Tests no paramétricos de comparaciones múltiples con algoritmo de control en el análisis de algoritmos evolutivos: Un caso de estudio con los resultados de la sesión especial en optimización continua CEC'2005

Salvador García
Dept.de Ciencias Computación e Inteligencia Artificial
ETS Ingeniería Informática
Univ. de Granada
18071 Granada
salvagl@decsai.ugr.es

Daniel Molina
Dept. de Informática
Univ. de Cádiz
11510 Cádiz
daniel.molina@uca.es

Manuel Lozano, Francisco Herrera
Dept. de Ciencias de la Computación e Inteligencia Artificial
ETS Ingeniería Informática
Univ. de Granada
18071 Granada
lozano@decsai.ugr.es, herrera@decsai.ugr.es

Resumen

El análisis de experimentos en el campo de los algoritmos evolutivos es un aspecto de crucial importancia en los avances y propuestas de nuevas técnicas. El interés en llevar a cabo una rigurosa metodología de comparación estadística entre algoritmos es cada vez más solicitado entre la comunidad científica. Por simplicidad y ausencia de condiciones iniciales en su aplicación, las técnicas estadísticas no paramétricas han demostrado ser una buena herramienta para comparar algoritmos. Sin embargo, en comparaciones múltiples de algoritmos, los tests estadísticos más conocidos son demasiado conservativos e impiden detectar muchas diferencias entre resultados que realmente están patentes.

En este trabajo nos centramos en introducir las técnicas de comparaciones más potentes que se pueden utilizar en estadística no paramétrica, y mostrar un caso de uso en donde utilizamos los resultados proporcionados en la Sesión Especial de Optimización Continua organizada en el IEEE CEC'2005, mostrando el incremento de potencia de dichas técnicas con respecto a las clásicas.

1. Introducción

En los últimos años ha habido un creciente número de propuestas de nuevos algoritmos evolutivos (AEs) en el ámbito de optimización continua. Todas ellas requieren una validación que se consigue con respecto a un estudio empírico que considere técnicas anteriores comparadas con la nueva propuesta.

Por este motivo, el interés por parte de la comunidad científica de AEs en disponer de una metodología en el análisis de experimentos también aumenta y se busca un procedimiento general para validar los resultados obtenidos de una experimentación y condicionar las conclusiones del trabajo a los mismos.

En la literatura especializada, podemos encontrar propuestas para utilizar tests estadísticos en el análisis de experimentos [4]. La mayoría de los trabajos validan resultados a partir de estadísticos básicos, como medias y varianzas, junto con técnicas estadísticas paramétricas [5, 14]. Sin embargo, es bien conocido que las técnicas estadísticas paramétricas requieren una serie de condiciones sobre la muestra de resultados que no siempre se cumplen: independencia, normalidad y homogeneidad de varianzas [25, 20]. En [6] se lleva a cabo un estudio que involucra varios AEs en donde se muestra que los resultados obtenidos no cumplen dichas propiedades. Además, se propone el uso de técnicas estadísticas no paramétricas para analizar resultados.

Sin embargo, en comparaciones múltiples que consideran un algoritmo control, los procedimientos clásicos (como el de Bonferroni) son demasiado conservativos y no permiten detectar todas las diferencias existentes en una comparación múltiple [19].

En este trabajo nos centramos en introducir las técnicas de comparaciones más potentes que se pueden utilizar en estadística no paramétrica, y mostrar un caso de uso en donde utilizamos los resultados proporcionados en la Sesión Especial de Optimización Continua organizada en el IEEE CEC'2005, mostrando el incremento de potencia de dichas técnicas con respecto a las clásicas. Los procedimientos que vamos a usar son: Bonferroni [20, 25], Šidák [21], Holm [10], Holland-Copenhaver [9], Hochberg [8], Hommel [11] y Rom [17].

Para ello, el trabajo se organiza de la siguiente forma. En la Sección 2 se describen las características de la Sesión Especial de Optimización Continua del IEEE CEC'2005. La Sección 3 introduce la estadística no paramétrica y enumera y describe los procedimientos de comparaciones múltiples , basados en un método control, utilizados en el trabajo. El caso de uso experimental que utiliza estas técnicas lo encontramos en la Sección 4. Finalmente, la Sección 5 concluye el trabajo.

2. Preliminares: Sesión Especial de Optimización Continua organizada en el IEEE CEC'2005

En esta sección describimos brevemente los algoritmos que se utilizaron en la Sesión Especial de Optimización Continua del CEC'2005, las funciones de test consideradas y las características de la experimentación que hemos tomado como referencias.

