



# Toolkit for the Automatic Comparison of Optimizers (TACO): Herramienta *online* avanzada para comparar metaheurísticas

1<sup>st</sup> Daniel Molina

Departamento de Ciencias de la Computación e I.A.  
Universidad de Granada  
Granada, España  
dmolina@decsai.ugr.es

2<sup>nd</sup> Antonio LaTorre

DATSI, ETSIINF  
Centro de Simulación Computacional  
Universidad Politécnica de Madrid  
Madrid, España  
atorre@fi.upm.es

**Resumen**—El campo de las metaheurísticas es un campo en continua innovación, en donde los investigadores proponen frecuentemente nuevas propuestas, propuestas que tienen que ser adecuadamente comparadas con el estado-del-arte para probar su efectividad. El uso de benchmarks estándar permite comparar directamente los resultados de los algoritmos entre sí, pero incluso esas comparaciones suponen un proceso laborioso y repetitivo que consume gran parte del proceso de investigación en metaheurísticas. En este trabajo proponemos Toolkit for the Automatic Comparison of Optimizers, TACO, una herramienta *online* avanzada para comparar metaheurísticas, que además de almacenar los resultados de distintos algoritmos competitivos para distintos benchmarks (para usarlos como algoritmos de referencia), es capaz de realizar de forma automática una variedad de comparaciones. De hecho, permite comparar no sólo los algoritmos previamente almacenados, sino también los resultados de nuevos algoritmos sin necesidad de registrarse, e incluso para funciones específicas (no limitado a un benchmark concreto). Consideramos que es una herramienta muy útil para todo tipo de investigaciones en metaheurísticas. Como prueba de su conveniencia, ofrecemos una instancia de demostración de sus características, sobre la que se han realizado las distintas comparaciones que incluimos en este trabajo.

**Index Terms**—Comparativa de algoritmos, Herramienta web, Optimización Global de Alta Dimensionalidad.

## I. INTRODUCTION

El uso de metaheurísticas es muy común en problemas de optimización de diversa índole, ya que son algoritmos capaces de obtener muy buenos resultados en un tiempo limitado [1]. Este interés conduce a que se presenten un número continuo de propuestas cada año en la literatura científica. Un paso necesario para que la comunidad científica pueda considerar una propuesta es el de realizar una adecuada validación experimental en la que, para un conjunto de problemas (*benchmark*), se compare la propuesta presentada con distintos algoritmos, especialmente algoritmos del *estado-del-arte*.

Habitualmente, la comparativa implica realizar una misma experimentación tanto con la propuesta como con los algoritmos de referencia (usados para comparar). Afortunadamente, cada vez está más consolidado el uso de *benchmarks* específicos en los que se fijan a priori las condiciones experimentales, por lo que el problema se reduce a comparar los resultados de

la propuesta con los otros algoritmos publicados, simplificando el proceso. Sin embargo, realizar una correcta experimentación sigue siendo una tarea muy laboriosa y costosa en tiempo. El investigador no sólo tiene que recopilar los resultados de su propuesta, sino también los de los algoritmos con los que se va a comparar. Y la recopilación de datos es sólo el primer paso: los datos deben normalizarse (obtener la mismas medidas y con la misma frecuencia), agruparse (en funciones similares para su análisis), organizarse (en tablas), y analizarse (mediante una o varias medidas estadísticas o visualizaciones). Y, por cada cambio en alguno de los algoritmos (mejora de la propuesta, por ejemplo), estos pasos deben repetirse, incrementando el tiempo dedicado a las comparativas.

En este trabajo proponemos *Toolkit for the Automatic Comparison of Optimizers*, TACO, accesible en <https://tacolab.org> (con código disponible), una herramienta *online* avanzada que combina un repositorio públicamente accesible de resultados para distintos *benchmarks* con una herramienta de análisis modular que permite comparar fácilmente los resultados de distintos algoritmos, tanto para la fase de desarrollo del algoritmo como para la obtención de tablas y gráficas finales para la publicación del trabajo. Entre sus características más interesantes están:

- Soporte para distintos benchmarks, cada uno con sus propias propiedades: funciones asociadas y grupos de funciones, tamaño del problema (dimensión), hitos, etc.
- Diseño modular: La aplicación permite analizar según una serie de informes, que pueden ser generales, o ser específicos para un determinado *benchmark*.
- Repositorio de resultados de distintos algoritmos: La base de datos no sólo almacena estadísticas comunes (como media, mínimo, ...) que puedan recogerse de colecciones, sino que pueden almacenar todos los resultados para realizar comparaciones estadísticas que lo requieran.
- Definición flexible de informes: La aplicación ofrece una capa de abstracción de gráficas (líneas, barras, etc.) para permitir incorporar nuevos informes
- Exportación de datos. Para poder usarse en publicaciones

tanto las gráficas como las tablas se pueden guardar localmente para usarlas en distintos formatos. Las gráficas se pueden guardar en los formatos bitmap PNG y JPEG, en informes PDF, o formato vectorial SVG. Las tablas en formato tanto Excel (para posterior análisis) como Latex (para incluirlo directamente en publicaciones). En las tablas se puede elegir el número de cifras decimales que se desean obtener (aunque en Excel no es importante, en Latex sí lo es para poder insertarlo directamente en las publicaciones).

TACO está inspirada en una herramienta previa [2] pero resuelve muchas de sus limitaciones.

- No requiere que el usuario almacene previamente los resultados para comparar. Cualquier usuario (sin registrarse) puede ejecutar su análisis simplemente subiendo los resultados en un formato soportado, y comparar entre sí tanto los de la base de datos como los indicados por fichero.
- TACO permite que los usuarios identificados posean repositorios propios, con un catálogo de resultados de carácter privado. Los usuarios pueden solicitar hacerlos públicos cuando el *paper* esté publicado.
- Una tecnología y diseño mucho más flexible y modular, para ir incorporando mejoras.

Existen otras herramientas de análisis, discutidas en [3], como jMetal<sup>1</sup> o MOEA framework<sup>2</sup>. Sin embargo, en esos casos el proceso de análisis suele estar asociado con el propio entorno (*framework*) de optimización, limitando su uso. TACO se centra en comparar resultados ya obtenidos, sin interferir en la forma de programar/ejecutar el código.

Otro aspecto muy importante de la herramienta es que no sólo está pensada como herramienta de soporte a investigadores para comparar su propuesta con otros algoritmos, sino que también está ideada para organizadores de competiciones (de hecho, surgió a partir de esa necesidad). Gracias a su diseño modular, es bastante directo incorporar una nueva sesión especial o competición de manera sencilla. Además, se puede usar también para que los propios participantes comparen sus propuestas con los algoritmos de referencia de la competición. Una vez que la competición haya tenido lugar, se pueden hacer públicos los resultados de todos los algoritmos participantes.

Este trabajo está organizado de la siguiente manera: La sección II describe brevemente la arquitectura de la aplicación. En la sección III se muestra el esquema de funcionamiento. En la sección IV resumimos algunas opciones disponibles y mostramos unas capturas relevantes. La sección V repasa el diseño modular de informes, con ejemplos concretos. Finalmente, en la sección VI resumimos las principales conclusiones obtenidas.

## II. ARQUITECTURA

La aplicación web está construida como una página estática que mediante llamadas Ajax pide los datos dinámicos y

<sup>1</sup><https://jmetal.github.io/jMetal/>

<sup>2</sup><http://moeframework.org/>

los visualiza usando JavaScript. Para ello, usa las siguientes tecnologías webs:

- HTML5, CSS3 y Javascript (Vue.js<sup>3</sup> framework) en el *frontend*.
- El entorno Python Flask<sup>4</sup> y SQL Alchemy<sup>5</sup> como *backend*.
- Intercambio de datos entre ambos mediante peticiones AJAX y mensajes JSON.

El *framework* Vue.js es usado para dar una imagen dinámica de la web, pidiendo por AJAX toda la información (*benchmarks*, dimensionalidad, algoritmos, informes, etc.) al *backend* (con almacenamiento local para incrementar el rendimiento). Este diseño permite incorporar fácilmente la herramienta en otra aplicación web, desde páginas estáticas a sistemas CMS como Wordpress o similares.

## III. MODOS DE TRABAJO

Existen dos formas de trabajar, se pueden comparar usando los resultados de un benchmark estándar del sistema, o usando un conjunto de funciones (es muy útil para comparar para funciones distintas, o un subconjunto de funciones).

### III-A. Comparando con un benchmark propio

Este es el caso más sencillo, en el que se desea únicamente comparar entre sí distintos resultados que posee el usuario.

