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

El teorema de "No free lunch" [25] demuestra 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 pueden hacer adecuados para la resolución del problema.

[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

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 [8,26]. 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 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, ...) [3,17,18].

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 [7,10,16,19]; b) los estudios mostrados en [1], 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 [4].

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 [5,22], que combina diferentes mecanismos para conseguir un buen equilibro entre diversidad y convergencia, como la prevención de incesto, la reinicialización de la población cuando el proceso de búsqueda se estanca y la competición entre padres e hijos en el proceso de reemplazamiento.
- AGE-NAM: Algoritmo Genético Estacionario [22] que utiliza un método de selección orientado a escoger padres distantes entre sí llamado NAM [6].
- AGE-WAMS: Algoritmo Genético Estacionario que utiliza un método de reemplazo que mantiene diversidad en la exploración llamado WAMS [2].

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 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 nuestro caso usamos torneos de tamaño 2.

Características propias del AGE-NAM.

- Selección: El Emparejamiento Variado Inverso (*Negative Assortative Mating, NAM)* [6]. 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 [9].

Características propias del AGE-WAMS:
- Selección por Torneo. En nuestros experimentos utilizamos N_{TS}=3.
- Reemplazo: Reemplazar el Peor Entre Semejantes (*Worst Among Most Similar Replacement, WAMS*) [2]. 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.

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 con valores 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 iniciales para el uso de los test paramétricos se incumplan, haciendo que el análisis estadístico 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.

En cuanto al estudio de la heterocedasticidad, la Tabla III muestra los resultados aplicando el test de Levene, en donde el símbolo "*" indica que las varianzas de las distribuciones de los diferentes algoritmos 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.

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.

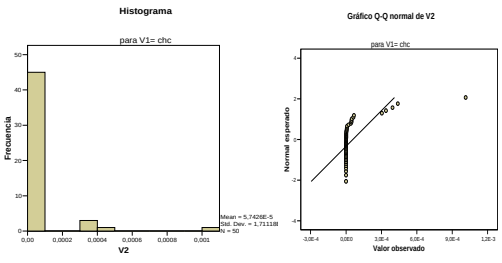

Fig. 1. Función F4 y algoritmo CHC: Histograma y Gráfico Q-Q.

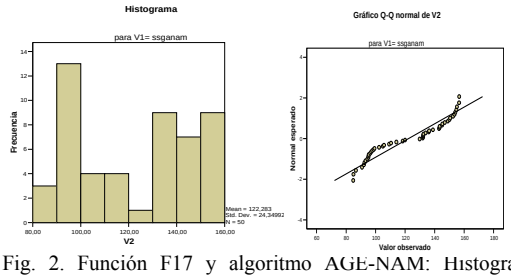

Fig. 2. Función F17 y algoritmo AGE-NAM: Histograma y Gráfico Q-Q.

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)	

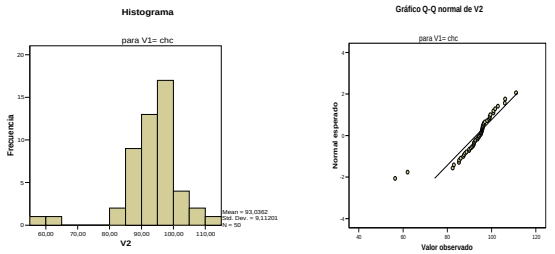

Fig. 3. Función F17 y algoritmo CHC: Histograma y Gráfico Q-Q.

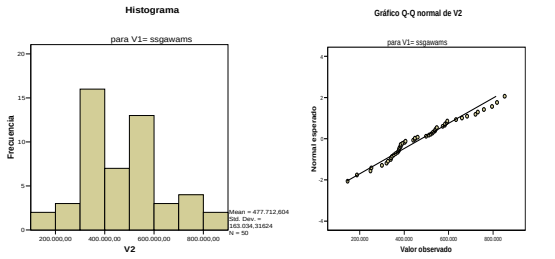

Fig. 4. Función F13 y algoritmo AGE-WAMS: Histograma y Gráfico Q-Q.

