Un estudio experimental sobre el uso de test no paramétricos para analizar el comportamiento de los algoritmos evolutivos en problemas de optimización

S. García[1], D. Molina[2], M. Lozano[3], F. Herrera[4]

Resumen— **En los últimos años existe un creciente interés por el análisis de experimentos en el ámbito de los algoritmos evolutivos y las metaheurísticas. Este interés queda patente por la publicación continua de trabajos que analizan y proponen diferentes tipos de problemas como base de comparación experimental de algoritmos, la propuesta de diferentes metodologías de comparación o las propuestas de uso de diferentes técnicas estadísticas para la comparación de algoritmos.**

En este trabajo nos centramos en el uso de técnicas estadísticas para el análisis del comportamiento de los algoritmos evolutivos en problemas de optimización. Se presenta un estudio sobre el uso de test no paramétricos para el análisis de resultados utilizando algunos modelos de algoritmos genéticos para la optimización de funciones continuas. Mostramos resultados donde queda patente la necesidad de utilizar estadística no paramétrica dado que los algoritmos genéticos utilizados no verifican las hipótesis de partida necesarias para el uso de tests paramétricos.

Palabras clave— **Análisis estadístico de experimentos, algoritmos evolutivos, test paramétricos, test no paramétricos.**

I. INTRODUCCIÓN

A partir del teorema de "No free lunch" [25] se sabe que no se puede encontrar ningún algoritmo metaheurístico que sea el mejor en comportamiento para cualquier problema. Por otra parte sabemos que podemos trabajar con diferentes grados de conocimiento sobre el problema que pretendemos resolver, y que no es lo mismo trabajar sin ningún conocimiento (hipótesis del teorema de "no free lunch") que trabajar con un conocimiento parcial del problema, conocimiento que nos permite el diseño de algoritmos con características específicas que los puedan hacer adecuados para su resolución.

Situados en este ámbito, el conocimiento parcial del problema y la necesidad de disponer de algoritmos para su resolución, se plantea la cuestión de decidir cuando un algoritmo es mejor que otro. En el caso del uso de metaheurísticas o algoritmos evolutivos esto lo debemos hacer atendiendo a criterios de eficiencia y/o eficacia. Cuando no se dispone de resultados teóricos que permitan comparar el comportamiento los algoritmos, nos tenemos que centrar en el análisis de los resultados empíricos.

En los últimos años existe un creciente interés por el análisis de experimentos en el ámbito de los algoritmos evolutivos y las metaheurísticas. Este análisis debería evitar una serie de problemas/decisiones que podrían invalidar las conclusiones del estudio. El trabajo de Hooker es pionero en esta línea, y muestra un interesante estudio acerca de lo que debemos hacer y no hacer cuando nos planteamos el análisis del comportamiento de una metaheurística sobre un problema [12].

En cuanto al diseño de experimentos, podemos encontrar dos tipos de trabajos, el estudio y diseño de problemas de test y el análisis estadístico de experimentos:

- Diferentes autores han centrado su interés en el diseño de problemas de test que sean adecuados para realizar un estudio comparativo entre algoritmos. Centrándonos en los problemas de optimización continua que utilizaremos en este trabajo, podemos señalar los trabajos pioneros de Whitley y coautores para el diseño de funciones de test complejas para optimización continua [23,24], y los trabajos recientes de Gallagher y Yuan [9,26]. De igual forma podemos encontrar trabajos que analizan casos de test para diferentes tipos de problemas.
- Centrados en el análisis estadístico de los resultados, si analizamos los trabajos publicados en revistas especializadas nos encontramos que la mayoría de los artículos realizan una comparación de resultados basada en el valor medio de un conjunto de ejecuciones

[1] Dpto. de Ciencias de la Computación e I.A., Universidad de Granada, 18071-Granada E-mail: salvagl@decsai.ugr.es
[2] Dpto. de Informática, Universidad de Cádiz, Granada E-mail: dmolina@decsai.ugr.es
[3] Dpto. de Ciencias de la Computación e I.A., Universidad de Granada, 18071-Granada E-mail: lozano@decsai.ugr.es
[4] Dpto. de Ciencias de la Computación e I.A., Universidad de Granada, 18071-Granada E-mail: herrera@decsai.ugr.es

sobre un caso concreto. En proporción, pocos trabajos utilizan técnicas estadísticas para comparar los resultados, aunque recientemente aumenta su uso y está siendo plateando como una necesidad por parte de muchos revisores. Cuando encontramos estudios estadísticos estos suelen estar basados en la media y varianza utilizando test paramétricos (ANOVA, t-test, ...) [4,16,17,18].

Además de estas dos líneas mencionadas, considerando el análisis de experimentos y el uso de técnicas estadísticas, cabe mencionar otras tres líneas de trabajo: a) aportaciones que utilizan las técnicas estadísticas para guiar la búsqueda de los algoritmos evolutivos [8,11,19]; b) los estudios mostrados en [2], donde además del uso de técnicas estadísticas para el análisis de experimentos propone un aprendizaje a partir de error, controlando el error que ocurre durante la experimentación; c) aunque prácticamente la totalidad de los estudios que podemos encontrar en la literatura especializada analizan la eficiencia y la eficacia, medida ésta como el error con respecto al óptimo conocido, existen otras medidas de análisis de los algoritmos evolutivos como la medida de movilidad utilizada en [14], que cuantifica la dispersión de los óptimos locales visitados durante el proceso de búsqueda analizando el comportamiento de los algoritmos a partir de esta medida.