En este caso, el usuario debe ofrecer sus datos por medio de un fichero de entrada aceptado con los distintos resultados de cada algoritmo para un número de funciones. Un ejemplo sería un Excel con tantas columnas como funciones, con los resultados de cada algoritmo en su propia hoja del Excel. Adicionalmente se puede establecer una columna *milestone* para guardar resultados intermedios. Una vez subido el fichero, el usuario selecciona el tipo de comparación de entre las disponibles de forma genérica (descritos en la sección V) y procede al análisis.

### III-B. Comparando con un benchmark estándar

Una sesión típica empieza seleccionando el *benchmark* sobre el que se quiere comparar. Luego, el usuario decide el tamaño del problema (si hay más de uno), y selecciona de una lista de algoritmos cuáles quiere comparar. Además, el usuario puede subir un fichero con datos de algoritmos propios para incorporarlos a la comparativa. Una vez elegidos todos los datos, el usuario selecciona el tipo de comparación de entre las disponibles (descritas en la sección V) y procede al análisis.

El modelo de datos está diseñado en torno al concepto de *benchmark*. Cada *benchmark* está asociado con una tabla diferente en donde se almacenan los resultados de los algoritmos. Esto permite que distintos *benchmarks* puedan tener un distinto número de funciones de forma sencilla. Agrupada en torno a los benchmarks, se almacena la siguiente información:

- Dimensión: algunos *benchmarks* poseen un mismo tamaño del problema, y otros plantean para las mismas

<sup>3</sup><https://vuejs.org/>

<sup>4</sup><http://flask.pocoo.org/>

<sup>5</sup><https://www.sqlalchemy.org/>



funciones/problemas distintos valores, para medir la escalabilidad del algoritmo.

- Hitos: un *benchmark* puede almacenar los resultados finales o en distintos momentos de la ejecución del algoritmo. El sistema guarda información de cada hito.
- Grupos o categorías de funciones: Los *benchmarks* pueden tener problemas/funciones con distintas características, y eso permite analizar los resultados por grupos de funciones (como comparar separadamente comportamiento en problemas unimodales y multimodales).
- Informes: Aparte de los informes disponibles para todos los *benchmark*, un *benchmark* puede tener asociado alguno específico (en particular, en las competiciones de alta dimensionalidad se usa un ranking específico, de la F1). Se muestra, para cada *benchmark*, el conjunto de posibles informes asociados.

#### IV. FRONTEND

Las Figuras 1, 2 y 3 resumen el flujo de trabajo más completo de la aplicación. La primera información cuando un usuario se conecta es un texto explicativo, donde se le pide que seleccione el *benchmark* de su interés de una lista desplegable (Figura 1), obtenida de forma dinámica. Una vez seleccionado el *benchmark*, la aplicación consulta al servidor los valores de dimensión para que el usuario elija de forma similar (si hubiese más de uno; si no, como en la Figura 1, se selecciona automáticamente y se va al paso siguiente).

The process is simple:

1. Select the wanted benchmark.
2. Select the dimension (if there is more of one for the benchmark).
3. Select the algorithm to compare. You can select from the Database and/or add results from an Excel file.
4. Select the report.
5. Push the button **Compare**.

#### Select Benchmark

Select the benchmark used for comparison:

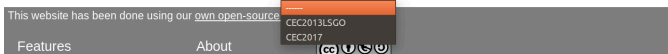


Figura 1: Selección del *benchmark* para la comparación. La lista desplegable es dinámica.

Luego, se listan los algoritmos almacenados en la base de datos asociados a ese *benchmark* (públicos o de ese usuario, si está identificado), y para esa dimensión (ver Figura 2) y los muestra con un *checkbox* para seleccionarlos. El usuario puede seleccionar de la lista los que desea, y, opcionalmente, subir un fichero con la información de sus propios algoritmos (TACO admite ficheros tanto en formato *csv* como *xls*) y un nombre. Como cada *benchmark* podría tener un formato diferente, encima del formulario para subir el fichero se muestra un enlace con un fichero Excel de ejemplo para el *benchmark* seleccionado y se puede consultar una plantilla para rellenar. Aunque el usuario puede indicar el nombre del algoritmo en el formulario, también puede indicarlo en el fichero de entrada en su propia columna (o en varias hojas), lo cual permite enviar en un único fichero los resultados de

distintos algoritmos, que serán comparados entre sí y con el resto de algoritmos seleccionados. En este punto, el usuario escogerá el tipo de informe de entre los disponibles que, como ya se indicó, dependerá del tipo de *benchmark* elegido.

