+02/2.48e-01	9.04e+02	8.31e+02/1.32e+00
f21	4.99e+02/0.00e+00 GDK	4.99e+02/0.00e+00 GDK	5.00e+02/1.31e-13	5.00e+02	8.59e+02/5.44e-01
f22	9.20e+02/1.60e+01 K	9.31e+02/1.88e+01 K	8.03e+02/1.86e+01	8.97e+02	1.56e+03/4.83e+02
f23	5.34e+02/2.70e-04 GDK	5.34e+02/5.10e-03 GDK	5.34e+02/2.22e-04	5.34e+02	8.66e+02/8.07e-01
f24	1.99e+02/0.00e+00 GDK	1.99e+02/0.00e+00 GDK	9.10e+02/2.11e+02	2.00e+02	2.13e+02/3.89e-01
f25	2.22e+02/1.39e+00 D	2.22e+02/4.95e+00 D	1.48e+02/9.21e-01	7.30e+02	2.13e+02/5.50e-01

G representa que el error medio del algoritmo propuesto es igual o inferior al del algoritmo G-CMA-ES de la sesión CEC'05 [34]
D representa que el error medio del algoritmo propuesto es igual o inferior al del algoritmo DE de la sesión CEC'05
K representa que el error medio del algoritmo propuesto es igual o inferior al del algoritmo K-PCX de la sesión CEC'05 [35]

En esta figura se representa la evolución sobre el número de generaciones del valor medio de la función objetivo de cada isla. Como se puede observar en esta gráfica, el EDA es el que obtiene los mejores resultados en las primeras generaciones, el GA en la segunda parte y, por último, el DE con mayor capacidad de búsqueda local, es el que se encarga de refinar las soluciones.

En medio de estos extremos se encuentran la mayor parte del resto de las funciones, en donde dos algoritmos van alternando su liderazgo en el proceso de búsqueda a lo largo de la evolución. El ejemplo más representativo de este efecto lo podemos encontrar en la función *f15*,

donde, gracias a la acción combinada de las estrategias de búsqueda tanto el DE como del GA, se consigue obtener un proporción de éxitos ² muy elevada: 88 % y 100 % para los modelos de tres y dos islas frente a los valores 4 %, 32 % y 48 % de los algoritmos secuenciales EDA, GA y DE. La colaboración de ambos algoritmos se encuentra representada en la figura 5 donde se muestra el valor medio de cada isla del algoritmo distribuido de dos islas a lo largo de la evolución.

Si comparamos los resultados de ambos modelos distribuidos para el conjunto de todas las funciones por me-
²nº ejecuciones que alcanzan el óptimo global / nº ejecuciones totales