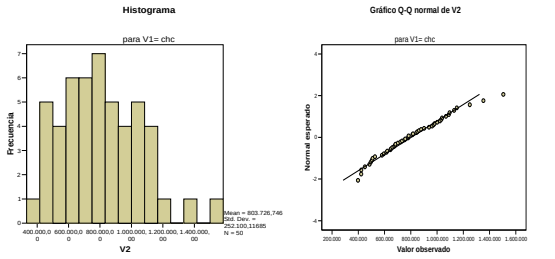

Fig. 5. Función F13 y algoritmo CHC: Histograma y Gráfico Q-Q.

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.

Por tanto, podemos afirmar que no se cumplen las condiciones necesarias para la aplicación de los tests paramétricos. Esto nos conduce hacia la necesidad de utilizar tests alternativos (tests no paramétricos).

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 ese tipo de datos. Podría ser interesante transformar los datos de valores 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 por el uso del 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 utilizado en este estudio 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 los resultados de los algoritmos 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. Los valores críticos del estadístico de Friedman coinciden exactamente con los establecidos en la distribución χ^2 cuando $N >$ *10* y $k > 5$. En caso contrario, los valores exactos pueden consultarse en [20,27].
- Test de Iman and Davenport [14]: Se trata de una medida derivada de la de Friedman a causa del efecto conservador indeseado que produce éste. El estadístico es

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

y 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 de control frente a los demás. La calidad de dos algoritmos es significativamente diferente si la correspondiente media de rankings es tan diferente como su 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 Holm [11]: Para contrastar el procedimiento de Bonferroni-Dunn, disponemos de un test que prueba secuencialmente las hipótesis ordenadas según su significancia. Denominaremos a los valores de *p* ordenados por $p_1, p_2, \ldots$, de tal forma que $p_1 \le p_2 \le \ldots \le p_{k-1}$. El método de Holm compara cada p_i con $\alpha/(k-i)$ comenzando desde el valor de *p* más significante. Si p_1 es menor que $\alpha/(k-1)$, la correspondiente hipótesis se rechaza y nos permite comparar p_2 con $\alpha/(k-2)$. Si la segunda hipótesis se rechaza, continuamos el proceso. En cuanto una determinada hipótesis no puede ser rechazada, todas las restantes se mantienen como aceptadas. El estadístico para comparar el algoritmo *i*-ésimo con el *j*-ésimo es:

$$z = (R_i - R_j) \Big/ \sqrt{\frac{k(k+1)}{6N}} .$$

El valor *z* se utiliza para encontrar la probabilidad correspondiente a partir de la tabla de la distribución normal, la cual es comparada con el correspondiente valor de α.

El test de Holm es más potente que Bonferroni-Dunn y no hace ninguna suposición adicional sobre las hipótesis chequeadas.

- 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, y el algoritmo asociado al mayor de los valores es el mejor.

A. Estudio experimental: Resultados y análisis

El conjunto de funciones se ha dividido en 2 grupos atendiendo al grado de dificultad.

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

En la Tabla IV se muestra el resultado de aplicar los test de Friedman e Iman-Davenport para ver si hay diferencias en los resultados. Señalamos 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 2 y en todas las funciones a la vez.

