Diseños experimentales y tests estadísticos, tendencias actuales en Machine Learning.

José Otero, Luciano Sánchez

Resumen— **En algunas disciplinas tales como análisis clínicos, farmacología, genómica, etc. el diseño experimental y los tests estadísticos se emplean de forma rutinaria. En Machine Learning, estas técnicas cada vez juegan un papel más importante a la hora de demostrar la relevancia de nuevas aportaciones o de comparar las existentes. Sin embargo la naturaleza de los algoritmos empleados en ML y de los problemas utilizados como benchmark presentan ciertas dificultades a la hora de plantear los experimentos. Existen distintas alternativas para superar estas dificultades y no existe acuerdo entre los miembros de la comunidad científica sobre la metodología a emplear en una experimentación de este tipo. En este artículo se pretende dar una visión general de las aproximaciones más habituales encontradas en la literatura, con énfasis en los trabajos centrados en el análisis de las metodologías existentes, particularizado para el caso de comparaciones entre dos algoritmos, entre más de dos algoritmos y para el caso de los algoritmos multicriterio. Con ello se pretende encontrar un posible consenso entre los miembros de la comunidad de ML que permita una comparación objetiva entre resultados de trabajos realizados por distintos investigadores.**

Palabras clave— **Diseño experimental, tests estadísticos, aprendizaje de máquinas.**

I. Introducción

El contexto teórico más amplio en el que se puede encuadrar el estudio de las metodologías experimentales de ML es el Análisis Experimental de Algoritmos, una extensa bibliografía anotada sobre esta cuestión se puede encontrar en [55]. En [56], se apuntan ya algunas cuestiones comunes a los algoritmos de ML como problemas de redondeo, no aleatoriedad del generador de números aleatorios, muestreo del espacio de entradas y salidas, medidas de precisión, no reproducibilidad, etc. En definitiva se trata de pasar de estudiar una entidad abstracta, el algoritmo, a un objeto físico, un computador ejecutando un programa con un conjunto de datos específico. En [45] se presenta un estudio más completo de estos temas. Específicamente centrados en el análisis de algoritmos propios de ML, se pueden consultar [4], [67] en donde se detallan los errores más frecuentes cometidos en experimentaciones de ML y se proponen ciertas soluciones. Existen textos completamente dedicados a este tema como por ejemplo [18], en donde además se incluye todas las nociones bñasicas sobre estadística necesarias para comprender el resto del texto. Es destacable el trabajo de Dietterich [25] que si bien ha recibido algunas críticas en lo que se refiere al test estadístico propuesto [37][10][12], contiene la taxonomía completa de los problemas que se pueden encontrar en los experimentos de ML.

En los trabajos sobre ML se suelen encontrar comparaciones entre algoritmos [63]. Estas comparaciones implican la realización de dos tareas, la medida de la precisión de los algoritmos y la propia comparación de esta precisión, para lo cual será necesario emplear algún tipo de test estadístico. Según [25], la métrica del error, la elección de los conjuntos de entrenamiento y test, y la estocasticidad, son cuestiones a tener en cuenta a la hora de decidir la metodología a seguir. Cuando se comparan más de dos algoritmos, es necesario tener en cuenta la problemática particular de los test de comparaciones múltiples [68]. Finalmente, los algoritmos multiobjetivo presentan dificultades adicionales relacionadas con su naturaleza, que implica la adopción de alguna métrica de calidad [17].

En este trabajo, un experimento consiste en resolver una serie de problemas usando una implementación de un algoritmo. El conjunto de problemas, medidas realizadas, los detalles de la implementación y, en general, el contexto que acompaña a la realización de los experimentos, y que puede ser relevante de cara a la extracción de conclusiones sobre las medidas realizadas, conforma el *diseño experimental* utilizado [16][64].

La elección de un diseño experimental adecuado para un problema de aprendizaje automático es un punto de controversia entre la comunidad científica [4][25][67][46][86]. En trabajos recientes, como [63], los algoritmos de aprendizaje se evalúan mediante la comparación de sus resultados sobre conjuntos de datos conocidos [8], utilizando un test estadístico para juzgar la relevancia de las diferencias. Este mismo enfoque será seguido en este trabajo, si bien somos conscientes de que algunos autores cuestionan el que sea posible extraer conclusiones sobre el rendimiento de un algoritmo utilizando los conjuntos de ejemplos más habituales, según lo que se conoce como ”No free lunch Thorem”, según el cual, el error promediado en todos los datasets posibles, de cualesquiera dos algoritmos es el mismo [44][73][82]. Por otra parte, la naturaleza de estos diseños experimentales es tal que frecuentemente se vulneran una o más de las condiciones que han de cumplirse para la aplicación de determinado test estadístico [62][18][67]. La estructura de este trabajo es como sigue, en primer lugar se revisan los test estadísticos y diseños experimentales más habituales en aprendizaje automático cuando se comparan dos algoritmos. En segundo lugar se comentan los test de comparaciones múltiples más utilizados, con énfasis en los que son de más utilidad en ML. A continuación se detalla la problemática de los algoritmos multiobjetivo y se comentan las aportaciones más recientes en este sentido.

Dpto. Informática, Grupo de Metrología y Modelos, Universidad de Oviedo. E-mail: jotero@lsi.uniovi.es, luciano@uniovi.es

II. Diseños experimentales más habituales

A. Validación cruzada

La validación cruzada [74][78] es el diseño experimental más utilizado entre los investigadores en aprendizaje automático. En este método, los datos disponibles se dividen aleatoriamente en un conjunto de entrenamiento y un conjunto de test. El conjunto de entrenamiento se subdivide, a su vez, en dos conjuntos disjuntos

- El *conjunto de estimación*, usado para seleccionar el algoritmo.
- El *conjunto de validación*, usado para probar o validar el algoritmo.

La motivación de esta división está en validar el algoritmo sobre un conjunto de datos diferente del empleado para estimar sus parámetros.