En este trabajo nos centramos en estudiar el uso de las técnicas estadísticas para el análisis del comportamiento de los algoritmos evolutivos en problemas de optimización, analizando el uso de los test estadísticos paramétricos y no paramétricos [20,27]. Analizaremos las condiciones necesarias para el uso de los primeros, y mostraremos resultados utilizando los segundos. Un estudio similar para analizar los algoritmos de aprendizaje automático se puede encontrar en [5].

Para realizar este estudio utilizamos algunos modelos de algoritmos genéticos (AGs) para la optimización de funciones continuas en este ámbito. Mostramos resultados donde queda patente la necesidad de utilizar estadística no paramétrica dado que los AGs utilizados no verifican las hipótesis de partida necesarias para el uso de tests paramétricos.

El trabajo se organiza de la siguiente forma. En la sección II describimos los 4 AGs utilizados en nuestro estudio, y las funciones de test consideradas. La sección III muestra el estudio sobre las hipótesis iniciales necesarias para el uso de los test paramétricos. La sección IV muestra un estudio sobre el uso de test no paramétricos. Las conclusiones finales y los trabajos futuros se muestran en la sección V.

II. PRELIMINARES: ALGORITMOS GENETICOS Y FUNCIONES DE TEST

En esta sección describiremos brevemente los algoritmos utilizados, las funciones de test, y las características de la experimentación.

A. Algoritmos genéticos

En la literatura especializada podemos encontrar diferentes propuestas de AGs para optimización continua. A continuación describimos brevemente los 4 algoritmos utilizados en este estudio.

- AGG: Algoritmo Genético Generacional.
- CHC: Modelo CHC [6,22], que combina diferentes mecanismos para conseguir un buen equilibro entre diversidad y convergencia, como por ejemplo la prevención de incesto o la reinicialización de la población cuando el proceso de búsqueda se estanca.
- AGE-NAM: Algoritmo Genético Estacionario [22] que utiliza un método de selección orientado a escoger padres distantes entre sí llamado NAM [7].
- AGE-WAMS: Algoritmo Genético Estacionario que utiliza un método de reemplazo que mantiene diversidad en la exploración llamado WAMS [3].

Características de CHC:
- Tamaño de la población: 50 individuos.
- Cruce: BLX-0.5.

Características comunes de los AGs:
- Tamaño de la población: 60 individuos.
- Cruce: BLX-0.5.
- Mutación: BGA, aplicada al 12.5% de los genes.

Características propias del AGG:
- Probabilidad de cruce: 0.6
- Selección: Ranking lineal [1]. Se ordenan los individuos de la población por su valor de la función objetivo en orden descendente de mejor a peor. A cada individuo se le asigna un valor de probabilidad de ser elegido padre, en función de la posición que ocupa en dicho listado: Será mayor dicha probabilidad cuanto menor sea su posición. Se seleccionan ambos padres utilizando dichas probabilidades.

Características propias del AGE-NAM.
- Selección: El Emparejamiento Variado Inverso (*Negative Assortative Mating, NAM)* [7]. En

esta selección se escoge un padre aleatoriamente, y para calcular el otro se seleccionan aleatoriamente N_{NAM} individuos de la población, y se escoge el más distante al primero (aplicando una medida de distancia). Está orientado a generar diversidad. En nuestros experimentos utilizamos N_{NAM}=3.

- Reemplazo: RW. Se reemplaza el peor elemento de la población si lo mejora. Ofrece alta presión selectiva, incluso cuando sus padres son elegidos aleatoriamente [10].

Características propias del AGE-WAMS:

- Selección: Selección por Torneo (*Tournament Selection, TS)*. Se muestrea aleatoriamente un grupo de N_{TS} individuos de la población y se selecciona el que posea el mejor valor para la función objetivo. Origina bastante presión selectiva. En nuestros experimentos utilizamos N_{TS}=3.
- Reemplazo: Reemplazar el Peor Entre Semejantes (*Worst Among Most Similar Replacement, WAMS*) [3]. Se compone de los siguientes pasos. Primero, se muestran de la población aleatoriamente C_f grupos de C_s elementos cada uno. Después, se identifica para cada grupo el individuo más similar al descendiente considerado. Este proceso genera C_f individuos como candidatos para ser reemplazados, de los que se selecciona aquel con peor valor de la función objetivo. El descendiente reemplazará a éste si es mejor. En nuestros experimentos utilizamos C_f=6 y C_s = 9.

B. *Funciones de test*

El conjunto de funciones de tests utilizado es el conjunto diseñado para la Sesión Especial de Optimización Continua organizado en el IEEE Congress on Evolutionary Computation de 2005 celebrado en Londres.

Se puede consultar en [21] 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.

2 Funciones unimodales no incluidas en [21] para disponer de 7 funciones unimodales y aplicar tests no paramétricos sobre este conjunto de funciones (unimodales). Las funciones adicionales son:

- Función Zhakarov.
- Función escalonada llamada Step.

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 Weierstrass desplazada y rotada.
  - Problema 2.13 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 asegurar 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).