#### CEC'2013 Large Scale Global Optimization

Benchmark for the Large Scale Global Optimization competitions.

Dimension value: 1000

#### Algorithms to compare

- BICCA
- CC-CMA-ES
- DECC-G
- DEEPSO
- MLSHADE-SPA
- MOS
- NEW RESTART
- SACC
- SHADEILS
- VMODE

Figura 2: Selección de los algoritmos. El usuario puede elegir de la lista y/o subir un fichero con los suyos.

#### Algorithms to compare

- BICCA
- CC-CMA-ES
- DECC-G
- DEEPSO
- MLSHADE-SPA
- MOS
- NEW RESTART
- SACC
- SHADEILS
- VMODE

If you want to compare existing results with your own proposal, set a Excel or CSV file with the results (opti the required format is indicated [here](#)

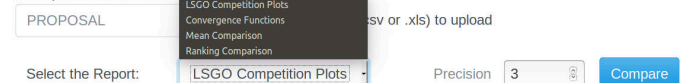


Figura 3: Selección del informe de la comparación. La lista de informes disponibles depende del *benchmark* elegido.

Por último, para lanzar el análisis basta con pulsar en el botón *Compare*. En las siguientes secciones detallamos los distintos tipos de informes soportados (hasta el momento), y ejemplos del tipo de análisis que la aplicación permite.

#### V. TIPOS DE INFORMES

El modelo de informes de TACO ha sido particularmente diseñado para ser flexible. Existen informes asociados a cada *benchmark* que pueden dividirse en las siguientes categorías:

- Informes de propósito general, accesibles para todos los *benchmarks*. Ejemplos de éstos serían tablas comparativas de estadísticas comunes (media, ranking), gráficas de convergencia, o algunas comparativas estadísticas.

- Específicos para un *benchmark*, ya que algunos pueden necesitar informes específicos (con medidas particulares). Un ejemplo de esto, ya incluido, es el informe usado en las competiciones de alta dimensionalidad (*LSGO competitions at IEEE CEC*), que asigna a cada algoritmo un número de puntos en función de su *ranking*, y se suman para valorar la bondad de cada algoritmo. Este informe es muy específico de esa comunidad de usuarios, y tiene poco sentido que esté disponible para otros *benchmarks*.

Esta primera versión de TACO incluye:

- Tres informes de tipo general: Comparación de medias, ranking medios, y gráficas de convergencia. Actualmente se están desarrollando comparativas estadísticas que esperamos estén disponibles en el momento del congreso.
- Un informe específico: El ya descrito usando el criterio de la F1 para el CEC'2013 de alta dimensionalidad.

A continuación los describimos muy brevemente, mostrando ejemplos reales usando las competiciones de optimización real del CEC'2017 [4], y la competición de alta dimensionalidad del CEC'2013 [5].

#### V-A. Informe comparativo de medias

Es el más sencillo. Computa la media de cada función y algoritmo, y muestra las medias para cada hito en una tabla, destacando la menor media, como se puede observar en la Figura 4. Además de remarcar la mejor media, incluye una línea resumen al final. Como toda tabla, los botones permiten exportar los datos en Excel y en Latex. Adicionalmente, este informe genera un resumen de los distintos valores, en una tabla final como la reflejada en la Figura 5.

#### V-B. Informe de ranking

A veces los valores medios interesan menos que la posición relativa de cada algoritmo. En este informe se ordena para cada función los algoritmos por su posición (1 al mejor, 2 al siguiente, y así sucesivamente), destacando el mejor, como se puede observar en la Figura 6. Además de remarcar la mejor media, incluye en la línea final el ranking medio por algoritmo, y una tabla final con los ranking medios para cada hito.

#### V-C. Informe de gráficas de convergencia

En este informe se muestra la evolución del error medio para cada función y algoritmo en cada hito. La curva puede ser más o menos suave, dependiendo del número de hitos asociados al *benchmark*. La Figura 7 muestra un ejemplo para el CEC'2013, donde sólo hay registrados tres hitos, y es más abrupta que en el caso de la Figura 8, correspondiente al CEC'2017, con más hitos y, por tanto, más suave. En la Figura 8, se muestra claramente que LSHDE\_SPACMA ofrece una mejora suave pero continua, mientras que otros algoritmos como RB-IPOP-CMA-ES convergen más rápidamente pero se estancan antes, confirmando lo indicado en la tabla de medias.