TABLA IV
RESULTADOS DEL TEST DE FRIEDMAN E IMAN-DAVENPORT

	Valor Friedman	Valor en χ^2	Valor Iman-Davenport	Valor en F_F
F1-F14	**11,657**	7,815	**4,994**	3,72
F15-F25	**21,873**	7,815	**19,657**	3,59
Todas	**30,744**	7,815	**16,672**	2,72

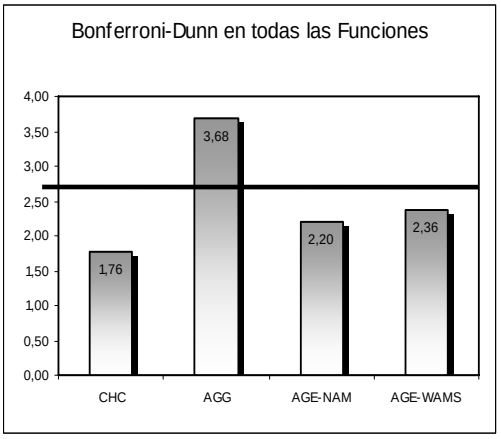

Fig. 6. Gráfica de Bonferroni-Dunn considerando todas las funciones

Atendiendo a estos resultados, es necesario un análisis estadístico a posteriori en todos los casos.

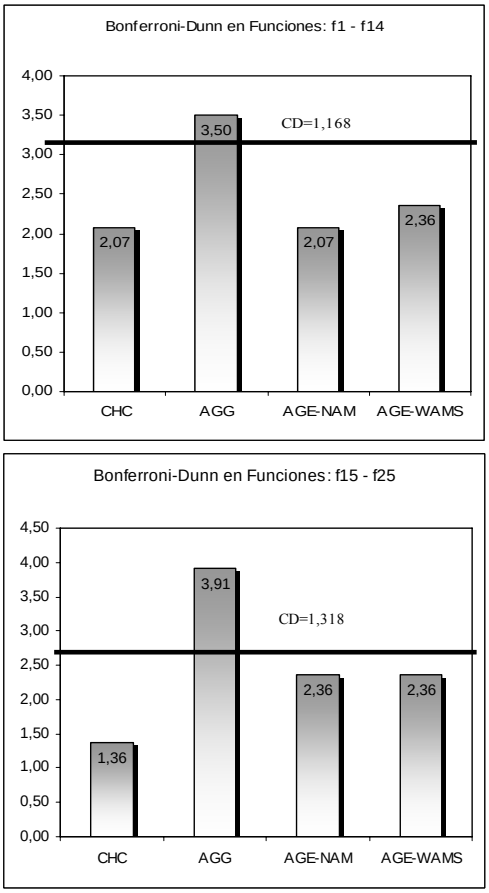

Fig. 7. Gráficas de Bonferroni-Dunn considerando los subgrupos de funciones con hipótesis rechazada por Friedman

Se muestra gráficamente, mediante diagramas de barras, la aplicación del test de Bonferroni-Dunn en las Figuras 6 y 7. Estas gráficas representan un diagrama de 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 (valor de CD), 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 peores que los aportados por el algoritmo control. Nótese que este tipo de gráficas representan la comparación de el algoritmo de control (el de mejor comportamiento) con el resto.

Es conocido que este test no muy potente, de ahí que no muestre todas las potenciales diferencias significativas entre los algoritmos. Por ello, se han desarrollado tests más potentes como el test de Holm (anteriormente descrito) que utilizaremos a continuación.

Aplicaremos el test de Holm para comparar el mejor algoritmo (algoritmo control, CHC en todos los casos) con todos los demás, con un valor de $p = 0,05$. Para mostrar los resultados de este test, utilizaremos un formato de tabla específico (ver Tablas V – VII). Estas tablas se componen de 6 columnas y una fila por comparación entre el algoritmo control y otro algoritmo de los restantes. Las dos primeras columnas identifican el orden y el nombre del algoritmo respectivamente. La columna identificada por z es el valor obtenido por la expresión explicada en la Sección IV sobre este test. La siguiente es el valor p asociado al z correspondiente a partir de la distribución normal y a continuación el valor de α / i que se ha de comparar. Finalmente, la última columna muestra si se rechaza (R) la hipótesis asociada a la fila o se acepta (A).