C. *Características de la experimentación*

Los experimentos han sido realizados siguiendo las instrucciones indicadas en el documento asociado a la competición. Las principales características son:

- Cada algoritmo se ejecuta 50 veces para cada función de test, y se calcula la media del 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 dimension D=30 y D=50.
- Cada ejecución termina o bien cuando el error obtenido es menor que 1e-8, o cuando se alcanza el número máximo de evaluaciones que para esta dimensión es 1e5.

III. ESTUDIO DE LAS CONDICIONES INICIALES PARA EL USO DE TEST PARAMÉTRICOS

En esta sección vamos a analizar las condiciones que se deben cumplir para el uso de los test paramétricos y estudiamos su cumplimiento para el conjunto de funciones y algoritmos utilizados.

A. Condiciones para el uso de los test paramétricos

En [20], la distinción que se hace entre test paramétricos y no parámetricos se basa en el nivel de medida representado por los datos que van a ser analizados. De esta manera, un test paramétrico es aquel que utiliza datos reales pertenecientes a un intervalo.

Esto no implica que siempre que dispongamos de este tipo de datos, haya que usar un test paramétrico. Puede darse el caso de que una o más suposiciones que hacen los test paramétricos se violen, haciendo que el análisis estadística pierda credibilidad.

Para utilizar los test paramétricos es necesario que cumplan las siguientes condiciones [20,27]:

- Independencia: En estadística, dos sucesos son independientes cuando el que haya ocurrido uno de ellos no modifica la probabilidad de ocurrencia del otro.
- Normalidad: Una observación es normal cuando su comportamiento sigue una distribución normal o de Gauss con una determinada media μ y varianza σ. Un test de normalidad sobre una muestra nos indica la presencia o no de esta condición sobre los datos observados. Utilizaremos dos tests de normalidad:
  - Kolmogorov-Smirnov: que compara la distribución acumulada de los datos observados con la distribución acumulada esperada por una distribución Gaussiana, obteniendo el valor de *p* basándose en la discrepancia entre ambas.
  - Shapiro-Wilk: que analiza los datos observados para calcular el nivel de simetría y curtosis (o forma de la curva) para después calcular su diferencia con respecto a los de una distribución Gaussiana, obteniendo el valor de *p* a partir de la suma de cuadrados de esas discrepancias.
- Heterocedasticidad: Esta propiedad indica que existe una violación de la hipótesis de igualdad de varianzas. El test de Levene se utiliza para comprobar si *k* muestras presentan o no esta homogeneidad en las varianzas. Cuando los datos observados no cumplen la condición de la normalidad, es más fiable el resultado de utilizar este test con respecto al test de Bartlett [27], que se trata de otro test que verifica la misma propiedad.

En nuestro caso está clara la independencia de los sucesos puesto que son ejecuciones independientes del algoritmo con semillas iniciales aleatoriamente generadas. A continuación mostramos un análisis de la normalidad, utilizando los test de Kolmogorov-Smirnov y Shapiro-Wilk, junto a un análisis de heterocedasticidad utilizando el test de Levene.

B. Test de normalidad sobre el conjunto de funciones y algoritmos

Aplicamos el test de normalidad de Kolmogorov-Smirnov con probabilidad de error $p = 0,05$ (utilizamos SPSS). La Tabla I muestra los resultados donde el símbolo "*" indica que no se cumple la normalidad y el valor entre paréntesis se trata del valor *p* de confianza necesario para rechaza la hipótesis de normalidad. La Tabla II muestra los resultados aplicando el test de normalidad de Shapiro-Wilk. La Tabla III muestra los resultados aplicando el test de Levene, en donde el símbolo "*" indica que las varianzas para una determinada función no son homogéneas.

Claramente en ambos caso queda patente el incumplimiento de las condiciones de normalidad y homocedasticidad necesarias para el uso de test paramétricos.

C. Análisis sobre 3 funciones: f4, f13 y f17

A continuación presentamos el estudio realizado para las funciones f4, f13 y f17. Su descripción se puede encontrar en el Apéndice.

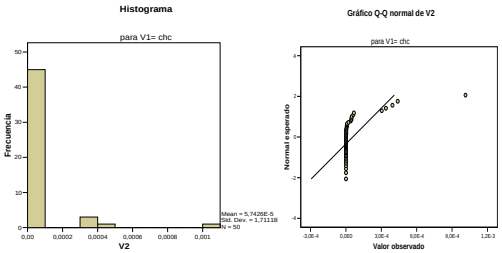

Fig. 1. Función F4 y algoritmo CHC: Histograma y Gráfico Q-Q.

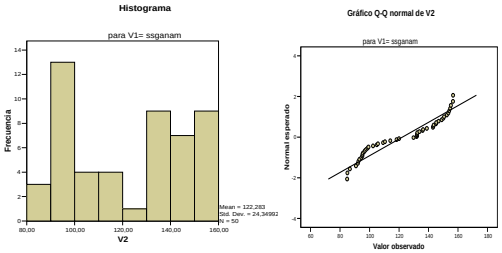

Fig. 2. Función F17 y algoritmo AGE-NAM: Histograma y Gráfico Q-Q.