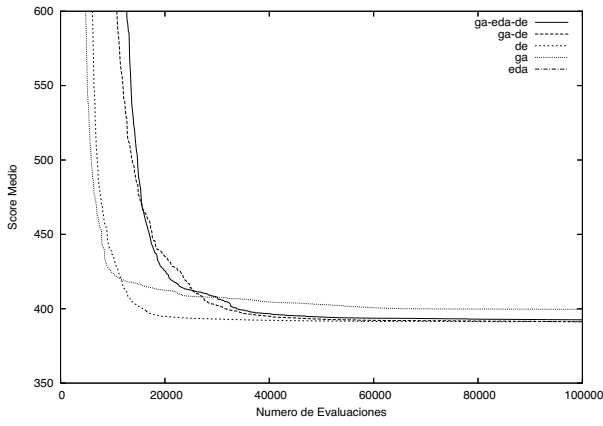


Fig. 1

EVOLUCIÓN DEL SCORE MEDIO EN LA FUNCIÓN f_6 CON 10 DIMENSIONES

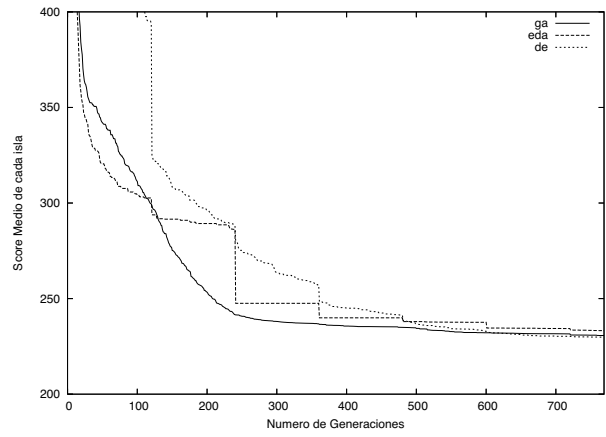


Fig. 4

EVOLUCIÓN DEL SCORE MEDIO DE CADA ISLA DEL ALGORITMO GA-EDA-DE EN LA FUNCIÓN F17 CON 10 DIMENSIONES

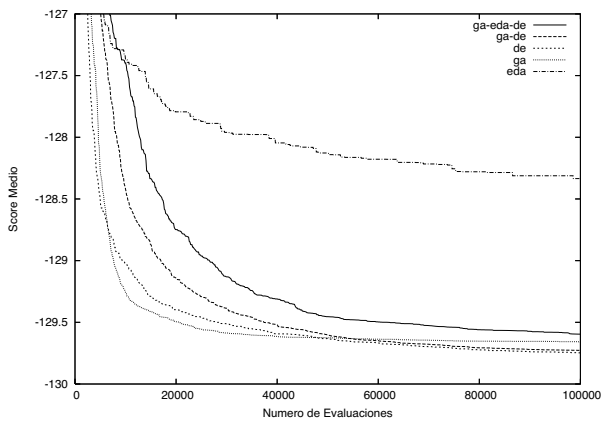


Fig. 2

EVOLUCIÓN DEL SCORE MEDIO EN LA FUNCIÓN f_{13} CON 10 DIMENSIONES

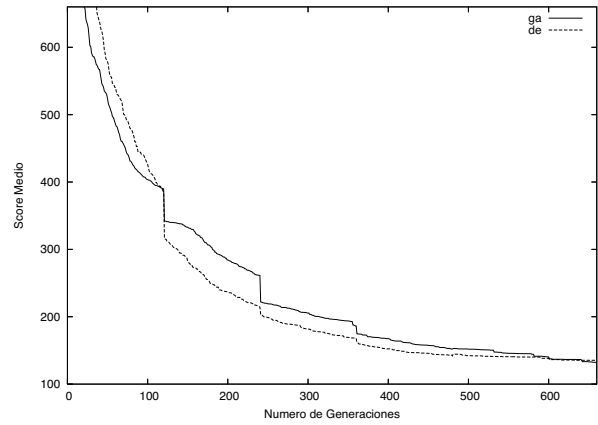


Fig. 5

EVOLUCIÓN DEL SCORE MEDIO DE CADA ISLA DEL ALGORITMO GA-DE EN LA FUNCIÓN F15 CON 10 DIMENSIONES

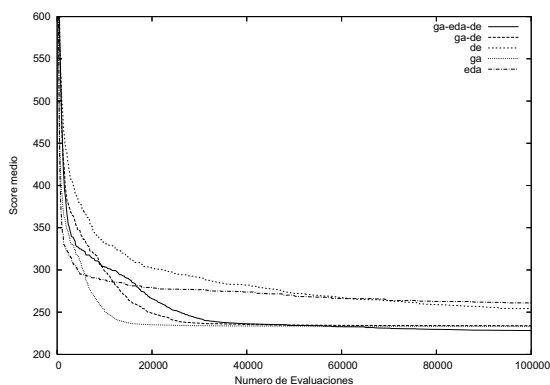


Fig. 3

EVOLUCIÓN DEL SCORE MEDIO EN LA FUNCIÓN F17 CON 10 DIMENSIONES

dio de un test estadístico no paramétrico de Wilcoxon, veremos que la versión distribuida de tres islas obtiene resultados significativamente mejores³ que la de dos en tres de las veinte funciones mientras que la de dos mejora a la de tres en seis funciones. Por lo tanto, la versión de dos islas obtiene mejores resultados globales tanto a nivel de errores medios como a nivel del test de Wilcoxon. La razón principal de este fenómeno radica en que la inserción de la tercera isla con el algoritmo EDA, sólo influye considerablemente en un número reducido de funciones con la principal desventaja de que se limita todavía más el número máximo de generaciones.