2.1. Algoritmos evolutivos

En la mencionada competición, distintos algoritmos se enfrentaron a las 25 funciones de test para obtener la solución más cercana al óptimo. En concreto fueron 11 algoritmos, pertenecientes a diferentes paradigmas de computación evolutiva: algoritmos meméticos, algoritmos de estimación de distribuciones, optimización basada en nubes de partículas, evolución diferencial, etc... Los acrónimos y referencias de todos son: *BLX-GL50* [7], *BLX-MA* [13], *CoEVO* [15], *DE* [18], *DMS-L-PSO* [12], *EDA* [24], *G-CMA-ES* [1], *K-PCX* [22], *L-CMA-ES* [2], *L-SaDE* [16] y *SPC-PNX* [3].

2.2. Funciones de test

Se puede consultar en [23] la descripción completa de las funciones, además en el enlace se incluye el código fuente. El conjunto de funciones de test está compuesto por las siguientes funciones:

5 Funciones Unimodales

- Función Esfera desplazada.
- Problema 1.2 de Schwefel desplazado.
- Función Elíptica rotada ampliamente condicionada.
- Problema desplazado Schwefel 1.2 con ruido en el Fitness.
- Problema de Schwefel 2.6 con el óptimo global en la frontera.

20 Funciones Multimodales

- 7 Funciones básicas
    - Función Rosenbrock desplazada.
    - Función Griewank desplazada y rotada sin fronteras.
    - Función Ackley desplazada y rotada con óptimo global en la frontera.
    - Función Rastrigin desplazada.
    - Función Rastrigin desplazada y rotada.
    - Función Wieierstrass desplazada y rotada.
    - Problema 2.12 de Schwefel.
- 2 Funciones Expandidas.

- 11 Funciones híbridas. Cada una de éstas se han definido mediante composición de 10 de las 14 funciones anteriores (distintas en cada caso).

Todas las funciones han sido desplazadas para segurar que nunca se encuentre su óptimo en el centro del espacio de búsqueda. En dos funciones, además, el óptimo no se encuentra dentro del rango de inicialización, y el dominio de búsqueda no está limitado (el óptimo se encuentra fuera del rango de inicialización).

2.3. Características de la experimentación

Las principales características de la experimentación de esta competición son:

- Cada algoritmo se ejecuta 25 veces para cada función de test, y se calcula la media de error del mejor individuo de la población.

- Se ha realizado el estudio con dimensión $D = 10$ y los algoritmos realizan 100000 evaluaciones de la función. En la competición mencionada se realizaron igualmente experimentos con dimensión $D = 30$ y $D = 50$.

- Cada ejecución termina o bien cuando el error obtenido es menor que 10^{-8}, o cuando se alcanza el número máximo de evaluaciones.

3. Procedimientos de comparaciones múltiples aplicables a la estadística no paramétrica

En la literatura especializada, existen numerosas técnicas de comparaciones múltiples, como por ejemplo, los procedimientos de Tukey, Scheffe, Dunnet, etc... [20, 25]. Sin embargo, la mayoría de ellos requieren un diseño ANOVA anterior a su aplicación. La familia de procedimientos derivados del test de Bonferroni no requieren dicha condición, por lo que pueden ser aplicados en un conjunto de hipótesis cualquiera; en nuestro caso, el obtenido por el test de Friedman. Dentro de la familia de procedimientos Bonferroni, existen tres tipos de métodos: directos, incrementales y decrementales.

En esta sección, introduciremos las técnicas estadísticas no paramétricas que utilizaremos en el estudio. Después, describiremos todos los procedimientos de la familia Bonferroni desde el más conservador hasta el más potente [19].

3.1. Técnicas estadísticas no paramétricas

En esta sección introduciremos brevemente los tests no paramétricos más conocidos.

Para diferenciar a un test no paramétrico del paramétrico hay que comprobar el tipo de datos que el test utiliza. Un test no paramétrico es aquel que utiliza datos de tipo nominal o datos ordinales o que representan un orden estableciendo un ranking. Esto no implica que solamente deban ser usados para ese tipo de datos [20]. Podría ser interesante transformar los datos reales a datos basados en orden, de tal forma que se pueda aplicar un test no paramétrico sobre datos idóneos en tests paramétricos cuando éstos no cumplen las condiciones necesarias. Como normal general, un test no paramétrico es menos restrictivo que un paramétrico, aunque menos robusto que un paramétrico que se realiza sobre datos que cumplen todas las condiciones necesarias [25].