Existen numerosas variantes de la validación cruzada. La que se se ha mencionado es conocida como el método *hold out*, y es menos utilizada en la actualidad que la *multifold cross validation* o *k-fold cross validation*. Esta última consiste en dividir el conjunto de ejemplos de que se dispone en k conjuntos disjuntos de igual tamaño, T_1,..,T_k. Se realizan k experimentos, usando como conjunto de entrenamiento en la iteración i-ésima $\bigcup_{j \neq i} T_j$ y como conjunto de test T_i. Cada algoritmo da lugar a una muestra de k estimaciones del error, y las diferencias entre dos algoritmos se juzgan mediante un contraste acerca de las diferencias entre las medias o las medianas del error muestral, como se verá a continuación.

La mayor ventaja de este diseño experimental es que las estimaciones del error sobre los conjuntos de test son independientes (los conjuntos de test no se solapan). Sin embargo, sí existe un cierto solapamiento en lo que se refiere al conjunto de entrenamiento, ya que cada pareja de conjuntos de entrenamiento comparte una alta fracción de los ejemplos. Por este motivo, este diseño experimental no estudia de forma adecuada la variabilidad inducida por la utilización de distintos ejemplos para el entrenamiento. Adicionalmente, existe un claro desequilibrio entre el número de ejemplos utilizado para test y para train cuando $k > 3$. Esta circunstancia tiene dos efectos: por una parte, los algoritmos cuyo error decrece cuanto mayor sea el número de ejemplos utilizados para el train verán estimada de forma optimista su error producido. Por otra parte, esta estimación del error tendrá una mayor variabilidad [13]. En la literatura del tema se conocen estos dos efectos como *bias* y *varianza*, en varios trabajos recientes se han estudiado de forma rigurosa estos efectos [6][52][57] si bien no es un problema nuevo [33][34][66].

Algunos autores [24] proponen utilizar una estrategia determinista para realizar las particiones del conjunto de ejemplos, con objeto de que las particiones sean homogéneas y contengan ejemplos lo más diversos posible. Con esto se consigue eliminar la variabilidad en la estimación del error que se produce en algoritmos ”inestables” [14]. Esta alternativa determinista permite repetir una experimentación sin necesidad de conocer las particiones del conjunto de ejemplos. La cuestión de la repetitibilidad de una experimentación ha sido también estudiada en [11], analizando distintos tipos de tests estadísticos y diseños experimentales.

Existen más variaciones de la validación cruzada. La técnica *complete cross validation* [46] utiliza todas las posibles particiones del conjunto de ejemplos con un tamaño dado, lo que mejora la estimación del error de generalización. En este caso es posible reducir el número de particiones, con la ayuda de diferentes criterios [53][54]. *Leave one out* [49][26] es el caso extremo en que cada conjunto de test contiene un único elemento.

A.1 Tests empleados en combinación con la validación cruzada:

En condiciones bastante generales, podemos afirmar que el objeto de la comparación de dos algoritmos es decidir si el valor medio de su medida de error sobre la población completa coincide, o es distinto [4].

Si se ha seguido el diseño *multifold cross validation*, se dispone de k estimaciones del error de cada algoritmo, como resultado de evaluarlo sobre cada uno de los conjuntos T_i. Ese conjunto de valores puede considerarse, a su vez, como una muestra de k realizaciones independientes de una variable aleatoria “error muestral”, asociada al algoritmo. Si se desea contrastar que dos algoritmos de aprendizaje son distintos, es válido definir como hipótesis nula del contraste la afirmación “Las dos muestras de errores proceden de poblaciones con medias iguales”. Si los dos algoritmos se han probado sobre las mismas particiones, los datos están apareados y las dos muestras de errores pueden restarse elemento a elemento, con lo que la hipótesis nula equivalente sería “La diferencia entre los errores muestrales de ambos algoritmos tiene media cero”.

Si las muestras fuesen normales, el test escogido sería el test t [19]. Dado que ninguno de los parámetros de la población de errores muestrales es conocido, el número de grados de libertad del estadístico t sólo depende de que las muestras estén apareadas y de que las varianzas de las poblaciones sean iguales o distintas; esto último suele decidirse mediante un test F [72].

Uno de los contrastes de bondad de ajuste más utilizados es el de Kolmogorov-Smirnov [15], aunque es conocido que, si la media y la varianza de la población son estimadas a partir de la muestra, como es el caso en este diseño, el test es conservador; la tendencia actual es usar en su lugar el test de Shapiro-Wilk [71] o bien el test omnibus de D’Agostino-Pearson [1].

Si alguna de las muestras no es normal, el test t no es aplicable, y debe recurrirse a contrastar la hipótesis de que las medianas de las distribuciones del error son iguales, mediante un test no paramétrico. Si las muestras están apareadas, puede emplearse un test de signos para la mediana de las diferencias o bien un test Wilcoxon [81]. Para muestras no apareadas, uno de los más frecuentes es el de Mann-Whitney [51].

Como resumen, en la figura 1 se muestra un esquema con todas las decisiones que se deben tomar cuando se comparan dos algoritmos mediante validación cruzada, en lo que se refiere al test que debe utilizarse.

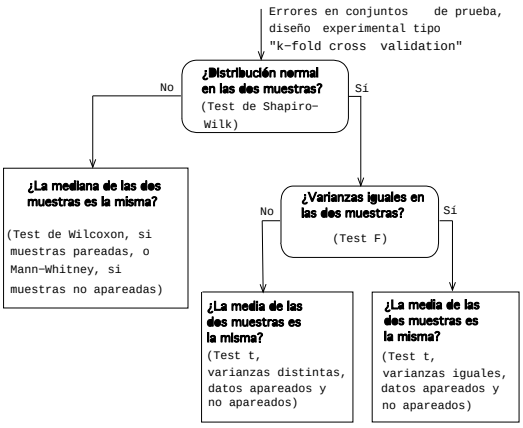

Fig. 1. Esquema de los test realizados en el diseño experimental tipo "validación cruzada".

B. *Otros diseños experimentales y tests estadísticos*

En este apartado se pasa revista brevemente a otros diseños experimentales menos frecuentes, que han caído en desuso entre la comunidad científica tras determinados estudios empíricos [25] o que se restringen a un software especifico [80].