Con respecto a las funciones de 30 dimensiones, vemos que las tendencias observadas en el análisis anterior son muy parecidas al de las observadas con estas funciones. En las dos primeras funciones, f_6 y f_7 , son donde peores resultados se obtienen frente a las funciones f_9 y f_{15} , donde se obtienen los mejores resultados. El número de funciones donde los algoritmos propuestos

³ p -valor $< 0,01$

han igualado o mejorado el resultado de los algoritmos de referencia se ha visto incrementado en dos funciones, siendo las funciones compuestas en las que más se ha mejorado. De todas formas, tanto los errores medios como las desviaciones típicas han incrementado notablemente su valor en la mayoría de las funciones, complicando todavía más la comparación entre los algoritmos. Con respecto a los dos algoritmos propuestos analizados por medio del test de Wilcoxon, el de tres islas ha obtenido resultados significativamente superiores al de dos en 6 de las 20 funciones, frente a 1 del algoritmo de tres islas.

A la vista de los resultados tanto en 10 como en 30 dimensiones de las funciones $f_9 - f_{10}$ y $f_{15} - f_{16}$, se puede deducir que los problemas rotados suponen una mayor dificultad para los algoritmos propuestos. Igualmente sucede con las comparaciones entre las funciones continuas y no continuas ($f_{21} - f_{23}$), las que utilizan una matriz ortogonal frente a una mal condicionada ($f_{21} - f_{22}$) y las que tienen el óptimo global en el rango de inicialización frente a las que no lo tienen ($f_{24} - f_{25}$). Sin embargo, parece que la reducción del polo de atracción ($f_{18} - f_{19}$) y el tener o no el óptimo en el extremo de la región ($f_{18} - f_{20}$), no influyen en el proceso de búsqueda. De todas formas, al no haber alcanzado en ninguna de estas funciones el óptimo global, no se puede asegurar que estas características no supongan una dificultad para el proceso de búsqueda de estos algoritmos.

V. CONCLUSIONES

En este trabajo se ha analizado el comportamiento de dos algoritmos distribuidos heterogéneos sobre un subconjunto de funciones que se propuso para la sesión especial de optimización continua del CEC'05. Ambos algoritmos han obtenido unos resultados muy competitivos demostrando que la interacción de los algoritmos GA, EDA y DE puede resultar muy beneficiosa de cara a resolver funciones con distintos grados de dificultad en un espacio de búsqueda continuo. El modelo de dos islas con los algoritmos GA-DE ha obtenido globalmente mejores resultados que el de tres islas debido, principalmente, al mayor número de generaciones disponibles y al lastre derivado del bajo rendimiento del algoritmo EDA en algunas de las funciones analizadas. Finalmente, de cara a futuras sesiones especiales, nos gustaría proponer aumentar el número de evaluaciones máximo así como plantear funciones cuya dificultad no radique en la rugosidad extrema y discontinuidad del espacio de búsqueda, con el fin de ser capaces de alcanzar el óptimo global y de realizar en base a ese criterio las comparaciones entre algoritmos. Con las limitaciones actuales, tanto los algoritmos propuestos como los de referencia, obtienen unos errores medios muy altos, quedándose muy lejos del óptimo global y siendo muy difícil evaluar la calidad del proceso de búsqueda de cada una de ellos.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia (TIN2007- 67148), la Consejería de Educación de la Comunidad de Madrid y el Fondo Social

Europeo (F.S.E.). El autor con agradecimiento reconoce los recursos informáticos, conocimientos técnicos y asistencia proporcionada por el Centro de Supercomputación y Visualización de Madrid (CeSViMa) y la Red española de Supercomputación.