En [6] se muestra como las condiciones necesarias para aplicar tests paramétricos no se cumplen en la aplicación de AEs para optimización continua. Además, se introducen los tests no paramétricos más conocidos, como el test de Wilcoxon, el test de Friedman y el test de Bonferroni-Dunn [20, 25].

En este trabajo, nos interesa abordar las comparaciones múltiples, de tal forma que un algoritmo se pueda comparar con 2 o más algoritmos simultáneamente mientras que el nivel de confianza estadístico es previamente definido. Un test de comparaciones en pareja, como el test de Wilcoxon, se puede emplear también con este propósito, pero el error en familia no se controla y se puede desbordar (ver [6]).

Los test de comparaciones múltiples se basan en analizar un conjunto de hipótesis (familia de hipótesis, compuesta por m hipótesis) y rechazar el mayor número posible de ellas controlando que no se produzcan errores de Tipo I [20]. Para nuestro estudio, suponemos que tenemos M algoritmos y N funciones. Necesitamos usar el test de Friedman para obtener un valor de ranking medio R_i para el algoritmo i sobre todas las funciones. El estadístico para poder comparar el resultado obtenido por dos algoritmos se calcula mediante la expresión

$$z = (R_i - R_j)\sqrt{\frac{M(M+1)}{6N}}$$

El valor de z de cada hipótesis se puede utilizar para calcular la probabilidad de error (valor p) mediante la distribución normal [20]. En total, dispondremos de una familia de hipótesis, cada una asociada a su correspondiente valor p.

3.2. Procedimiento original de Bonferroni

Es un procedimiento directo que ajusta el nivel de confianza $\alpha' = \alpha/m$. Se rechaza una hipótesis individual H_i con $p_i < \alpha'$.

3.3. Procedimiento de Šidák

Es un procedimiento directo que ajusta el nivel de confianza a $\alpha' = 1-(1-\alpha)^{1/m}$ [21]. Se rechaza una hipótesis individual H_i con $p_i < \alpha'$. Este procedimiento es más potente que el de Bonferroni, pero necesita que los test estadísticos sean ortogonalmente dependientes, propiedad que se verifica en el caso de tests de dos colas (two-tailed) que emplean la distribución normal, como es nuestro caso.

3.4. Procedimiento de Holm

Es un procedimiento incremental que define diferentes niveles de rechazo por hipótesis [10]. Sea $p_1, ..., p_m$ los valores p ordenados (del más pequeño al más grande) y $H_i, ..., H_m$ las correspondientes hipótesis, el procedimiento de Holm rechaza H_1 hasta $H_{(i-1)}$ si i es el entero más pequeño tal que $p_i > \alpha/(m-i+1)$. Este procedimiento incrementa significativamente la potencia del de Bonferroni.

3.5. Procedimiento de Holland-Copenhaver

Holland y Copenhaver en [9] recomiendan usar la misma técnica que el procedimiento de Holm, pero usando la desigualdad de Šidák. Siguiendo el mismo esquema que Holm, el procedimiento de Holland-Copenhaver rechaza H_1 hasta $H_{(i-1)}$ si i es el entero más pequeño tal que $p_i > 1-(1-\alpha)^{1/(m-i+1)}$. Este procedimiento es más potente que el de Holm, pero solo es aplicable bajo las mismas condiciones que el de Šidák.

3.6. Procedimiento de Hochberg

Hochberg [8] desarrolló un procedimiento decremental que consiste en utilizar la misma ordenación y niveles $\alpha'_1, ..., \alpha'_m$ ajustados que Holm. Si $p_{(m-1)}$ es menor que $\alpha'_{(m-1)}$, entonces todas las hipótesis restantes son rechazadas, si no $H_{(m-1)}$ es aceptada, y $p_{(m-2)}$ es comparada con $\alpha'_{(m-2)}$; así sucesivamente. Es más potente que el procedimiento de Holm, pero requiere que los test estadísticos sean independientes.

3.7. Procedimiento de Hommel

Es un procedimiento decremental que también requiere la independencia de los tests estadísticos [11]. Determina los valores críticos ajustados en dos fases. Sea $J = \{i' \in \{1, ..., m\} : p_{(m-i'+k)} > k\alpha/i'; k = 1, ..., i'\}$. La primera fase del procedimiento utiliza los valores p obtenidos para determinar el número de miembros en J. La segunda fase obtiene el nivel de significancia de rechazo usando $\alpha' = \alpha/j'$, donde j' es el número más grande en J. Si J está vacío, todas las $H_i (i = 1, ..., m)$ son rechazadas. Si J no está vacío, H_i es rechazado siempre que $p_i \leq \alpha/j'$. Este procedimiento es más potente que el de Hochberg.