TABLA V
TEST DE HOLM PARA FUNCIONES DEL PRIMER GRUPO (F1-F14)

i	algoritmo	z	p	α / i	Hip.
3	AGG	2,928	0,003	0,017	R
2	AGE-WAMS	0,586	0,558	0,025	A
1	AGE-NAM	9,101	1,000	0,050	“

TABLA VI
TEST DE HOLM PARA FUNCIONES DEL SEGUNDO GRUPO (F15-F25)

i	algoritmo	z	p	α / i	Hip.
3	AGG	4,624	3,7E-6	0,017	R
2	AGE-WAMS	1,817	0,069	0,025	A
1	AGE-NAM	1,817	0,069	0,050	“

TABLA VII
TEST DE HOLM PARA TODAS LAS FUNCIONES

i	algoritmo	z	p	α / i	Hip.
3	AGG	5,258	1,5E-7	0,017	R
2	AGE-WAMS	1,643	0,100	0,025	A
1	AGE-NAM	1,204	0,228	0,050	“

Como podemos observar tanto en las gráficas de las Figuras 6 y 7 y las Tablas V-VII, los resultados obtenidos son los mismos para el test de Bonferroni-Dunn y el test de Holm. Ambos rechazan la hipótesis de igualdad de medias existente entre CHC y AGG con un valor $p=0,05$; pero no distinguen diferencia de comportamiento entre CHC con AGE-WAMS y CHC con AGE-NAM. Un test de comparación por parejas (como el test de Wilcoxon) puede llegar a detectar diferencias que ahora no se observan, como veremos más adelante.

TABLA VIII
TEST DE WILCOXON PARA LAS FUNCIONES DEL SEGUNDO GRUPO (F15–F25)

alg.	R^+	R^-	Hip. *p=0,01*	Hip. *p=0,02*	Hip. *p=0,05*	Hip. *p=0,1*
AGG	66,0	0,0	R	R	R	R
AGE-WAMS	57,0	9,0	A	A	R	R
AGE-NAM	59,0	7,0	A	R	R	R

TABLA IX
TEST DE WILCOXON PARA TODAS LAS FUNCIONES

alg.	R^+	R^-	Hip. *p=0,01*	Hip. *p=0,02*	Hip. *p=0,05*	Hip. *p=0,1*
AGG	304,0	21,0	R	R	R	R
AGE-WAMS	257,0	68,0	R	R	R	R
AGE-NAM	228,0	97,0	A	A	A	R

A continuación, se presenta un último estudio con las tablas de comparaciones correspondiente al test de Wilcoxon entre CHC y el resto de métodos (comparaciones dos a dos). Las Tablas VIII y IX recogen los resultados de este procedimiento, en donde se especifican los valores de R^+ (asociado a CHC) y R^- (explicados en la Sección IV) del test y si se aceptan o rechazan las hipótesis para distintos valores de *p*. En la Tabla VIII, el valor crítico del test de Wilcoxon es 5, 7, 10 y 13 para *p=0,01; p=0,02, p=0,05* y *p=0,10*, respectivamente. En la Tabla IX, el valor crítico del test de Wilcoxon es 68, 76, 89 y 100 para *p=0,01; p=0,02, p=0,05* y *p=0,10*; respectivamente.

Como este test realiza comparaciones de parejas de algoritmos de forma independiente, no se controla que el nivel de significación alcance un determinado umbral que buscamos cuando hacemos la comparación múltiple del método. A esto último se le denomina error producido por familia (*familywise error rate: FWER*). La verdadera significación estadística para test de comparaciones por parejas viene dada por:

$$p = P(\text{Rechazar } H_0 \mid H_0 \text{ cierto}) =$$
$$= 1 - P(\text{Aceptar } H_0 \mid H_0 \text{ cierto}) =$$
$$= 1 - P(\text{Aceptar } A_k = A_i, i = 1,...,k-1 \mid H_0 \text{ cierto}) =$$
$$= 1 - \prod_{i=1}^{k-1} P(\text{Aceptar } A_k = A_i \mid H_0 \text{ cierto}) =$$
$$= 1 - \prod_{i=1}^{k-1} \left[1 - P(\text{Rechazar } A_k = A_i \mid H_0 \text{ cierto})\right] =$$
$$= 1 - \prod_{i=1}^{k-1} (1 - p_{H_i})$$