REFERENCIAS

- [1] T. Back, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*, IOP Publishing Ltd., Bristol, UK, UK, 1997.
- [2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.
- [3] A. Latorre, J. M. Peña, V. Robles, and S. Muelas, "Using multiple offspring sampling to guide genetic algorithms to solve permutation problems," in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, New York, NY, USA, 2008, pp. 1119–1120, ACM.
- [4] E. Alba, F. Luna, A. J. Nebro, and J. M. Troya, "Parallel heterogeneous genetic algorithms for continuous optimization," *Parallel Comput.*, vol. 30, no. 5-6, pp. 699–719, 2004.
- [5] A. E. Eiben, I. G. Sprinkhuizen-Kuyper, and B. A. Thijssen, "Competing crossovers in an adaptive ga framework," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 787–792.
- [6] D. Schlierkamp-Voosen and H. Muhlenbein, "Adaptation of population sizes by competing subpopulations," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, 1996, pp. 330–335.
- [7] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 43–63, 2000.
- [8] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 5, pp. 443–462, 2002.
- [9] J. C. Potts, T. D. Giddens, and S. B. Yadav, "The development and evaluation of an improved genetic algorithm based on migration and artificial selection," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 1, pp. 73–86, 1994.
- [10] John H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [11] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [12] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer, October 2001.
- [13] Heinz Muhlenbein and Gerhard Paass, "From recombination of genes to the estimation of distributions i. binary parameters," in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, London, UK, 1996, pp. 178–187, Springer-Verlag.
- [14] R. Storn and K. Price, "Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep., International Computer Science Institute, 1995.
- [15] Hussein A. Abbass, "An evolutionary artificial neural network approach for breast cancer diagnosis," *Artificial Intelligence in Medicine*, vol. 25, pp. 265–281, 2002.
- [16] P. Sheth and B. V. Babu, "Optimization of kinetic parameters in pyrolysis of biomass using differential evolution," in *Proc. of International Conference on Neural Network and Genetic Algorithm in Materials Science and Engineering*, January 2008.
- [17] Salomon M., Perrin G.R., and Heitz F., "Differential evolution for medical image registration," in *International Conference on Artificial Intelligence*, June 2001.
- [18] R. Storn, "Differential evolution design of an iir-filter," in *IEEE International Conference on Evolutionary Computation ICEC'96*, 1996, pp. 268–273, IEEE Press.
- [19] E. Alba and J. M. Troya, "Analyzing synchronous and asynchronous parallel distributed genetic algorithms," *Future Generation Computer Systems*, vol. 17, no. 4, pp. 451–465, January 2000.
- [20] G. Rudolph, "On takeover times in spatially structured populations: array and ring," in *Proceedings of the Second Asia-Pacific Conference on Genetic Algorithms and Applications*, 2000, pp. 144–151.

- [21] L. Shyh-Chang, Iii, and E. D. Goodman, "Coarse-grain parallel genetic algorithms: categorization and new approach," in *Parallel and Distributed Processing, 1994. Proceedings.*, 1994, pp. 28–37.
- [22] E. Cantu-Paz, *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publishers, 2000.
- [23] P. Adamidis and V. Petridis, "Co-operating populations with different evolution behaviours," 1996.
- [24] T. Hiroyasu, M. Miki, and M. Negami, "Distributed genetic algorithms with randomized migration rate," in *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, 1999, vol. 1, pp. 689–694 vol.1.
- [25] D. Schlierkamp-Voosen and H. Mühlenbein, "Strategy adaptation by competing subpopulations," *Parallel Problem Solving from Nature ? PPSN III*, pp. 199–208, 1994.
- [26] E. Takashima, Y. Murata, N. Shibata, and M. Ito, "Self adaptive island ga," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, 2003, vol. 2, pp. 1072–1079 Vol.2.
- [27] Z. Skolicki and K. De Jong, "Improving evolutionary algorithms with multi-representation island models," in *Proceedings of 8th International Conference on Parallel Problem Solving from Nature, 2004*.
- [28] S. Tsutsui and Y. Fujimoto, "Forking genetic algorithm with blocking and shrinking modes (fga)," in *Proceedings of the 5th International Conference on Genetic Algorithms*, San Francisco, CA, USA, 1993, pp. 206–215.
- [29] V. Bachelet, P. Preux, and E-G, "Parallel hybrid meta-heuristics: Application to the quadratic assignment problem," in *Proceedings of the Parallel Optimization Colloquium*, 1996, pp. 233–242.
- [30] R. Venkateswaran and C. S. Raghavendra, "Cooperative genetic algorithm for optimization problems in distributed computer systems," in *Proc. of the Second Online Workshop on Evolutionary Computation*, 1996, pp. 49–52.
- [31] F. Herrera, M. Lozano, and A. M. Sánchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *International Journal of Intelligent Systems*, vol. 18, no. 3, pp. 309–338, 2003.
- [32] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *The 2005 IEEE Congress on Evolutionary Computation.*, 2005, vol. 1, pp. 506–513 Vol.1.
- [33] P. 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.
- [34] A. Auger and N. Hansen, "A restart cma evolution strategy with increasing population size," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 2005, vol. 2, pp. 1769–1776 Vol. 2.
- [35] A. Sinha, S. Tiwari, and K. Deb, "A population-based, steady-state procedure for real-parameter optimization," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 2005, vol. 1, pp. 514–521 Vol.1.