3.8. Procedimiento de Rom

Rom en [17] desarrolló una modificación al procedimiento de Hochberg para incrementar su potencia. El incremento se consigue identificando los niveles de significancia apropiados que controlan los errores de Tipo I cuando los test estadísticos son independientes. Funciona exactamente igual que el procedimiento de Hochberg, pero los valores α' se calculan mediante la expresión

$$\alpha'_{m-i+1} = \left[\sum_{j=1}^{i-1} \alpha^j - \sum_{j=1}^{i-2} \binom{i}{j} \alpha'^{(i-j)}_{(m-j)} \right] / i,$$

en donde $\alpha'_m = \alpha$ y $\alpha'_{(m-1)} = \alpha/2$.

4. Caso de estudio: Sesión Especial de Optimización Continua del CEC'2005

Esta sección desarrolla la aplicación de las técnicas estadísticas antes descritas con un caso de estudio de los resultados publicados en la Sesión Especial de Optimización Continua organizada en el congreso IEEE CEC'2005.

Como mencionamos antes, nos centraremos en la dimensión $D = 10$. Hemos considerado usar la tasa de error obtenida por cada algoritmo en la competición. La tabla de resultados completa puede consulsarte en el apéndice, Tabla 3. Para este estudio, los tests no paramétricos solo requieren el error medio obtenido por cada algoritmo en cada función. El estudio se centrará en el algoritmo que obtuvo el menor error medio en la competición, *G-CMA-ES* [1]. Estudiaremos el rendimiento de este algoritmo con respecto al resto, y determinaremos si los resultados que ofrece son mejores que los ofrecidos por los demás, fijado un nivel de confianza. De esta forma, los tests estadísticos resultantes son independientes, lo que permite aplicar sin problema todas las técnicas estadísticas presentadas en la Sección 4. Consideraremos un nivel de confianza $\alpha = 0{,}10$.

La Tabla 1 muestra los rankings obtenidos por cada algoritmo utilizando el test de Friedman. Este test es bien conocido, y está integrado en los principales paquetes de software estadístico: SPSS, R, SAS, etc... El proceso es también explicado en [6].

Mediante el procedimiento descrito en la Sección 3, se calcula la diferencia de rankings entre el algoritmo *G-CMA-ES* con respecto al resto, obteniendo el valor z de cada comparación. Con él, calculamos el valor p_i, $i = 1, ..., m$, para cada una de las m hipótesis.

Algoritmo	Ranking
BLX-GL50	5.30
BLX-MA	7.14
CoEVO	8.44
DE	5.66
DMS-L-PSO	5.02
EDA	6.74
G-CMA-ES	3.34
K-PCX	6.80
L-CMA-ES	6.22
L-SaDE	4.92
SPC-PNX	6.42

Tabla 1: Rankings de los algoritmos obtenidos mediante el test de Friedman

La Tabla 2 contiene el cálculo de los valores p_i correspondientes a cada hipótesis, junto a los valores α' ajustados por cada procedimiento de comparaciones múltiples utilizado, en orden incremental con respecto al valor de p_i.

Conociendo el funcionamiento de los 7 procedimientos estadísticos de comparaciones múltiples descritos en este trabajo, podemos señalar lo siguiente:

- Bonferroni rechaza las hipótesis con un $p < 0{,}01$, esto es, *G-CMA-ES* es mejor que *CoEVO*, *BLX-MA*, *K-PCX*, *EDA*, *SPC-PNX* y *L-CMA-ES* (6/10 algoritmos).
- Šidák rechaza las mismas hipótesis que Bonferroni (6/10 algoritmos).
- Holm rechaza una hipótesis más que Bonferroni, la comparación con respecto a *DE* (7/10 algoritmos).
- Holland-Copenhaver rechaza las mismas hipótesis que Holm (7/10 algoritmos).
- Hochberg es capaz de rechazar todas las hipótesis, puesto que $p_{10} < 0{,}10$ (10/10 algoritmos).
- Hommel también rechaza todas las hipótesis (10/10 algoritmos).