TABLA I
TEST DE NORMALIDAD DE KOLMOGOROV-SMIRNOV

	f1	f2	f3	f4	f5	u1	u2
CHC	* (.02)	* (.00)	* (.00)	* (.00)	* (.00)	(.20)	* (.03)
AGG	* (.00)	* (.00)	* (.00)	* (.00)	(.20)	(.05)	(.08)
AGE-NAM	(.20)	* (.00)	* (.00)	* (.00)	(.09)	(.05)	* (.01)
AGE-WAMS	* (.00)	* (.00)	* (.02)	* (.00)	(.20)	* (.00)	* (.00)
	f6	f7	f8	f9	f10	f11	f12
CHC	* (.00)	* (.00)	(.20)	* (.00)	* (.00)	(.20)	* (.00)
AGG	* (.00)	(.13)	* (.02)	* (.00)	* (.01)	* (.04)	* (.00)
AGE-NAM	* (.00)	* (.01)	(.20)	* (.00)	* (.00)	* (.00)	* (.00)
AGE-WAMS	* (.00)	(.20)	(.20)	* (.00)	* (.00)	(.20)	* (.00)
	f13	f14	f15	f16	f17	f18	f19
CHC	(.20)	* (.03)	* (.00)	* (.00)	* (.01)	* (.00)	* (.00)
AGG	(.20)	(.06)	* (.00)	* (.00)	* (.03)	* (.00)	* (.00)
AGE-NAM	(.07)	(.20)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)
AGE-WAMS	* (.03)	(.20)	* (.00)	* (.01)	(.20)	* (.00)	* (.00)
	f20	f21	f22	f23	f24	f25	
CHC	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	
AGG	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	
AGE-NAM	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	
AGE-WAMS	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	

TABLA II
TEST DE NORMALIDAD DE SHAPIRO-WILK

	f1	f2	f3	f4	f5	u1	u2
CHC	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.03)	* (.00)
AGG	* (.00)	* (.00)	* (.00)	* (.00)	(.07)	* (.02)	* (.01)
AGE-NAM	(.11)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)
AGE-WAMS	* (.00)	* (.00)	* (.00)	* (.00)	* (.02)	* (.00)	* (.00)
	f6	f7	f8	f9	f10	f11	f12
CHC	* (.00)	* (.00)	(.39)	* (.00)	* (.00)	(.07)	* (.00)
AGG	* (.00)	* (.01)	(.13)	* (.00)	* (.02)	(.10)	* (.00)
AGE-NAM	* (.00)	* (.00)	(.35)	* (.00)	* (.00)	* (.00)	* (.00)
AGE-WAMS	* (.00)	(.92)	(.47)	* (.00)	* (.01)	(.89)	* (.00)
	f13	f14	f15	f16	f17	f18	f19
CHC	(.28)	(.07)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)
AGG	(.29)	* (.01)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)
AGE-NAM	* (.00)	(.25)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)
AGE-WAMS	(.21)	* (.04)	* (.00)	* (.00)	* (.03)	* (.00)	* (.00)
	f20	f21	f22	f23	f24	F25	
CHC	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	
AGG	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	
AGE-NAM	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	
AGE-WAMS	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	

TABLA III
TEST DE HETEROCEDASTICIDAD DE LEVENE (BASADO EN MEDIAS)

	f1	f2	f3	f4	f5	u1	u2
LEVENE	(.07)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)	* (.00)
	f6	f7	f8	f9	f10	f11	f12
LEVENE	(.21)	* (.00)	* (.04)	* (.00)	* (.00)	* (.00)	* (.00)
	f13	f14	f15	f16	f17	f18	f19
LEVENE	* (.02)	* (.02)	* (.00)	* (.00)	* (.00)	* (.00)	(.08)
	f20	f21	f22	f23	f24	f25	
LEVENE	(.14)	* (.01)	* (.00)	* (.00)	* (.00)	* (.00)	

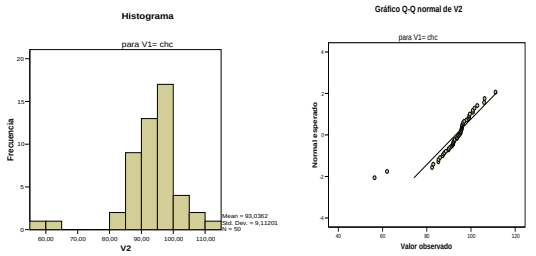

Fig. 3. Función F17 y algoritmo CHC: Histograma y Gráfico Q-Q.

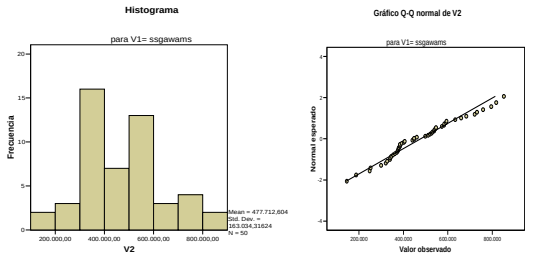

Fig. 4. Función F13 y algoritmo AGE-WAMS: Histograma y Gráfico Q-Q.

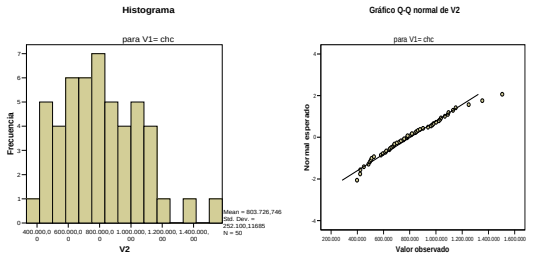

Fig. 5. Función F13 y algoritmo CHC: Histograma y Gráfico Q-Q.