### Accuracy: 3.0e+06

Functions	CC-CMA-ES	DEEPSO	MOS	SHADEILS	VMODE
F01	5.770e-09	1.440e+08	0.000e+00	2.694e-24	8.510e-04
F02	1.330e+03	1.490e+04	8.320e+02	9.999e+02	5.510e+03
F03	1.510e-13	2.040e+01	9.170e-13	2.005e+01	3.410e-04
F04	2.190e+09	4.770e+09	1.740e+08	1.478e+08	8.480e+09
F05	7.280e+14	1.450e+07	6.940e+06	1.391e+06	7.280e+14
F06	5.870e+05	1.020e+06	1.480e+05	1.023e+06	1.990e+05
F07	7.440e+06	1.540e+07	1.620e+04	7.405e+01	3.440e+06
F08	3.880e+14	5.420e+12	8.000e+12	3.173e+11	3.260e+13
F09	3.710e+08	9.170e+08	3.830e+08	1.636e+08	7.510e+08
F10	7.550e+05	9.070e+07	9.020e+05	9.181e+07	9.910e+06
F11	1.580e+08	5.600e+08	5.220e+07	5.105e+05	1.580e+08
F12	1.270e+03	1.540e+10	2.470e+02	6.182e+01	2.340e+03
F13	6.690e+08	8.750e+08	3.400e+06	1.003e+05	2.430e+07
F14	7.100e+07	4.330e+08	2.560e+07	5.760e+06	9.350e+07
F15	3.030e+07	7.040e+06	2.350e+06	6.254e+05	1.110e+07
Best	2	0	3	11	0

Excel Latex

Figura 4: Ejemplo de informe comparativo de medias para CEC'2013.

### Bests by algorithm

	120000	600000	3000000
CC-CMA-ES	2	2	2
DEEPSO	0	0	0
MOS	6	3	3
SHADEILS	6	10	11
VMODE	1	0	0

Excel Latex

Figura 5: Ejemplo de informe comparativo final de medias para CEC'2013.

#### V-D. Informe F1 del benchmark CEC'2013

Este informe es específico para el *benchmark* CEC'2013 de alta dimensionalidad, y es un buen ejemplo de la flexibilidad del *framework*. En esta competición, se ordena a los participantes según el sistema de puntos de la F1, en el que cada corredor recibe un número de puntos según su



Accuracy: 3.0e+06

Functions	CC-CMA-ES	DEEPSO	MOS	SHADEILS	VMODE
F01	3	5	1.5	1.5	4
F02	3	5	1	2	4
F03	1	5	2	4	3
F04	3	4	2	1	5
F05	4.5	3	2	1	4.5
F06	3	4	1	5	2
F07	4	5	2	1	3
F08	5	2	3	1	4
F09	2	5	3	1	4
F10	1	4	2	5	3
F11	3.5	5	2	1	3.5
F12	3	5	2	1	4
F13	4	5	2	1	3
F14	3	5	2	1	4
F15	5	3	2	1	4
Mean	3.200	4.333	1.967	1.833	3.667

Excel    Latex

Figura 6: Ejemplo de informe comparativo de rankings para CEC'2013.



Figura 7: Ejemplo de gráfica de convergencia para la función 4 del benchmark CEC'2013

posición final: 25 puntos para el ganador, 18 para el segundo, etc. Análogamente, el informe asigna a cada algoritmo los puntos correspondientes a su posición para cada función. Los resultados son luego sumados para cada grupo de funciones (unimodales vs. multimodales, separables vs. no-separables, etc.) o globalmente. La Figura 9 muestra los datos desglosados de acuerdo a la separabilidad de las funciones, con una