i	G-CMA-ES vs.	z	p	Bonferroni	Šidák	Holm/Hommel Hochberg	Holland	Rom
1	CoEVO	5,4366	$5{,}4301 \cdot 10^{-8}$	0,01	0,0105	0,01	0,0105	0,0111
2	BLX-MA	4,0508	$5{,}104 \cdot 10^{-5}$	0,01	0,0105	0,0111	0,0116	0,0123
3	K-PCX	3,6884	$2{,}2569 \cdot 10^{-4}$	0,01	0,0105	0,0125	0,0131	0,0138
4	EDA	3,6244	$2{,}8962 \cdot 10^{-4}$	0,01	0,0105	0,0143	0,0149	0,0158
5	SPC-PNX	3,2833	0,0010	0,01	0,0105	0,0167	0,0174	0,0184
6	L-CMA-ES	3,0701	0,0021	0,01	0,0105	0,02	0,0209	0,0221
7	DE	2,4731	0,0134	0,01	0,0105	0,025	0,026	0,0274
8	BLX-GL50	2,0894	0,0367	0,01	0,0105	0,0333	0,0345	0,0333
9	DMS-L-PSO	1,7909	0,0733	0,01	0,0105	0,05	0,0513	0,05
10	L-SaDE	1,6843	0,0921	0,01	0,0105	0,1	0,1	0,1

Tabla 2: Valores p_i y α' para Bonferroni, Šidák, Holm, Holland-Copenhaver, Hochberg, Hommel, Holland y Rom; ordenados (consideramos $\alpha = 0{,}10$)

- Rom también rechaza todas las hipótesis (10/10 algoritmos).

Como vemos, los procedimientos de comparaciones múltiples más potentes son capaces de distinguir como mejor el método *G-CMA-ES* del resto. Podríamos destacar, como norma general, los siguientes aspectos en relación a estos procedimientos:

- La mejora que aporta el procedimiento de Šidák no es demasiado significativa con respecto al procedimiento Bonferroni.
- El procedimiento de Holm es menos conservativo que el Bonferroni y no hace ninguna suposición adicional sobre su uso. Además, tampoco se trata de un procedimiento muy complejo, por lo que nosotros recomendamos su uso.
- Al igual que sucede con el procedimiento de Šidák, la mejora que se consigue con el procedimiento de Holland-Copenhaver con respecto al de Holm es pequeña.
- En el caso de tests independientes (por ejemplo cuando comparamos los resultados considerando un algoritmo control), Se pueden utilizar los procedimientos de Hochberg, Hommel y Rom.
- Las mejoras de los procedimientos de Hommel y Rom son poco significativas con respecto al procedimiento de Hochberg. En muchas ocasiones, la dificultad de llevar a cabo uno de estos dos procedimientos no merece la pena para obtener una ligera diferencia con respecto a Hochberg.
- Nosotros recomendamos usar el procedimiento de Hochberg por ser igual de sencillo de aplicar que el procedimiento de Holm. En el caso de que algún p_i de alguna hipótesis esté muy cerca del correspondiente α' y no se rechace, podríamos intentar aplicar uno de los dos procedimientos más complejos (Hommel o Rom).

5. Conclusiones

En este trabajo hemos introducido algunos de los procedimientos de comparaciones múltiples, que consideran un algoritmo control, aplicables a la estadística no paramétrica, desde los más conocidos a los más complejos, y los hemos utilizado en un caso de estudio que comprende los resultados de la Sesión Especial en Optimización Continua organizada por el congreso IEEE CEC'2005.

De entre todos los procedimientos, recomendamos el procedimiento de Holm y el de Hochberg, considerando un equilibrio entre potencia y simplicidad en su aplicación. Sin embargo, en cirscunstancias idóneas, un procedimiento más potente podría ser capaz de detectar un mayor número de diferencias entre algoritmos.

Agradecimientos

Este trabajo ha sido financiado por el MCYT a través del proyecto TIN2005-08386-C05-01.

Referencias

[1] Auger, A., Hansen, N., A restart CME evolution strategy with increasing population size, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1769-1776.

[2] Auger, A., Hansen, N., Performance evaluation of an advanced local search evolutionary algorithm, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1777-1874.

[3] Ballester, P.J., Stephenson, J., Carter, J.N., Gallagher, K., Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 498-505.

[4] Bartz-Beielstein, T., *Experimental research in evolutionary computation: The new experimentalism.* Springer-Verlag, 2006.

[5] Czarn, A., MacNish, C., Vijatan, K., Turlach, B., Gupta, R., Statistical exploratory analysis of genetic algorithms, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 4, 2004, pp. 405-421.

[6] García, S., Molina, D., Lozano, M., Herrera, F., Un estudio experimental sobre el uso de test no paramétricos para analizar el comportamiento de los algoritmos evolutivos en problemas de optimización, en *Actas del Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, (MAEB07)*, 2007, pp.275-285.