- 5x2cv y 5x2cv-f En [25] se analizó el comportamiento del método *k-fold cross validation*, combinado con el empleo de un test t. En ese trabajo se puso de manifiesto que, dado que en el numerador del estadístico t aparece la media de las diferencias del error entre los dos algoritmos, y en el denominador la varianza, cuando la estimación de la varianza era moderadamente baja, una mala estimación de la media provocaba picos en los valores del estadístico t.

Dietterich propuso en ese trabajo sustituir el numerador del estadístico por la diferencia en el error de uno sólo de los experimentos (en lugar de la media de todos ellos) y justificó que es más efectivo realizar $k/2$ ejecuciones de un test *2-fold cross validation*, con diferentes permutaciones de los datos, que realizar un test *k-fold cross validation*. Como solución de compromiso entre la potencia del test y el tiempo de cálculo, propone realizar 5 ejecuciones de un test de validación cruzada con $k = 2$, de ahí el nombre 5x2cv. Los resultados de las 5 permutaciones se combinan mediante el estadístico 5x2cv-t, definido por el mismo autor, que sigue una distribución t con 5 grados de libertad.

Con posterioridad a la definición del diseño experimental 5x2cv, en [3] propuso reemplazar el estadístico 5x2cv-t por una variante que no dependiese del orden en que se realizasen los experimentos. El nuevo estadístico se denominó 5x2cv-f, ya que sigue una distribución $F_{10,5}$. En el mismo estudio se justificó también que el test 5x2cv-f es más potente que el 5x2cv-t bajo ciertas condiciones.

- Resampling: muestrea los datos disponibles en el conjunto de ejemplo D, n veces, obteniendo cada vez un conjunto de test $D_{t,i}$ y un conjunto de train $D - D_{t,i}$. Los algoritmos A y B se entrenan sobre $D - D_{t,i}$ y su precisión se mide utilizando $D_{t,i}$, obteniendo los valores $P_{A,i}$ y P_B, i, respectivamente. Se construye una muestra de tamaño n con los valores $x_i = P_{A,i} - P_B, i, i \in [1, n]$ y se aplica un test-t. Este test posee un elevado error de Tipo I [25] aunque puede corregirse [12].
- Repetir k-fold cross validation: si se desea obtener un mayor número de muestras se puede repetir k-fold cv r veces, generando de forma aleatoria distintas particiones cada vez. De esta forma se obtienen una muestra de tamaño $r \times k$ de la diferencia en precisión.
- Promediado en folds: es el método recomendado en Weka [80]. Se repite k fold cv y se obtiene un único valor para la diferencia en precisión a partir de cada ejecución de cv, promediando los k valores obtenidos en cada una de ellas. Este diseño experimental presenta un elevado error de Tipo I si se usa un test t según [10].
- Promediado en ejecuciones: similar al anterior pero se promedian los resultados obtenidos en el fold $j \in [1, k]$ en las distintas ejecuciones $i \in [1, r]$. Este diseño experimental también presenta un elevado error de Tipo I si se usa un test t según [10].
- Repetir k-fold cross validation manteniendo particiones y variando semilla: algunos investigadores [2] utilizan este diseño experimental con el fin de tener en cuenta el efecto de la semilla con la que se inicializa el generador de números aleatorios. Según [25] este es uno de los efectos a analizar en problemas de ML si bien en otros trabajos se propone otro tipo de diseños experimentales cuando se trata de analizar el efecto (en general) de distintos parámetros en un algoritmo [5] o del efecto de la semilla en particular [21].
- Test bootstrap tras cross validation: en [69] se realizó un estudio empírico que comparaba 5x2cv, 5x2cv-f, los tests clásicos combinados y dos tests basados en bootstrap. Se compararon los tests en dos aspectos distintos: frente a la dificultad del problema y frente al número de folds (excepto 5x2cv y 5x2cv-f, como es natural). A la vista de los resultados de la primera comparativa, el orden de preferencia (mejor a peor) es bootstrap, 5x2cv, tests clásicos y 5x2cvf. En lo que se refiere a la segunda comparativa, en donde se realizaron pruebas con 10, 50 y 100 folds, no parece existir una ganancia apreciable entre utilizar 50 o 100 folds.

C. *Consideraciones finales al respecto de la validación cruzada.*

En [37] se realizó un estudio empírico comparativo en el que se utilizaba el problema propuesto en [36] como benchmark de los diseños experimentales y tests aquí expuestos. Este estudio se amplió en [69] incluyendo en la comparativa dos tests basados en remuestreo [27][22], uno de ellos basado en permutaciones [18] y otro basado en "Tilted EDFs". En las conclusiones del trabajo mencionado derivadas de la comparativa realizada, se propone reemplazar los tests clásicos combinados por el test basado en remuestreo y "Tilted EDFs". Los tests combinados presentaban una potencia alta pero una tasa de error de Tipo I muy elevada, del mismo orden que 5x2cv o 5x2cv-f. Por el contrario, los tests basados en remuestreo, en concreto los basados en "Tilted EDFs" mostraban una potencia comparable a los anteriores pero un error de Tipo I excepcionalmente bajo, recomendándo-

se por lo tanto el uso de estos tests.

III. Test de comparaciones múltiples, estudio bibliográfico.

Estos tests se usan cuando el investigador desea confrontar su método con los más representativos del estado del arte. En ese caso puede plantear varios test de hipótesis simultáneos. Por ejemplo, para n algoritmos:

$$H_{0i} : \mu_1 = \mu_i$$
$$H_{1i} : \mu_1 \neq \mu_i$$
$$i \in 2..n$$

en donde el algoritmo con el índice 1 se compara con el resto de los n-1 algoritmos.
Es importante destacar que, lo que finalmente pretende al investigador, es poder realizar una afirmación compuesta por los resultados de todos los tests (con una determinada seguridad) como la siguiente: "el algoritmo 1 posee distinto error que 2, igual que el 3,...etc".
Sin embargo, cuando se realiza un test múltiple, se pone de relieve el efecto multiplicativo del error de tipo I: si en cada test la probabilidad de cometer un error de tipo I (rechazar la hipótesis nula cuando es cierta, en nuestro caso encontrar diferencias cuando no existen) es α, la probabilidad de no cometer ninguno en n tests es $(1-\alpha)^n$, luego la probabilidad de cometer *al menos* un error de tipo I en el conjunto de tests es $\alpha_t = 1-(1-\alpha)^n$. Si el número de experimentos que se comparan es r, en total se tienen $n = r*(r-1)/2$ tests individuales. Este efecto, que es bien conocido en estadística [43], [40], suele ignorarse en aprendizaje de máquina [67].
Existen textos recientes dedicados enteramente a este tema [43], [40] incluso en combinación con remuestreo [84] o en el marco de determinado software [85] en los cuales se pueden encontrar recopilados la mayor parte de los métodos que se citan en este artículo. También se pueden consultar algunas monografías sobre este tema como [70], [65], [20], [83]. Más relacionados con el área de Machine Learning están [61], en el que se propone una metodología para comparar varios algoritmos y [23] en donde lo que se propone es como comparar *dos algoritmos* sobre *varios datasets*. En [68] se presenta un estudio empírico preliminar sobre este tipo de tests, si bien el estudio debe de ser completado analizando todas las posibles configuraciones de la hipótesis nula, en el marco de una experimentación en Machine Learning, tal y como se propone en [28] de modo genérico.
Existen dos alternativas clásicas, propuestas por Fisher [30] a la hora de abordar este problema, los métodos de dos pasos y los métodos de un paso junto con una tercera clase de métodos, más reciente, denominados multipaso por algunos autores [40], secuenciales por otros [83] o en función de las comparaciones realizadas [43].

A. *Métodos de dos pasos.*

Los métodos de dos pasos *intentan* proteger el test de comparaciones múltiples frente al error de tipo I mediante un test F previo (ANOVA, análisis de la varianza, o alguna de sus contrapartidas paramétricas), en el cual se contrasta si existe alguna diferencia entre dos de los algoritmos al menos. Para r experimentos las hipótesis nula y alternativa de este test F previo serían:

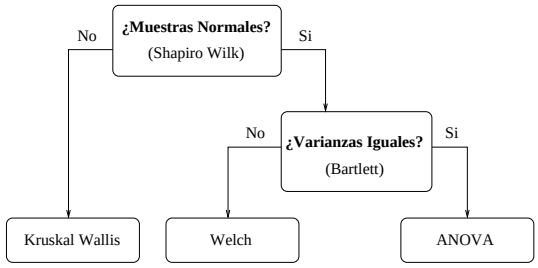

Fig. 2. Esquema de los test necesarios para escoger el test F adecuado a las condiciones paramétricas de las muestras.

$$H_0 : \mu_1 = \mu_2 = ... = \mu_r$$
$$H_1 : \mu_i \neq \mu_j \; para \; algún \; i, j \in 1..r$$

El test ANOVA se puede enmarcar dentro de lo que se conoce como "General linear models" o GLM y en concreto, tal y como se suelen usar en ML, en los "General linear univariate models" o GLUM. Finalmente, los "Generalized linear models" o GLZ [59] son la categoría mas amplia en la que se puede enmarcar el ANOVA, ya que en este caso puede escogerse modelizar la variable de respuesta mediante una distribución distinta a la Normal.
El diseño experimental *multiple fold cross-validation* encaja en la terminología del análisis de la varianza de la siguiente forma [64]: cada uno de los algoritmos contrastados se considera un *nivel* del *factor* del cual se quiere averiguar si tiene *efecto* (incidencia) significativo sobre el resultado, en este caso el error cometido. Más aún, el modelo se dirá fijo y de un sólo factor, es decir, sólo se contrasta si la elección del algoritmo incide o no de forma significativa en el error obtenido para el catálogo (prefijado, no escogido al azar) de algoritmos analizado.
En función de cuales de las siguientes condiciones: normalidad, homocedasticidad e independencia se verifiquen para las distribuciones de las muestras del error, se podrá aplicar uno de los siguientes tests:

- Todas: Análisis de la varianza (ANOVA según sus siglas en Inglés).
- Normalidad e independencia: test de Welch.
- Independencia: test de Kruskal Wallis.
- Ninguna: test de Q de Cochran.

En la figura 2 se muestra el encadenamiento los tests más frecuentemente utilizados para la verificación de las condiciones paramétricas de los tests mencionados más arriba. Si se rechaza la hipótesis nula (existen diferencias significativas), se realizan los test de comparaciones múltiples *escogiendo el mismo nivel de significación que para el test F*. A este tipo de planteamiento corresponde el tests LSD de Fisher [18]. Es importante resaltar que la protección frente a errores de tipo I se produce sólo

bajo la hipótesis nula, es decir, frente al hecho de encontrar diferencias entre algún resultado, cuando realmente no existen, en el test F que se aplica en primer lugar. Sin embargo esto no ocurre en el caso de los tests individuales [40], [43], una vez rechazada la hipótesis nula, el test de comparaciones múltiples no está protegido frente al error de tipo I al mismo nivel que el test F, es decir la probabilidad de encontrar diferencias que no existen entre dos algoritmos es mayor que el nivel escogido para la afirmación compuesta.

B. *Métodos de un paso.*

Los métodos denominados de un paso tienen en cuenta el efecto multiplicativo del error de tipo I y ajustan los niveles de significación (o los p-valores) de cada test individual con el fin de asegurar que en el test múltiple no se excede determinada probabilidad de ese error. De este modo, si se fija una probabilidad de error de tipo I para el test global de α_t, se tiene para cada test individual $\alpha = 1 - (1 - \alpha_t)^{1/n}$ (siendo n el número de comparaciones), lo cual se conoce como el ajuste de Dunn-Sidák [76]. Si se aproxima $(1 - \alpha)^n$ por $1 - n\alpha$ se tiene $\alpha = \alpha_t/n$ lo que se conoce como el ajuste de Bonferroni [9].