A partir de esta expresión, en la Tabla VIII, podemos deducir que CHC es mejor que el resto de algoritmos con un valor

$$p = 1 - ((1 - 0{,}01)\cdot(1 - 0{,}05)\cdot(1 - 0{,}02)) = 0{,}078 .$$

Para la Tabla IX, el resultado del valor de *p* es:

$$p = 1 - ((1 - 0{,}01)\cdot(1 - 0{,}01)\cdot(1 - 0{,}10)) = 0{,}118 .$$

Un análisis de todos los resultados experimentales nos permite concluir:

- Analizando el orden en las Figuras 6 y 7, veíamos que los AGEs son muy similares, observación que es corroborada cuando calculamos los valores *z* para el test de Holm.
- Las mayores diferencias registradas entre los cuatro métodos estudiados en este trabajo se producen cuando consideramos el estudio de las funciones difíciles (segundo grupo).
- El peor de los algoritmos utilizados, AGG, muestra peor comportamiento que el algoritmo control CHC cuando aplicamos el test de Bonferroni-Dunn. Como era de esperar, esta misma diferencia es también detectada por el test de Holm y la aplicación del test de Wilcoxon en pareja con CHC.
- Como hemos visto, aunque CHC es el algoritmo con mejor rendimiento de los cuatro empleados, ninguno de los tests de múltiples comparaciones utilizados considera, con un valor *p=0,05,* que exista diferencia con respecto a los AGEs.
- Sin embargo, hemos comprobado que una aplicación del test de Wilcoxon considerando la pareja CHC y AGE-WAMS con un valor *p=0,05* rechaza la hipótesis nula indicando diferencias entre ambos (a favor de CHC). Con AGE-NAM esto solo sucede en el segundo grupo de funciones, mientras que considerando todas ellas es necesario un valor de *p=0,10*.
- Aunque pueda parecer que el test de Wilcoxon es más potente que Holm, este test no controla el FWER, por lo que se produce una acumulación de probabilidad de error cuando lo usamos en comparaciones múltiples. Aún así, considerando las funciones del segundo grupo, podemos afirmar que CHC es significativamente mejor que el resto de métodos cuando consideramos las funciones complejas con un valor *p=0,078* (que entra dentro del rango de valores *p* comúnmente empleados en estudios experimentales, $p \leq 0{,}1$), y mejor que el resto considerando todas las funciones con un valor *p = 0,118*.

En general, podemos concluir que el algoritmo CHC es el mejor de los cuatro estudiados, pero hemos calculado, estadísticamente hablando, que la diferencia con respecto al peor (AGG) es clara, mientras que las existentes con respecto a los dos métodos intermedios no lo son tanto. Podríamos afirmar que el error cometido al afirmar que "CHC es mejor que el resto de algoritmos estudiados" es bastante menor cuando consideramos el grupo de funciones difíciles (segundo grupo) que cuando consideramos el estudio con todas las funciones.

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 son una buena herramienta para el análisis de los algoritmos.

Existen tests estadísticos más potentes que el procedimiento de Bonferroni-Dunn y Holm, 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 Hommel y Hochberg. Una aplicación de los mismos la podemos encontrar en [15].

AGRADECIMIENTOS

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

REFERENCIAS

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

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

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

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

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

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

[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] Ortiz-Boyer, D., Hervás-Martínez, C., García-Pedrajas, N., CIXL2: A crossover operator for evolutionary algorithms based on population features. Vol. 24. pp. 1-48. 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.

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