[7] García-Martínez, C., Lozano, M., Hybrid real-coded genetic algorithms with female and male differentiation, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 896-903.

[8] Hochberg, Y., A sharper Bonferroni procedure for multiple tests of significance, *Biometrika*, Vol. 75, 1998, pp. 811-818.

[9] Holland, B.S., Copenhaver, M.D., An improved sequentially rejective Bonferroni test procedure, *Biometrics*, Vol. 43, 1987, 417-423.

[10] Holm, S., A simple sequentially rejective multiple test procedure, *Scandinavian Journal of Statistics*, Vol. 6, 1979, pp. 65-70.

[11] Hommel, G., A stagewise rejective multiple test procedure based on a modified Bonferroni test, *Biometrika*, Vol. 75, 1988, pp. 383-386.

[12] Liang, J.J., Suganthan, P.N., Dynamic multi-swarm particle swarm optimizer with local search, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 522-528.

[13] Molina, D., Lozano, M., Herrera, F., Adaptive local search parameters for real-coded memetic algorithms, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 888-895.

[14] Mori, N., Takeda, M., Matsumoto, K., A statistical comparison study between Genetic Algorithms and Bayesian Optimization Algorithms, *Progress of Theoretical Physics Supplement*, Vol. 157, 2005, pp. 353-356.

[15] Pošík, P., Real-parameter optimization using the mutation step co-evolution ,en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 872-879.

[16] Qin, A.K., Suganthan, P.N., Self-adaptive differential evolution algorithm for numerical optimization, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1785-1791.

[17] Rom, D.M., A sequentially rejective test procedure based on a modified Bonferroni inequality, *Biometrika*, Vol. 77, 1990, pp. 663-665.

[18] Rônkkônen, J., Kukkonen, S., Price, K.V., Real-Parameter optimization with

differential evolution, in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 506-513.

[19] Shaffer, J.P., Multiple hypothesis testing, *Annual Review of Psychology*, Vol. 46, 1995, pp. 561-584.

[20] Sheskin, D.J., *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press. 2003.

[21] Šidák, Z., Rectangular confidence regions for the means of multivariate normal distributions, *Journal of the American Statistical Association*, Vol. 62, 1967, pp. 623-633.

[22] Sinha, A., Tiwari, S., Deb, K., A population-based, steady-steate procedure for real-parameter optimization, en *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 514-521.

[23] Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S., Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real Parameter Optimization. Technical Report. Nanyang Technological University. Mayo 2005.

[24] Yuang, B., Gallagher, M., Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA, in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1792-1799.

[25] Zar, J.H., *Biostatistical Analysis*. Prentice Hall. 1999.

Apéndice: Resultados de error obtenidos por los algoritmos en cada función

Algoritmo	f1	f2	f3	f4	f5	f6	f7
BLX-GL50	10^{-9}	10^{-9}	$5{,}705 \cdot 10^{2}$	10^{-9}	10^{-9}	10^{-9}	$1{,}172 \cdot 10^{-2}$
BLX-MA	10^{-9}	10^{-9}	$4{,}771 \cdot 10^{4}$	$1{,}997 \cdot 10^{-8}$	$2{,}124 \cdot 10^{-2}$	1,49	$1{,}971 \cdot 10^{-1}$
CoEVO	10^{-9}	10^{-9}	10^{-9}	10^{-9}	2,133	$1{,}246 \cdot 10$	$3{,}705 \cdot 10^{-2}$
DE	10^{-9}	10^{-9}	$1{,}94 \cdot 10^{-6}$	10^{-9}	10^{-9}	$1{,}59 \cdot 10^{-1}$	$1{,}46 \cdot 10^{-1}$
DMS-L-PSO	10^{-9}	10^{-9}	10^{-9}	$1{,}885 \cdot 10^{-3}$	$1{,}138 \cdot 10^{-6}$	$6{,}892 \cdot 10^{-8}$	$4{,}519 \cdot 10^{-2}$
EDA	10^{-9}	10^{-9}	$2{,}121 \cdot 10$	10^{-9}	10^{-9}	$4{,}182 \cdot 10^{-2}$	$4{,}205 \cdot 10^{-1}$
G-CMA-ES	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-9}
K-PCX	10^{-9}	10^{-9}	$4{,}15 \cdot 10^{-1}$	$7{,}94 \cdot 10^{-7}$	$4{,}85 \cdot 10$	$4{,}78 \cdot 10^{-1}$	$2{,}31 \cdot 10^{-1}$
L-CMA-ES	10^{-9}	10^{-9}	10^{-9}	$1{,}76 \cdot 10^{6}$	10^{-9}	10^{-9}	10^{-9}
L-SaDE	10^{-9}	10^{-9}	$1{,}672 \cdot 10^{-2}$	$1{,}418 \cdot 10^{-5}$	0,012	$1{,}199 \cdot 10^{-8}$	0,02
SPC-PNX	10^{-9}	10^{-9}	$1{,}081 \cdot 10^{5}$	10^{-9}	10^{-9}	$1{,}891 \cdot 10$	$8{,}261 \cdot 10^{-2}$