C. *Métodos secuenciales.*

Lamentablemente la potencia de un test (1-probabilidad de error de tipo II) decrece junto con α, lo cual hace que los mencionados ajustes sean conservadores, es decir, tienden a no rechazar la hipótesis nula [83]. Por este motivo se ha desarrollado el tercer tipo de métodos mencionado al principio de esta sección (métodos secuenciales) que permiten incrementar la potencia de los tests múltiples. Entre estos ajustes se encuentran el de Holm [41], el de Simes-Hochberg [77], [39] o el de Hommel [42]. En esencia se trata de ordenar los p-valores obtenidos en los tests individuales y comenzando por un extremo de la ordenación, aplicar el ajuste de Bonferroni o el de Dunn-Sidak al test actual, rechazando o aceptando la hipótesis correspondiente. A continuación, se elimina esa hipótesis del test múltiple, actualizando por lo tanto el número de tests (que determina el nivel de significancia de los tests individuales) y se procede con el test múltiple resultante del mismo modo.
Finalmente, en una experimentación de aprendizaje de máquina, no sólo es importante limitar la probabilidad de que se rechace incorrectamente al menos una hipótesis nula cierta, sino que además es importante controlar su número. En este sentido, en [7], se propone una técnica para controlar esta magnitud, que los autores denominan "False Discovery Rate" o FDR por sus siglas. Recientemente en [79] y [29] han aparecido sucesivas modificaciones de esta técnica.

D. *Consideraciones finales sobre los tests de comparaciones múltiples*

Del estudio empírico realizado en [68] y de las fuentes bibliográficas consultadas, parece evidente que la realización de un test ANOVA previo no protege al test múltiple frente al error de Tipo I. Por este motivo se recomienda la utilización de un ajuste de p-valores en combinación con el test basado en remuestreo recomendado a su vez en la sección anterior. De entre los ajustes comentados, el de Holm es válido independientemente de las condiciones paramétricas de las muestras y más potente que el ajuste de Bonferroni. Los ajustes de Hochberg y Hommel requieren independencia.

IV. Comparación de algoritmos multiobjetivo.

Lo que hemos expuesto hasta el momento no es válido para el caso de los algoritmos multiobjetivo, ya que si lo que se pretende es obtener una aproximación del frente de Pareto, entonces no se obtiene una única solución al problema de optimización sino un conjunto de soluciones que constituyen precisamente la aproximación buscada [48]. Por este motivo es necesario escoger una métrica adecuada del rendimiento de los algoritmos [47][60][90]. Cuando los algoritmos comparados son estocásticos, al igual que en el caso de los algoritmos vistos en secciones anteriores, la medida de la precisión que se use se podría tomar como una variable aleatoria. Existen tres aproximaciones a este problema [48]. La primera de ellas se basa en la obtención de indicadores a partir de cada aproximación del frente de Pareto obtenido, para cada ejecución y algoritmo, con el fin de obtener de una muestra del indicador para cada algoritmo [87]. La segunda se basa en la estimación de la denominada "Attainment Function" a partir del conjunto de aproximaciones obtenido [32] por cada algoritmo. La tercera consiste en puntuar cada aproximación obtenida por cada algoritmo en cada ejecución en función de la dominancia y aplicar el test adecuado a las muestras de rangos obtenidas [31][35]. A continuación se examina cada una de ellas, siguiendo la notación utilizada en [48], en la que se considera la comparación de $q \geq 2$ algoritmos estocásticos, ejecutándose cada uno de los $i \in 1, ..., q$ algoritmos un cierto número de veces $r_i \geq 1$, obteniéndose las aproximaciones del frente de Pareto $A_1^1, A_2^1, ..., A_1^q, ..., A_{r_q}^q$, en donde los subíndices denotan el algoritmo y los superíndices la ejecución.

A. *Ranking basado en dominancia*

Esta aproximación se basa en agrupar la colección completa de aproximaciones del frente de Pareto $A_1^1, A_2^1, ..., A_1^q, ..., A_{r_q}^q$ y ordenarlas utilizando algún criterio basado en la definición de dominancia [31][35]. Por ejemplo, en [31] se entiende que una aproximación A es mejor que otra B ($A \lhd B$) cuando todos los elementos de B son dominados al menos por un elemento de A y A no es indiferente a B. La indiferencia ocurre cuando A domina débilmente a B y viceversa. Utilizando la relación $A \lhd B$ se ordenan las aproximaciones asignando a cada una el rango $1 + |\{C_j \in C : C_j \lhd C_i\}|$. De esta forma se transforman las muestras de aproximaciones al frente de Pareto en muestras de rangos, de modo que se puede testear si la distribución de rangos es significa-

tivamente distinta entre los algoritmos utilizando el test de Mann-Whitney (en el caso de dos algoritmos) o el de Kruskal-Wallis [68][48] para $q > 2$.

B. Comparación basada en la "Attainment Function"

El resultado de la ejecución de un algoritmo multiobjetivo estocástico, entendido como un conjunto de vectores objetivo, seguirá una cierta *distribución*. El número de vectores objetivo proporcionados por el algoritmo puede ser, en si mismo, una variable aleatoria. La "Attainment Function" está basada en el concepto de alcanzar una meta. En este contexto un vector de objetivos se alcanza cuando es débilmente dominado por la aproximación del frente de Pareto que proporciona el algoritmo. Se define como una función del espacio objetivo en el intervalo $[0, 1]$ que hace corresponder a cada vector objetivo la probabilidad de que este sea alcanzado en una ejecución del algoritmo. Esta función se puede estimar de forma empírica a partir de r ejecuciones de un algoritmo, mediante la ecuación 1. En dicha ecuación, A^i es la i-ésima ejecución del algoritmo, $I(.)$ es 1 si su argumento es cierto, 0 en caso contrario y z es el vector objetivo argumento de $\alpha(.)$.

$$\alpha_r(z) = \frac{1}{r}\sum_{i=1}^{r} I(A^i \preceq \{z\}) \tag{1}$$

Esta estimación puede usarse para comparar dos algoritmos mediante el test estadístico apropiado, en [75] se propone la utilización del test de Kolmogorov-Smirnov con el fin de determinar si existen diferencias significativas entre dos algoritmos, pero no proporciona información sobre cual es el mejor.
La información sobre la dispersión de los resultados puede representarse de forma gráfica representando el subconjunto de los vectores objetivo que se han alcanzado un determinado % del total de r repeticiones. Este subconjunto puede representarse a su vez por la superficie que divide el espacio objetivo en dos partes, aquella que se alcanza con la frecuencia obtenida de forma empírica o mayor y la que no se ha alcanzado con esa frecuencia.
Si la mencionada representación se aplica al frente de Pareto combinado obtenido a partir de dos algoritmos, las diferencias entre ambos algoritmos pueden representarse gráficamente mediante niveles de gris, proporcionales a la diferencia de frecuencias con las que se alcanza cada punto por cada algoritmo [50].