Desde la Figura 1 hasta la 5, se muestran distintos ejemplos de representaciones gráficas de histogramas y gráficos Q-Q. Un histograma representa una variable estadística en forma de barras, de manera que la superficie de cada barra es proporcional a la frecuencia de los valores representados. Un gráfico Q-Q representa una confrontación entre los cuantiles de los datos observados y los de una distribución normal.

En las Figuras 1 y 2 observamos un caso típico de falta de normalidad absoluta. Las Figuras 3 y 4 muestra una representación gráfica de lo que se rechaza como normal con un nivel de confianza del 95%, pero no es rechazable con un nivel superior de confianza (99% en la Figura 3 y 97% en la 4) (ver Tabla I). Por último, la Figura 5 muestra un claro ejemplo en donde ningún test empleado puede rechazar la hipótesis de normalidad.

IV. SOBRE EL USO DE TEST NO PARAMÉTRICOS BASADOS EN EL ORDEN

En esta sección introducimos brevemente los test no paramétricos y presentamos un estudio experimental utilizando los 4 algoritmos y el conjunto de funciones de test.

Para diferenciar a un test no paramétrico del paramétrico hay que comprobar el tipo de datos que el test utiliza, tal y como vimos en la Sección III.A. Un test no paramétrico es aquel que utiliza datos de tipo nominal o datos ordinales o que representan un orden en forma de ranking. Esto no implica que solamente deban ser usados para ese tipo de datos. Podría ser interesante transformar los datos reales dentro de un intervalo a datos basados en orden, de tal forma que se pueda aplicar un test no paramétrico sobre datos típicos de tests paramétricos cuando éstos no cumplen las condiciones necesarias supuestas por el test. Como norma general, un test no paramétrico es menos restrictivo que un paramétrico, aunque menos robusto que un paramétrico cuya aplicación se realiza sobre datos que cumplen todas las condiciones necesarias.

A continuación, explicamos la funcionalidad básica de cada test no paramétrico junto al objetivo que se persigue con su utilización:

- Test de Friedman: Se trata de un equivalente no paramétrico al test de medidas-repetidas ANOVA. Calcula el orden de los resultados observados por algoritmo (r_j para el algoritmo j con k algoritmos) para cada función, asignando al mejor de ellos el orden 1, y al peor el orden k. Bajo la hipótesis nula, que se forma a partir de suponer que todos los resultados de cada algoritmo son equivalentes y, por tanto, sus rankings son similares, el estadístico de Friedman

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right],$$

se distribuye acorde a χ_F^2 con k - *1* grados de libertad, siendo $R_j = \frac{1}{N}\sum_i r_i^j$, y N el número de funciones.
- Test de Iman and Davenport [13]: Se trata de una medida derivada de la de Friedman a causa del efecto conservador indeseado que produce

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2},$$

se distribuye acorde a una distribución F con k – *1* y (k - *1*)(N - *1*) grados de libertad.
- Test de Bonferroni-Dunn: Si se rechaza la hipótesis nula en alguno de los anteriores tests, podemos proceder con test a posteriori. El test de Bonferroni-Dunn es similar al test de Tukey

para ANOVA y se utiliza cuando queremos comparar un algoritmo frente a los demás. La calidad de dos algoritmos es significativamente diferente si la correspondiente media de rankings es tan diferente como tu diferencia crítica

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} .$$

El valor de q_α es el valor crítico de Q' para una múltiple comparación no paramétrica con un control (Tabla B.16 en [27]).

- Test de Ranking de Signos de Wilcoxon: Se trata de una alternativa no paramétrica al t-test por parejas. Su funcionamiento se basa en calcular las diferencias entre los resultados de dos algoritmos y calcular un ranking utilizando dicho valor, ignorando signos, a través de todas las funciones. Nótese que en este caso, el ranking *d* va desde 1 hasta *N*, en vez de hasta *k*, como era el caso de los tres tests anteriores. Tras sumar los rankings diferenciándolos entre si son negativos o positivos, obtenemos dos valores R^+ y R^-. Si el menor de ellos es menor o igual al valor de la distribución *T* de Wilcoxon para *N* grados de libertad (Tabla B.12 en [27]), se rechaza la hipótesis nula.

A. Estudio experimental: Resultados y análisis

El conjunto de funciones se ha dividido en 3 grupos atendiendo al grado de dificultad.

- El primer grupo contiene las funciones unimodales u1 y u2 y de f1 a f5, todas aquellas funciones en las que todos los algoritmos participantes en la competición alcanzaban siempre el óptimo.
- El segundo grupo contiene las funciones multimodales que van desde f6 a f14, funciones para las que algunos algoritmos alcanzaban el óptimo Podríamos considerarlas de dificultad media.
- El tercer grupo contiene las restantes funciones, desde la función f15 a f25,; funciones difíciles desde la perspectiva de alcanzar el óptimo.

A continuación se muestra el resultado de aplicar los test de Friedman e Iman-Davenport para ver si hay diferencias en los resultados en la Tabla IV.

La Tabla IV nos indica en negrita el mayor valor entre los dos que se comparan, y si éste se corresponde con el valor que nos proporciona el estadístico, nos informa del rechazo de la hipótesis nula. En este ejemplo, tanto el test de Friedman como el de Iman-Davenport nos advierte de la existencia de diferencias significativas entre los resultados observados en las funciones del grupo 1 y grupo 3 y en todas las funciones a la vez. Un análisis estadísico a posteriori en el caso de las funciones f6 hasta la f14 (grupo 2) no va a ser necesario, puesto que los resultados son similares.