Algoritmo	f8	f9	f10	f11	f12	f13	f14
BLX-GL50	$2{,}035 \cdot 10$	1,154	4,975	2,334	$4{,}069 \cdot 10^{2}$	$7{,}498 \cdot 10^{-1}$	2,172
BLX-MA	$2{,}019 \cdot 10$	$4{,}379 \cdot 10^{-1}$	5,643	4,557	$7{,}43 \cdot 10$	$7{,}736 \cdot 10^{-1}$	2,03
CoEVO	$2{,}027 \cdot 10$	$1{,}919 \cdot 10$	$2{,}677 \cdot 10$	9,029	$6{,}046 \cdot 10^{2}$	1,137	3,706
DE	$2{,}04 \cdot 10$	$9{,}55 \cdot 10^{-1}$	$1{,}25 \cdot 10$	$8{,}47 \cdot 10^{-1}$	$3{,}17 \cdot 10$	$9{,}77 \cdot 10^{-1}$	3,45
DMS-L-PSO	$2 \cdot 10$	10^{-9}	3,622	4,623	2,4001	$3{,}689 \cdot 10^{-1}$	2,36
EDA	$2{,}034 \cdot 10$	5,418	5,289	3,944	$4{,}423 \cdot 10^{2}$	1,841	2,63
G-CMA-ES	$2 \cdot 10$	$2{,}39 \cdot 10^{-1}$	$7{,}96 \cdot 10^{-2}$	$9{,}34 \cdot 10^{-1}$	$2{,}93 \cdot 10$	$6{,}96 \cdot 10^{-1}$	3,01
K-PCX	$2 \cdot 10$	$1{,}19 \cdot 10^{-1}$	$2{,}39 \cdot 10^{-1}$	6,65	$1{,}49 \cdot 10^{2}$	$6{,}53 \cdot 10^{-1}$	2,35
L-CMA-ES	$2 \cdot 10$	$4{,}49 \cdot 10$	$4{,}08 \cdot 10$	3,65	$2{,}09 \cdot 10^{2}$	$4{,}94 \cdot 10^{-1}$	4,01
L-SaDE	$2 \cdot 10$	10^{-9}	4,969	4,891	$4{,}501 \cdot 10^{-7}$	0,22	2,915
SPC-PNX	$2{,}099 \cdot 10$	4,02	7,304	1,91	$2{,}595 \cdot 10^{2}$	$8{,}379 \cdot 10^{-1}$	3,046