C. Comparación basada en indicadores de calidad unarios

Un indicador de calidad unario de un algoritmo multiobjetivo es una función $f : \Omega \mapsto \mathbb{R}$ que asigna un número real a cada aproximación del frente de Pareto [48]. Es deseable que el indicador proporcione una estructura de preferencia, de modo que sea preferible utilizar un algoritmo A en lugar de otro B si $I(A) > I(B)$ y que la diferencia en el valor de los indicadores esté relacionada con la diferencia en la calidad del resultado de los algoritmos correspondientes. Existen distintas definiciones de indicadores [90][38], a modo de ejemplo se explicará el indicador hipervolumen I_H [87]. Sea A el subconjunto de vectores del espacio objetivo que define la aproximación al frente de Pareto obtenida por un algoritmo. Sea Z' el subconjunto del espacio de objetivos dominado débilmente por A (cualquier vector de Z' está dominado débilmente por lo menos por un elemento de A). Entonces I_H es el hipervolumen de Z', con respecto a un punto de referencia, escogido de forma arbitraria, en el espacio de objetivos. Cuanto mayor sea I_H, mejor será el conjunto de aproximación. Este indicador posee la siguiente propiedad, si $A \lhd B$ entonces $I_H(A) > I_H(B)$. Por lo tanto si $I_H(A) < I_H(B)$ entonces A no puede ser mejor que B.
Si se convierten las muestras de aproximaciones al frente de Pareto $A_1^1, A_2^1, ..., A_1^q, ..., A_{r_q}^q$ en muestras de números reales utilizando un indicador $I_H(A_1^1), i_H(A_2^1), ...,$ $I_H(A_1^q), ..., I_H(A_{r_q}^q)$, la comparación de algoritmos utilizando un test estadístico es directa.
Finalmente, según [90], las comparaciones basadas en indicadores de calidad unarios tienen serias limitaciones ya que no pueden describir de forma completa la calidad de una aproximación del frente de Pareto y por lo tanto del algoritmo que la produce. Sin embargo pueden ser útiles cuando se comparan aproximaciones de Pareto basándose en algún criterio de preferencia [38]. En el siguiente apartado se comentan los indicadores de calidad binarios, que solucionan los problemas comentados.

D. Comparación basada en indicadores de calidad binarios

Un indicador de calidad binario de un algoritmo multiobjetivo es una función $f : \Omega \times \Omega \mapsto \mathbb{R}$ que asigna un número real a dos aproximaciones del frente de Pareto producidas por dos algoritmos distintos [48]. Es fácil concluir que cuando se comparan más de dos algoritmos se obtendrán más indicadores binarios que unarios, puesto que se deben obtener todos los emparejamientos posibles y el indicador puede no ser conmutativo. Por ejemplo en [89] se define el indicador $I_C(A, B)$ que devuelve la fracción de soluciones en B que son dominadas al menos por una solución en A. En [88] se propone una versión binaria del indicador I_H, denominada I_{H2}. Este indicador devuelve el hipervolumen del subespacio dominado débilmente por A pero no por B. En la literatura se han definido más indicadores de calidad binarios que no aparecen reflejados en este trabajo por cuestiones de espacio, véase [90]. Sin embargo ni en [90] ni en [48] aparece de forma clara la forma de utilizar los indicadores binarios para comparar algoritmos multiobjetivo estocástico, es más, en [48] la experimentación se hace con indicadores unarios.

E. Consideraciones finales acerca de las comparaciones de algoritmos multiobjetivo

En esta sección se han explicado las distintas alternativas disponibles para comparar algoritmos multiobjetivo. De entre las alternativas presentadas, sólo la comparación basada en dominancia proporciona resultados susceptibles de ser tratados con rigor mediante los tests es-

tadísticos adecuados, respondiendo además a la pregunta de cuál es el mejor algoritmo. En la literatura consultada no se ha encontrado una experimentación realizada utilizando indicadores de calidad binarios. Por este motivo se recomienda el uso de la alternativa que utiliza la dominancia para comparar algoritmos.

V. CONCLUSIONES

En este trabajo se ha tratado de presentar una panorámica de los métodos más extendidos para la realización de una tarea a la que se enfrenta cualquier investigador a la hora de presentar un trabajo, la comparación entre algoritmos. Como se ha podido observar, no existe un consenso definitivo en la comunidad científica sobre este tema, ni siquiera sobre si tiene o no sentido proclamar que un algoritmo es mejor que otro. Sin embargo existen algunos trabajos en los que se analizan en profundidad las metodologías existentes y de ellos se pueden extraer algunas recomendaciones, que se han ido desgranando al final de cada sección. Este trabajo se cimenta en la experimentación o el análisis de terceros, es evidente que sería de gran interés enfrentar a las distintas metodologías a los algoritmos y problemas más habituales, con el fin de determinar si son realmente útiles en la tarea para la que han sido diseñadas. Por cuestiones de tiempo y espacio no se pueden presentar dichos experimentos aquí, si bien es el propósito de los autores emprender esta tarea en el futuro.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia de España, en el marco del proyecto TIN2005-08386-C05.

REFERENCIAS

[1] D'Agostino, R. B. and Stephens, M. A., eds. Goodness-of-fit Techniques. New York: Dekker (1986)

[2] R. Alcalá, J. Alcalá-Fdez, M. J. Gacto y F. Herrera. Obtención de Sistemas Basados en Reglas Difusas Precisos y Compactos mediante Algoritmos Genéticos Multiobjetivo. XII Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF'06. Ciudad Real. 20-22 Septiembre 2006

[3] Alpaydin E.: Combined 5x2cv-F test for Comparing Supervised Classification Learning Algorithms. Neural Computation 11 (1999) 1885-1892

[4] Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., Stewart Jr., W. R.: Designing and Reporting on Computational Experiments with Heuristic Methods. Journal of Heuristics, 1 (1995) 9-32

[5] Thomas Bartz-Beielstein. Experimental Analysis of Evolution Strategies: Overview and Comprehensive Introduction. Reihe CI 157/03. SFB 531, Universität Dortmund. Dortmund, Germany. 2003.