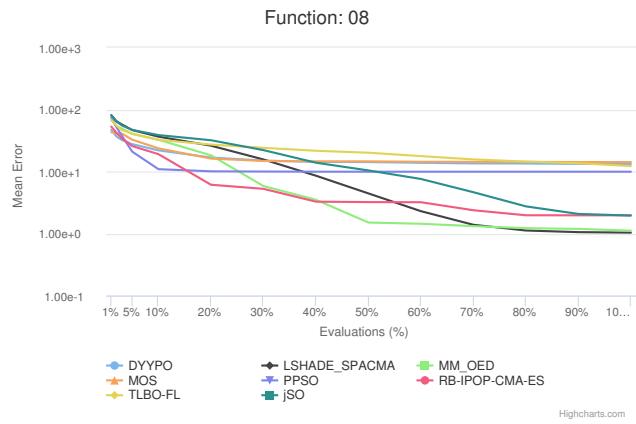


Figura 8: Ejemplo de gráfica de convergencia para la función 8 del benchmark CEC'2017

gráfica por cada hito del *benchmark*. A su vez, la Figura 10 presenta una comparativa global (sumando los puntos de cada hito). El color de cada componente de la figura representa el subgrupo de funciones, de acuerdo a las características antes mencionadas y, al pasar el ratón por cada componente, se indica el valor exacto de puntos y el nombre de esa categoría.

De ambas figuras destaca que MOS [8] obtiene los mejores resultados a partir de un número de evaluaciones, y prácticamente en todas las categorías.

V-E. Inclusión de nuevos informes

Para permitir una mayor flexibilidad, cada informe es una clase, que define un API de funciones que deben implementarse. Estas funciones devolverán, por medio de una librería, las tablas y figuras que se incorporarán al análisis. Por cada elemento (ya sea una tabla o una figura) el informe necesita devolver una estructura con la siguiente información: el objeto a devolver; el tipo del objeto (tabla o figura); y el orden del elemento, para permitir combinar tanto tablas como figuras de forma alternada dentro de un mismo informe.

Para generar las tablas, la aplicación utiliza *dataframes* de la librería *pandas*, considerada el estándar para cálculo científico. Para crear figuras, se ofrecen unas funciones propias que sirven como capa de abstracción para generar fácilmente distintos tipos de gráficas, tales como diagramas de barras, de líneas, etc. El motivo de esta abstracción es simplificar el proceso, y permitir a los informes ser independientes de las librerías de visualización que hay por debajo, para poder cambiarlas, llegado el caso, sin modificar los informes existentes.

Actualmente, se da soporte tanto a la librería *Holoviews*<sup>6</sup>, (librería de software libre para web, que presenta limitaciones de funcionalidad y es lenta), como a la librería *Highcharts*<sup>7</sup> (más rápida y con mayor funcionalidad, pero que es comercial, aunque permite el uso gratuito si se usa sin fines comerciales).

<sup>6</sup><http://holoviews.org>

<sup>7</sup><https://www.highcharts.com>

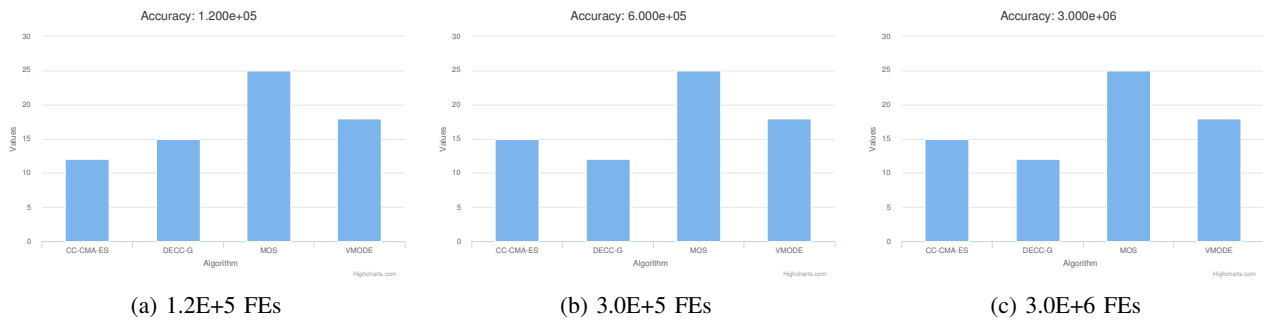


Figura 9: Resultados para el criterio de la F1 para funciones no separables.