Algoritmo	f15	f16	f17	f18	f19	f20	f21
BLX-GL50	$4 \cdot 10^{2}$	$9{,}349 \cdot 10$	$1{,}09 \cdot 10^{2}$	$4{,}2 \cdot 10^{2}$	$4{,}49 \cdot 10^{2}$	$4{,}46 \cdot 10^{2}$	$6{,}893 \cdot 10^{2}$
BLX-MA	$2{,}696 \cdot 10^{2}$	$1{,}016 \cdot 10^{2}$	$1{,}27 \cdot 10^{2}$	$8{,}033 \cdot 10^{2}$	$7{,}628 \cdot 10^{2}$	$8 \cdot 10^{2}$	$7{,}218 \cdot 10^{2}$
CoEVO	$2{,}938 \cdot 10^{2}$	$1{,}772 \cdot 10^{2}$	$2{,}118 \cdot 10^{2}$	$9{,}014 \cdot 10^{2}$	$8{,}445 \cdot 10^{2}$	$8{,}629 \cdot 10^{2}$	$6{,}349 \cdot 10^{2}$
DE	$2{,}59 \cdot 10^{2}$	$1{,}13 \cdot 10^{2}$	$1{,}15 \cdot 10^{2}$	$4 \cdot 10^{2}$	$4{,}2 \cdot 10^{2}$	$4{,}6 \cdot 10^{2}$	$4{,}92 \cdot 10^{2}$
DMS-L-PSO	4,854	$9{,}476 \cdot 10$	$1{,}101 \cdot 10^{2}$	$7{,}607 \cdot 10^{2}$	$7{,}143 \cdot 10^{2}$	$8{,}22 \cdot 10^{2}$	$5{,}36 \cdot 10^{2}$
EDA	$3{,}65 \cdot 10^{2}$	$1{,}439 \cdot 10^{2}$	$1{,}568 \cdot 10^{2}$	$4{,}832 \cdot 10^{2}$	$5{,}644 \cdot 10^{2}$	$6{,}519 \cdot 10^{2}$	$4{,}84 \cdot 10^{2}$
G-CMA-ES	$2{,}28 \cdot 10^{2}$	$9{,}13 \cdot 10$	$1{,}23 \cdot 10^{2}$	$3{,}32 \cdot 10^{2}$	$3{,}26 \cdot 10^{2}$	$3 \cdot 10^{2}$	$5 \cdot 10^{2}$
K-PCX	$5{,}1 \cdot 10^{2}$	$9{,}59 \cdot 10$	$9{,}73 \cdot 10$	$7{,}52 \cdot 10^{2}$	$7{,}51 \cdot 10^{2}$	$8{,}13 \cdot 10^{2}$	$1{,}05 \cdot 10^{3}$
L-CMA-ES	$2{,}11 \cdot 10^{2}$	$1{,}05 \cdot 10^{2}$	$5{,}49 \cdot 10^{2}$	$4{,}97 \cdot 10^{2}$	$5{,}16 \cdot 10^{2}$	$4{,}42 \cdot 10^{2}$	$4{,}04 \cdot 10^{2}$
L-SaDE	32	$1{,}012 \cdot 10^{2}$	$1{,}141 \cdot 10^{2}$	$7{,}194 \cdot 10^{2}$	$7{,}049 \cdot 10^{2}$	$7{,}13 \cdot 10^{2}$	$4{,}64 \cdot 10^{2}$
SPC-PNX	$2{,}538 \cdot 10^{2}$	$1{,}096 \cdot 10^{2}$	$1{,}19 \cdot 10^{2}$	$4{,}396 \cdot 10^{2}$	$3{,}8 \cdot 10^{2}$	$4{,}4 \cdot 10^{2}$	$6{,}801 \cdot 10^{2}$

Algoritmo	f22	f23	f24	f25
BLX-GL50	$7{,}586 \cdot 10^{2}$	$6{,}389 \cdot 10^{2}$	$2 \cdot 10^{2}$	$4{,}036 \cdot 10^{2}$
BLX-MA	$6{,}709 \cdot 10^{2}$	$9{,}267 \cdot 10^{2}$	$2{,}24 \cdot 10^{2}$	$3{,}957 \cdot 10^{2}$
CoEVO	$7{,}789 \cdot 10^{2}$	$8{,}346 \cdot 10^{2}$	$3{,}138 \cdot 10^{2}$	$2{,}573 \cdot 10^{2}$
DE	$7{,}18 \cdot 10^{2}$	$5{,}72 \cdot 10^{2}$	$2 \cdot 10^{2}$	$9{,}23 \cdot 10^{2}$
DMS-L-PSO	$6{,}924 \cdot 10^{2}$	$7{,}303 \cdot 10^{2}$	$2{,}24 \cdot 10^{2}$	$3{,}657 \cdot 10^{2}$
EDA	$7{,}709 \cdot 10^{2}$	$6{,}405 \cdot 10^{2}$	$2 \cdot 10^{2}$	$3{,}73 \cdot 10^{2}$
G-CMA-ES	$7{,}29 \cdot 10^{2}$	$5{,}59 \cdot 10^{2}$	$2 \cdot 10^{2}$	$3{,}74 \cdot 10^{2}$
K-PCX	$6{,}59 \cdot 10^{2}$	$1{,}06 \cdot 10^{3}$	$4{,}06 \cdot 10^{2}$	$4{,}06 \cdot 10^{2}$
L-CMA-ES	$7{,}4 \cdot 10^{2}$	$7{,}91 \cdot 10^{2}$	$8{,}65 \cdot 10^{2}$	$4{,}42 \cdot 10^{2}$
L-SaDE	$7{,}349 \cdot 10^{2}$	$6{,}641 \cdot 10^{2}$	$2 \cdot 10^{2}$	$3{,}759 \cdot 10^{2}$
SPC-PNX	$7{,}493 \cdot 10^{2}$	$5{,}759 \cdot 10^{2}$	$2 \cdot 10^{2}$	$4{,}06 \cdot 10^{2}$

Tabla 3: Error medio obtenido en la competición del IEEE CEC'2005 con dimensión 10