[6] Sergey V. Beiden, Marcus A. Maloof, Robert F. Wagner. A General Model for Finite-Sample Effects in Training and Testing of Competing Classifiers. IEEE PAMI, December 2003 (Vol. 25, No. 12) pp. 1561-1569.

[7] Benjamini, Y., Hochberg, T. 1995. Controlling the False Discovery Rate: a practical and powerful approach to multiple testing. J. Royal Stat. Soc. B 85: 289300.

[8] Blake, C.L., Merz, C.J. UCI Repository of machine learning databases. http://www.ics.uci.edu. University of California, Department of Information and Computer Science (1998)

[9] Bonferroni C.E. Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 8:3-62. 1936.

[10] R. R. Bouckaert. Choosing between two learning algorithms based on calibrated tests. En T. Fawcett y N. Mishra, eds, Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA. AAAI Press, 2003.

[11] R. R. Bouckaert. Estimating replicability of classifier learning experiments. En C Brodley, ed., Machine Learning, Proceedings of the Twenty-First International Conference (ICML 2004). AAAI Press, 2004.

[12] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. Journal of Machine Learning Research, 5:10891105, 2004.

[13] Bradford J. P.: Brodley C. E.: The effect of Instance-Space Partition on Significance. Machine Learning 42 (2001) 269-286

[14] Breiman, L.: Bagging predictors. Machine Learning 24 (1996) 123-140

[15] Chakravarti, Laha, and Roy. Handbook of Methods of Applied Statistics, Volume I, John Wiley and Sons, (1967). 392-394.

[16] Cochran W.G., Cox G.M. Experimental Designs. Wiley (1992)

[17] A Short Tutorial on Evolutionary Multiobjective Optimization. Carlos A. Coello. Eckart Zitzler et. al. eds. First International Conference on Evolutionary Multi-Criterion Optimization, pages 21-40. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001

[18] Cohen, P.R., Empirical Methods for Artificial Intelligence. MIT Press (1995)

[19] Cox, D.R., Hinkley, D.V. Theoretical statistics. London. Chapman & Hall (1974)

[20] Curran-Everett D. Multiple comparisons: philosophies and illustrations. Am J Physiol Regul Integr Comp Physiol 279: R1-R8, 2000.

[21] Andrew Czarn, Cara MacNish, Kaipillil Vijayan, Berwin Turlach, and Ritu Gupta. Statistical Exploratory Analysis of Genetic Algorithms. IEEE Transactions on Evolutionary Computation, VOL. 8, NO. 4, AUGUST 2004.

[22] Davison, A.C., Hinkley, D.V. Bootstrap Methods and Their Application. Cambridge University Press (1997).

[23] J. Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7 (2006) 1-30

[24] Diamantidis N. A., Karlis D., Giakoumakis E. A.: Unsupervised stratification of cross-validation for accuracy estimation. Artificial Intelligence 116 (2000) 1-16

[25] Dietterich, T.G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation 10 (7) (1998) 1895-1923

[26] Geisser, S: The Predictive Sample Reuse Method with Application. J. Amer. Stat. Ass. 70 (1975) 320-328

[27] Efron, B. y Tibshirani, R. An introduction to bootstrap. New York. Chapman & Hall.(1993)

[28] Einot, I. and Gabriel, K. R. (1975). A Study of the Powers of Several Methods of Multiple Comparison, Journal of the American Statistical Association, 70, page 351.

[29] Fernando, R.L., D. Nettleton, B.R. Southey, J.C.M. Dekkers, M.F. Rothschild, M. Soller. 2004. Controlling the proportion of false positives in multiple dependent tests. Genetics 166:611-619.

[30] Fisher, R. A. (1971). The design of experiments (9th ed.). New York, Hafner Publishing Company.

[31] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, In Stephanie Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[32] Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, pages 213-225. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[33] K. Fukunaga and R.R. Hayes, Effects of Sample Size in Classifier Design, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 8, pp. 873-885, Aug. 1989.

[34] K. Fukunaga and R.R. Hayes, Estimation of Classifier Performance, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 10, pp. 1087-1101, Oct. 1989.

[35] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA. 1989.

[36] Haykin, S. Neural Networks, A Comprehensive Foundation.Prentice Hall, 1999

[37] Herrera F., Hervás C., Otero J., Sánchez L.. Un estudio empírico preliminar sobre los tests estadísticos más habituales en el aprendizaje automático. Tendencias de la Minería de Datos en España 403-412. Lecture Notes in Computer Science. ed. Digital @3D.

[38] Michael Pilegaard Hansen and Andrzej Jaszkiewicz, Evaluating the quality of approximations to the non-dominated set. IMM-REP-1998-7. 1998.

[39] Hochberg, Y. A sharper Bonferroni procedure for multiple tests of significance. Biometrika, 75: 800-803. 1988.

[40] Hochberg, Y., Tamhane A. C., Multiple comparison procedures, 1987,John Wiley & Sons, Inc.

[41] Holm S. A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics, 6, 65-70. 1979.

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

[43] Hsu, J.C. (1996). Multiple Comparisons: Theory and Methods, Chapman and Hall,London.

[44] Holte R. C.: Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning 11(1) 1993 63-90

[45] David S. Johnson. A Theoretician's Guide to the Experimental Analysis of Algorithms. Proceedings of the 5th and 6th DIMACS Implementation Challenges. Goldwasser, Johnson and McGeoch (eds). American Mathematical Society (2002).

[46] Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of International Joint Conference on Artificial Intelligence (1995)

[47] J. Knowles y D. Corne. On metrics for comparing non-dominated sets. Congress on Evolutionary Computation (CEC 2002). Hilton Hawaiian Village Hotel Honolulu, Hawaii. May 12-17 2002

[48] Knowles, Joshua y Thiele, Lothar y Zitzler, Eckart. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. KTZ2006a. feb 2006.

[49] Lachenbruch, P.A., Mickey M. R.:Estimation of Error Rates in Discriminant Analysis, Technometrics 10 (1968)

[50] M. L. López-Ibañez, L. Paquete, and T. Stützle.Hybrid population-based algorithms for the bi-objective quadratic assignment problem. Journal of Mathematical Modelling and Algorithms, 5(1):111-137, 2006.