TABLA IV
RESULTADOS DEL TEST DE FRIEDMAN E IMAN-DAVENPORT

	Valor Friedman	Valor en χ^2	Valor Iman-Davenport	Valor en F_F
F1-U2	**8,486**	7,815	**4,068**	3,95
F6-F14	2,733	**7,815**	0,901	**3,72**
F15-F25	**21,873**	7,815	**19,657**	3,59
Todas	**26,867**	7,815	**12,904**	3,29

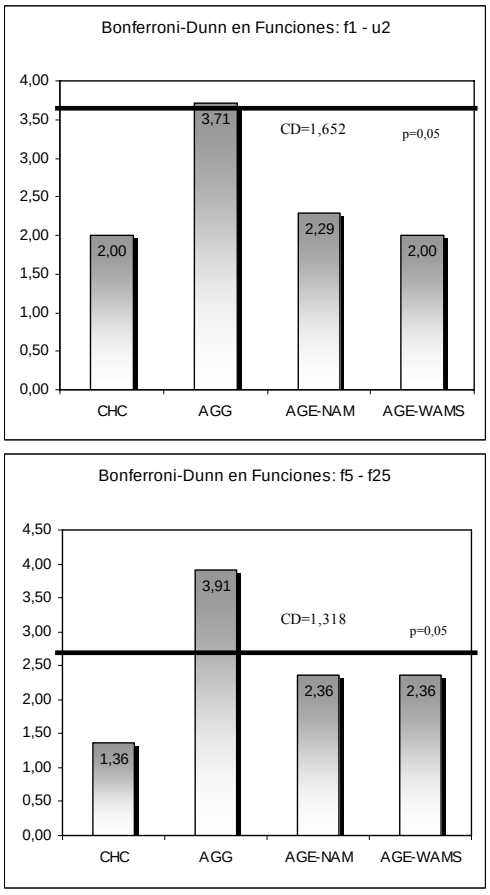

Fig. 6. Gráficas de Bonferroni-Dunn considerando los subgrupos de funciones con hipótesis rechazada por Friedman

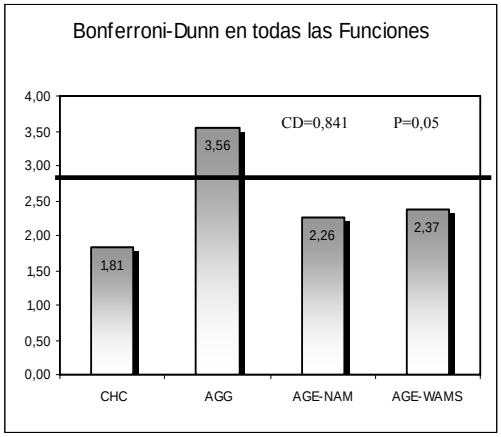

Fig. 7. Gráfica de Bonferroni-Dunn considerando todas las funciones

En los casos donde se rechaza la hipótesis nula, se muestra gráficamente, mediante diagramas de barras, la aplicación del test de Bonferroni-Dunn en las Figuras 6 y 7. Este tipo de gráficas representa un diagrama constituido par barras cuya altura es proporcional al orden medio que obtiene cada algoritmo. Si a la menor de ellas (que se corresponde con el mejor algoritmo), le sumamos la diferencia crítica obtenida por Bonferroni-Dunn, representando su resultado en una línea de corte en todo en gráfico, aquellas barras que superen la línea pertenecen a algoritmos cuyos resultados son significativamente peor que los aportados por el algoritmo control. Nótese que este tipo de gráficas representan la comparación de un algoritmo con el resto.

Es conocido que este test no muy sensible, de ahí que no muestre todas las diferencias significativas entre los algoritmos aunque intuimos que existen analizando el orden de los mismos.

Por ello vamos a aplicar el test de Wilcoxon para comparar los algoritmos entre sí 2 a 2 con un valor de $p = 0,05$. Para mostrar los resultados de este test, utilizaremos un formato de tabla específico (ver Tablas V – VII). Se trata de tablas cuadradas en donde se indica, tanto en la primera fila como columna, el nombre del algoritmo que se compara. La celda que coindice con la fila i y la columna j indica el resultado Wilcoxon que compara los algoritmos indicados en la fila i y columna j. En cada celda encontramos un signo +, - ó =; indicado que el algoritmo de la fila es mejor, peor o igual que el algoritmo de la columna, respectivamente.

TABLA V
TEST DE WILCOXON PARA FUNCIONES UNIMODALES (F1-U2)

	CHC	GGA	SSGA-NAM	SSGA-WAMS
CHC		=	=	=
GGA	=		-	-
SSGA-NAM	=	+		=
SSGA-WAMS	=	+	=	

TABLA VI
TEST DE WILCOXON PARA FUNCIONES MULTIMODALES (F15-F25)

	CHC	GGA	SSGA-NAM	SSGA-WAMS
CHC		+	+	+
GGA	-		-	-
SSGA-NAM	-	+		=
SSGA-WAMS	-	+	=	

TABLA VII
TEST DE WILCOXON PARA TODAS LAS FUNCIONES

	CHC	GGA	SSGA-NAM	SSGA-WAMS
CHC		+	=	+
GGA	-		-	-
SSGA-NAM	=	+		=
SSGA-WAMS	-	+	=	

Un breve análisis de estas tablas nos permite concluir:

- Analizando el orden en las Figuras 6 y 7, veíamos que los AGEs eran muy similares, observación que es corroborada por el test de Wilcoxon.
- Como hemos comentado, el test de Bonferroni-Dunn es poco robusto y no mostraba diferencias significativas entre el algoritmo CHC que mostraba el mejor orden y los dos AGEs. Al aplicar el test de Wilcoxon, encontramos diferencias significativas en el algoritmo AGE-WAMS para las funciones difíciles (f15-f25) y en el análisis global de las 27 funciones. En cuanto a la comparación con el AGE-NAM, solo muestra mejoras significativas en el conjunto de funciones difíciles (f15-f25), aunque consideradas en su conjunto las 27 funciones, las diferencias no lo son, mostrando valores de $R^+ = 235$ y $R^- = 143$, siendo necesaria una que el menor de ellos sea menor o igual a 107 (distribución T con $N = 27$).
- Como era de esperar, el peor de los algoritmos utilizados AGG muestra peor comportamiento

que los algoritmos AGE, y solo no muestra diferencias significativas con el algoritmo CHC para las funciones unimodales ($R^+ = 23$ y $R^- = 5$, *dif. = 2*). Cuando el número de funciones que se utilizan al aplicar Wilcoxon es pequeño (inferior a 10), el test es muy restrictivo a la hora de rechazar la hipótesis nula, y una pequeña diferencia existente en una sola función puede aumentar la suma del orden hasta tal punto que se supere el valor crítico. En este caso, la menor suma alcanza el valor de $R^- = 5$, que es muy inferior a la otra suma, que vale $R^+ = 23$. Wilcoxon requiere un valor, como mínimo, de $R^- = 2$ y $R^+ = 26$ para rechazar la hipótesis nula entre CHC y AGG.

En general podemos concluir que el algoritmo CHC es el mejor de los 4 estudiados puesto que muestra diferencias significativas en el orden, y estadísticamente es mejor que el resto en las funciones difíciles (f15-f25), aunque analizadas globalmente las 27 funciones no alcanza diferencias significativas para el test de Wilcoxon cuando se compara con AGE-NAM.

V. CONCLUSIONES

En este trabajo hemos estudiado el uso de las técnicas estadísticas para el análisis del comportamiento de los algoritmos evolutivos en problemas de optimización, analizando el uso de los test estadísticos paramétricos y no paramétricos.

Hemos dejado clara la necesidad de utilizar tests no paramétricos cuando se analizan algoritmos evolutivos para problemas de optimización continua, puesto que no se verifican las condiciones iniciales que garanticen la fiabilidad de los tests paramétricos.

En cuanto al uso de los test no paramétricos, hemos mostrado como utilizar el test de Friedman, Iman-Davenport, Bonferroni-Dunn y Wilcoxon, que en conjunto pueden ser una buena herramienta para el análisis de los algoritmos.

Existen algoritmos más robustos que el algoritmo de Bonferroni-Dunn, para contrastar un algoritmo de control que están igualmente basados en el orden, y que serán objeto de estudio en una extensión del presente trabajo. Algunos de ellos son los tests de Holm, Holmes y Hochberg. Una aplicación de los mismo la podemos encontrar en [15].

AGRADECIMIENTOS

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

REFERENCIAS

[1] Baker, J.E., Adaptive selection methods for genetic algorithms. In Proceedings of the 1st International Conference on Genetic Algorithms. pp. 101-111. 1985.

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

[3] Cedeño, W., Vemuri, V., Multi-niche crowding in genetic algorithms and its application to the assembly of dna restriction-fragments. Evolutionary Computation. Vol. 2. No. 4. pp. 321-345. 1995.

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

[5] Demsar, J., Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research. Vol. 7. pp. 1–30. 2006.

[6] Eshelman, L.J., The CHC adaptative search algorithm: How to have safe search when engaging in nontraditional genetic recombination, in Foundations of Genetic Algorithms, Rawlins, G.J.E., Ed., pp. 265–283. 1991.

[7] Fernandes, C., Rosa, A., A study of non-random matching and varying population size in genetic algorithm using a royal road function. Proc. of the 2001 Congress on Evolutionary Computation. pp. 60-66. 2001.

[8] François, O., Lavergne, C., Design of evolutionary algorithms - A statistical perspective. IEEE Transactions on Evolutionary Computation. Vol. 5. No. 2. pp. 129-148. 2001.

[9] Gallagher, M., Yuan B., A General-Purpose Tunable Landscape Generator. IEEE Transactions on Evolutionary Computation. Vol. 10. No. 5. pp. 590-603. 2006.

[10] Goldberg, D.E., Deb, K., A comparative analysis of selection schemes used in genetic algorithms. Foundation of Genetic Algorithm. pp. 69-93, 1991.

[11] Hervás-Martínez, C., Ortiz-Boyer, D., Analyzing the statistical features of CIXL2 crossover offspring. Soft Computing. Vol. 9. No. 4. pp. 270-279. 2005.

[12] Hooker, J.H., Testing Heuristics: We Have it All Wrong. Journal of Heuristics. Vol. 1. No. 1. pp. 33-42. 1995.

[13] Iman, R.L, Davenport, J.M, Approximations of the critical region of the Friedman statistic. Communications in Statistics. pp. 575-595. 1980.