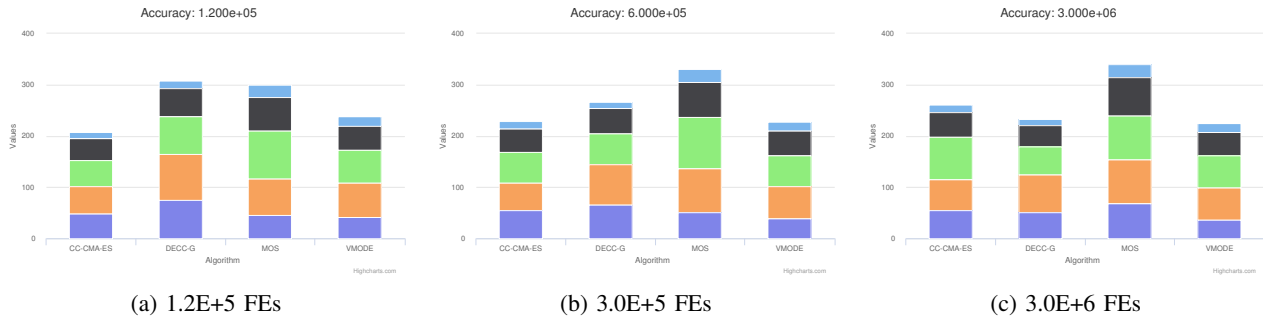


Figura 10: Resultados para el criterio de la F1 para todas las funciones.

En un futuro se plantea incorporar la librería *Altair*<sup>8</sup>, muy prometedora y que está actualmente en desarrollo.

## VI. CONCLUSIONES

En este trabajo hemos presentado TACO, Toolkit for the Automatic Comparison of Optimizers, una herramienta online avanzada para la comparación de algoritmos. Como se ha discutido, actualmente es capaz de manejar múltiples *benchmarks* y su información asociada (número de funciones, distinto número de dimensiones, hitos requeridos, grupos de funciones, etc). Además, para cada *benchmark* se pueden realizar, de forma automática, distintos tipos de análisis. La aplicación fue concebida como soporte para competiciones, pero es de utilidad para todos los investigadores que necesiten comparar sus propuestas, ya que los resultados son exportables para incluirlos en sus artículos y no requiere registro previo.

Aunque está suficientemente maduro y listo para ser usado en producción, existen planes para seguir extendiendo y mejorando TACO, que incluyen el aumento de la base de datos con más *benchmarks* y algoritmos, la inclusión de más información sobre los algoritmos, nuevos tipos de informes, especialmente con *tests estadísticos*, la exportación global de todas las tablas y figuras (actualmente, se deben exportar una a una), etc.

Como puede verse, TACO es una herramienta potente y flexible para comparar distintos algoritmos. Es más, aunque está enfocado a las metaheurísticas, puede usarse para comparar algoritmos de diferentes dominios (Machine Learning, etc.).

<sup>8</sup><https://altair-viz.github.io>

## AGRADECIMIENTOS

Este trabajo ha sido financiado, en parte, por ayudas del Ministerio de Ciencia, Innovación y Universidades (TIN2014-57481-C2-2-R, TIN2016-8113-R, TIN2017-83132-C2-2-R y TIN2017-89517-P) y de la Junta de Andalucía (P12-TIC-2958).

## REFERENCIAS

- [1] F. Glover and G. A. Kochenberger, *Handbook of metaheuristics*. Springer, 2003.
- [2] A. LaTorre, S. Muelas, and J. M. Peña, “A Comprehensive Comparison of Large Scale Global Optimizers,” *Information Sciences*, vol. 316, pp. 517–549, 2015.
- [3] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez, “Metaheuristic optimization frameworks: a survey and benchmarking,” *Soft Computing*, vol. 16, no. 3, pp. 527–561, 2012.
- [4] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization,” Tech. Rep., Oct. 2016.
- [5] J. Liang, B.-Y. Qu, P. Suganthan, and A. Hernández-Díaz, “Problem definitions and evaluation criteria for the cec 2013 special session and competition on real-parameter optimization,” Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, Tech. Rep., 2013.
- [6] R. Biedrzycki, “A version of ipop-cma-es algorithm with midpoint for cec 2017 single objective bound constrained problems,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1489–1494.
- [7] N. Awad, M. Ali, and P. Suganthan, “Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 372–379.
- [8] A. LaTorre and J. M. Peña, “A comparison of three large-scale global optimizers on the cec 2017 single objective real parameter numerical optimization benchmark,” in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1063–1070.