[51] Mann, H. B. and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Statist. 18, (1947) 50-60.

[52] Markatou, M, Tian, H, Biswas, S. and Hripcsak, G. Analysis of Variance of Cross-Validation Estimators of the Generalization Error. Journal of Machine Learning Research, 6, 1127-1168.(2005).

[53] Mitchell, T. Machine Learning. McGraw Hill (1997)

[54] Mullin M., Sukthankar R.: Complete Cross-Validation for Nearest Neighbor Classifiers. Proceedings of the International Conference on Machine Learning (2000)

[55] McGeoch, Catherine C. A Bibliography of Algorithm Experimentation. DIMACS Series in Discrete Mathematics and Thoretical Computer Science. 2002.

[56] McGeoch, Catherine C. Experimental analysis of Algorithms. Notices of the AMS. 48(3) (2001) 304-311.

[57] Claude Nadeau, Yoshua Bengio.Inference for the Generalization Error. Machine Learning, 52, 239281, 2003.

[58] O'Brien PC and Shampo MA: Statistical considerations for performing multiple tests in a single experiment. Mayo Clinic Proceedings, 63:813-815, 1988.

[59] Nelder, J. A. and R. W. M. Wedderburn (1972). Generalized liner models. Journal of the royal Statistical Society, A, 135: 370-384.

[60] Tatsuya Okabe, Yaochu Jin and Bernhard Sendhoff. A Critical Survey of Performance Indices for Multi-Objective Optimization, in Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003), Volume 2, pp. 878–885, IEEE Press, Canberra, Australia, December 2003

[61] J. Pizarro, E. Guerrero, and P. L. Galindo. Multiple comparison procedures applied to model selection. Neurocomputing, 48:155-173, 2002.

[62] Ruiz-Maya, L.: Métodos Estadísticos de Investigación (Introducción al Análisis de la Varianza), Instituto Nacional de Estadística. (1986)

[63] Tjen-Sien Lim, Wei-Yin Loh, Yu-Shan Shih. A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. Machine Learning 40(3) (2000) 203-228

[64] Lindman H.R. Analysis of variance in experimental design.Springer-Verlag (1992)

[65] Rafter J.A., Abell M. L., James P. Braselton. Multiple Comparison Methods for Means. SIAM Review Volume 44, Number 2 pp. 259-278. 2002 Society for Industrial and Applied Mathematics

[66] S.J. Raudys and A.K. Jain, Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, pp. 252-264, 1991.

[67] Salzberg S. L. On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. Data minig and Knowledge Discovery 1 (1997) 317-328

[68] Sánchez L., Otero J., Alcalá J. Un estudio empírico preliminar acerca del uso de los tests de comparaciones múltiples en aprendizaje automático. Actas del congreso MAEB05. Granada. Spain. 2005.

[69] Sánchez L., Otero J., Alcalá J.Assesing the differences in accuracy between GFSs with bootstrap tests. EUSFLAT 2005. Barcelona

[70] Sarkar S.K., Temple University, Philadelphia, Pennsylvania, U.S.A. Research Report.

[71] Shapiro, S. S. and Wilk, M. B. An analysis of variance test for normality (complete samples), Biometrika, 52, 3 and 4, (1965) 591-611

[72] Snedecor, G. W., Cochran, W. G.: Statistical Methods. Iowa State University Press, Ames, IA. (1989)

[73] Schaffer, C.: A conservation law for generalization performance. In Proceedings of the 1994 International Conference on Machine Learning (1994)

[74] Stone, M.: Cross-validatory choice and assesment of statistical predictions. J. Roy. Statist. Soc. 36 (1974) 111-147

[75] K. J. Shaw, A. L. Nortcliffe, M. Thompson, J. Love, C. M. Fonseca, and P. J. Fleming. Assessing the Performance of Multiobjective Genetic Algorithms for Optimization of a Batch Process Scheduling Problem, In 1999 Congress on Evolutionary Computation, pages 37-45, Washington, D.C., July 1999. IEEE Service Center.

[76] Sidak Z, Rectangular Confidence Regions for the Means of Multivariate Normal Distributions, American Statistical Association, 62, 626-633. 1967.

[77] Simes RJ. An improved Bonferroni procedure for multiple tests of significance. Biometrika, 73, 751-754. 1986.

[78] Stone, M. Cross-validation: A review. Mathematische Operationsforschung Statischen, Serie Statistics, 9 (1978) 127-139

[79] Storey J.D. 2002. A direct approach to false discovery rates. Journal of the Royal Statistical Society Series B: 64: 479498.

[80] Ian H. Witten and Eibe Frank (2005), WEKA, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[81] Wilcoxon, F. Individual Comparisons by Ranking Methods. Biometrics 1, (1945) 80-83

[82] Wolpert, D.H.: On the Connection Between In-Sample Testing and Generalization Error. Complex Systems 6 (1992) 47-94

[83] Walsh B. Multiple Comparisons: Bonferroni Corrections and False Discovery Rates. Lecture Notes for EEB581. Department of Ecology and Evolutionary Biology. University of Arizona. May 2004.

[84] Westfall P.H., Young S.S. (1993) Resampling-based Multiple Testing. Wiley, New York.

[85] Westfall, P.H., Tobias, R.D., Rom, D., Wolfinger, R.D. und Hochberg, Y. (1999) Multiple comparisons and multiple tests using the SAS system. Cary, NC.

[86] Whitley D., Watson J.P., Howe A., Barbulescu L. Testing, Evaluation and Performance of Optimization and Learning Systems. Keynote Address: Adaptive Computing in Design and Manufacturing (2002)

[87] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Transactions on Evolutionary Computation, 3(4):257-271, November 1999.

[88] Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Shaker Verlag, Aachen, Germany, ETH Zurich, TIK-Schriftenreihe No. 30, December, 1999. ISBN 3-8265-6831-1

[89] Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study PPSN-V, Amsterdam, pages 292-301, September, 1998.

[90] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review, IEEE Transactions on Evolutionary Computation, Vol. 7, No. 2, pp. 117–132, April 2003 .