[14] Lunacek, M., Whitley, D., Knight, J.N., Measuring mobility and the performance of global search algorithms. GECCO 2005 - Genetic and Evolutionary Computation Conference. pp. 1209-1216. 2005.

[15] Manly, K.F., Nettleton, D., Gene Hwang, J.T., Genomics, prior probability, and statistical tests of multiple hypotheses. Genome Research. Vol. 14. pp. 997-1001. 2004.

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

[17] Ozcelik, B., Erzurumlu, T., Comparison of the warpage optimization in the plastic injection molding using ANOVA, neural network model and genetic algorithm. Journal of Materials Processing Technology. Vol. 171. No. 3. pp. 437-445. 2006.

[18] Rojas, I., González, J., Pomares, H., Merelo, J.J., Castillo, P.A., Romero, G., Statistical analysis of the main parameters involved in the design of a genetic algorithm. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews. Vol. 32. No. 1. pp. 31-37. 2002.

[19] Schmidt, C., Branke, J., Chick, S.E., Integrating techniques from statistical ranking into evolutionary algorithms. LNCS 3907. pp. 752-763. 2006.

[20] Sheskin, D.J., Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press. 2000.

[21] 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. May 2005.
[22] Whitley, D., An overview of evolutionary algorithms: Practical issues and common pitfalls. Information and Software Technology. Vol. 43. No. 14. pp. 817-831. 2001.
[23] Whitley, D., Beveridge, R., Graves, C., Mathias, K., Test driving three 1995 genetic algorithms: New test functions and geometric matching. Journal of Heuristics. Vol. 1. No. 1. pp. 77-104. 1995.
[24] Whitley, D., Rana, S., Dzubera, J., Mathias, K.E., Evaluating evolutionary algorithms. Artificial Intelligence. Vol. 85. pp. 245-276. 1996.
[25] Wolpert, D.H., Macready, W.G., No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. Vol. 1. No. 1. pp. 67–82. 1997.
[26] Yuan B., Gallagher M., On Building a Principled Framework for Evaluating and Testing Evolutionary Algorithms: A Continuous Landscape Generator. In Proceedings of the 2003 Congress on Evolutionary Computation, IEEE. pp. 451-458. 2003.
[27] Zar, J.H., Biostatistical Analysis. Prentice Hall. 1999.

APÉNDICE

En esta sección vamos a describir las tres funciones que se analizaron en la Sección III.C.

A. Problema desplazado Schwefel 1.2 con ruido en el Fitness (f4)

$$f(x) = \left(\sum\left(\sum z_j\right)^2\right) * (1 + 0.4|N(0,1)|) + f_{bias},$$

$$z = x - o, x = \left[x_1, x_2, ..., x_D\right]$$

D: Dimensión.

$o = \left[o_1, o_2, ..., o_D\right]$ es el óptimo global.

$f(o) = f_{bias}, f_{bias} = -450$

- Propiedades:
  - Unimodal.
  - Desplazada.
  - No separable.
  - Escalable.
  - Ruido en la función de fitness.
  - $x \in [-100,100]^{D.}$.

B. Combinación extendida de las funciones de Griewank f_8 y Rosenbrock f_2 (f13)

$$f_{13}(x_1, x_2, ..., x_D) = f_8(f_2(x_1, x_2)) + f_8(f_2(x_2, x_3)) + \\ + ... + f_8(f_2(x_{D-1}, x_D)) + f_8(f_2(x_D, x_1))$$

donde

$$f_8 = f_i(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$f_2 = \sum_{i=1}^{D}\left(100\left(x_i^{\,2} - x_{i+1}\right)^2 + (x_i - 1)^2\right)$$

$$z = x - o + 1, x = \left[x_1, x_2, ..., x_D\right]$$

D: Dimensión

$o = \left[o_1, o_2, ..., o_D\right]$, es el óptimo global.

$f(o) = f_{bias}, f_{bias} = -130$

- Propiedades:
  - Multimodal.
  - Desplazada.
  - No separable.
  - Escalable.
  - Ruido en la función de fitness.ç

C. Función compuesta 1 con Ruido en el Fitness (f17)

$$f(x) = G(x) * (1 + 0.2|N(0,1)|) + f_{bias}$$

D: Dimensión.

$o = \left[o_1, o_2, ..., o_D\right]$es el óptimo global.

$f(o) = f_{bias}, f_{bias} = 120$

- Propiedades:
  - Multimodal.
  - Desplazada.
  - No separable.
  - Escalable.
  - Un gran número de óptimos locales.
  - Mezcladas funciones con diferentes propiedades.
  - Ruido Gausiano en la función de fitness.
  - $x \in [-5,5]^{D.}$.

$$G(x) = \sum\left(z_i^2 - 10\cos(2\pi z_i) + 10\right),$$

$$z = \left((x - o_1)/\lambda_i\right) * M_i$$

$$G(x) = \sum_{i=0}^{N}\left(w_i * \left[f_i'\left((x - o_i)/\lambda_i M_i\right) + bias\right]\right) + f_{bias}$$

donde:

- Se combinan *N=9* funciones.
- w_i son pesos asociados a cada función f_i .
- M_i son matrices de transformación lineal.
- λ_i son constantes que comprimen o amplian cada función.

$$\lambda = [1,1,10,10,5/60,5/60,5/32,5/32,5/100,5/100]$$