Toolkit for the Automatic Comparison of Optimizers: comparing large-scale global optimizers made easy

1st Daniel Molina
Department of Computer Science
University of Granada
Granada, Spain
dmolina@decsai.ugr.es

2nd Antonio LaTorre
DATSI, ETSIINF
Center for Computational Simulation
Universidad Politécnica de Madrid
Madrid, Spain
atorre@fi.upm.es

***Abstract*—Large-scale global optimization is a research subject that has attracted significant attention, including both theoretical and practical studies, in recent years. In this sense, some of the main conferences in the field of Evolutionary Computation have been organizing special sessions on this topic for more than a decade. Those special sessions normally propose a well defined benchmark of functions to allow a fair comparison of participating algorithms. Being able to manually tackle all this information has become a difficult task for many researchers. Which algorithm obtained the best results on a particular benchmark? How does the new method that I am developing compare to that algorithm? To answer this question, we propose Toolkit for the Automatic Comparison of Optimizers, TACO, a web application that stores all this information and makes it possible to seamlessly analyze it and generate detailed reports with the results of these analyses. The application has been designed in such a flexible way that it is extremely easy to add new benchmarks and their associated (possibly specific) analyses. Of course, these benchmarks are not limited to large-scale global optimization, but potentially any type of optimization problems. Finally, we also provide a publicly accessible instance demonstrating the features of the application with ready-to-use results from some recent special sessions.**

***Index Terms*—Large-scale global optimization, Benchmarking, Web Application, Comparisons.**

I. Introduction

The comparison of a new large-scale global optimizer on a particular benchmark with existing results in the literature can be a time-consuming task. Not only the researcher has to gather her own results but she has also to collect those of the algorithms to be included in the comparison. This implies a continuous updating process that can be cumbersome. Furthermore, data gathering is only the first task in the comparison process: data must be normalized (in the sense of obtaining the same statistics and/or milestones), grouped (by some of the characteristics of the functions), arranged (in tables or similar structures) and analyzed (by one or several of the existing statistical or graphical techniques). Then, if the researcher wants to analyze the data from a different perspective, she will possibly have to rearrange them, which increases the amount of time that must be devoted to these tasks.

To alleviate this problem, we propose the *Toolkit for the Automatic Comparison of Optimizers*, TACO, freely available at **https://tflsgo.herokuapp.com/**, a web application that combines a publicly accessible repository of results with a modular analysis tool that allows a seamless comparison of the results of large-scale global optimizers on different benchmarks, both during the development of the algorithm (as a support tool to guide the process) and when reporting the results in a publication. It features a number of interesting characteristics such as:

- Support for multiple benchmarks, each of them with its own properties: associated functions and groups of functions, possible different problem sizes (dimensions), required milestones, etc.
- Modular design: the analyses available in the application are structured on top of reports. These reports can be general or benchmark-specific and each benchmark can decide which of them to use.
- Storage of full results of the experimentation: the database does not only store common statistics (minimum, maximum, mean, etc.) that can be collected from publications when performing a comparative study but instead it allows to upload the full results (error values for all the milestones and runs) of an experimentation. This allows some analyses (non-parametric statistical analyses, for example) that can not be carried out with only the summary information.
- Flexible definition of reports: each report decides which information to show and the most suitable format for it. The application supports both tables and plots and, in the latter case, it provides an abstraction layer for common plots (lines, bars, stacked bars, etc.) to ease the addition of new reports. These plots can be saved locally to use them in papers.

This is not the first attempt to create a tool like this. TACO is inspired by a previous one [1] but trying to overcome the limitations of this first approach:

- To conduct an analysis in TACO the user does not need to

previously store her results in the database nor get them validated by the administrators. Any user can run her analyses by just providing the results in a file with any of the supported formats (currently, csv and xls). Users can choose to run an analysis on data present in the database, on data provided in the aforementioned file or combining information of both sources.
- Apart from the previous use case (direct use of the application, no registration needed) TACO also supports multiple authenticated users that can manage their own results in independent repositories. Each user has a private area to store data (algorithms results on some benchmark). These data, along with the main data in the database and the results provided for direct use as illustrated in the previous use case, can be included in their analyses but are not visible to other users. Only when these results are considered to be definitive, the user can request the web application administrators to include them in the public database.
- The application has been completely revamped, programming it as a standalone application supported by standard web technologies instead of as a Wordpress plugin (with all the limitations and security problems that it implied).

Apart from the tool presented in [1], there have been some additional attempts to incorporate similar analyses in other tools, as reviewed in [2]. However, in these cases the analysis process is coupled with the optimization framework itself, what makes them unsuitable for their use as a general solution for a broad set of researchers. Some examples of optimization frameworks with reporting and analysis capabilities (mainly statistical analyses and basic charting) can be found in [3], [4], [5], [6].

The target audience of this new tool is not only researchers but also special session organizers. Due to the modular design of the application (that we will discuss later in this paper) it is straightforward to include a new special session in it that will allow the participants of such special session to upload their results and compare them with reference results provided by the organizers. Once the special session has taken place, the results of all the algorithms associated to that special session can be made publicly accessible.

The remainder of this paper is organized as follows. Section II briefly reviews the architecture of the application. In Section III we review the available options of the frontend of the application and show some relevant screenshots. Section IV analyzes the modular design of the reports system, whereas Section V presents a usage example of the application. Finally, this study concludes with some final remarks in Section VI.

II. Architecture

Figure 1 presents the client-server model adopted in TACO. As can be seen, it relies on standard web technologies:

- HTML5, CSS3 and Javascript (via the Vue.js[1] framework) in the frontend.
- The Python frameworks Flask[2] and SQL Alchemy[3] in the backend.
- Messages are exchanged between client and server via AJAX calls and JSON messages.

This design allows the application to be used as a standalone application or to embed it in another application, including static web servers: the dynamic content is generated in the backend, gathered by AJAX calls and visualized into the page using Javascript on the client.

One typical work session with TACO starts by selecting the benchmark for which the user wants to conduct the comparison. Then, depending on the benchmark, the user might need to choose among the available dimensionalities (if more than one) and, finally, she will be presented with a list of the algorithms for which there are results for that benchmark and problem size in the repository. Then, the user can select some of these algorithms and, optionally, upload her own results to conduct the comparison. Once all the data involved in the comparison is ready, the user can select the type of comparison among the available ones (this will be covered in detail in Section IV) and proceed to the analysis. The Vue.js framework is used extensively in this part of the application to dynamically retrieve all the aforementioned information (available benchmarks, dimensionalities, algorithms, reports, etc.) from the server. Furthermore, as much information as possible is cached locally to increase the performance of the application.

From the point of view of the data model, the concept of benchmark is at the core of the design. Each benchmark is associated with a different table to store its results. This permits different benchmarks to be made up of a different number of functions as the structure of each of these tables differs. Benchmarks also store (either directly in their table or in other associated tables) relevant information for the analysis of the results:

- Dimensions: some benchmarks are made up of functions with a fixed number of dimensions, whereas others are scalable and several problem sizes are typically studied to analyze the scalability of the algorithms. The data model used in our application allows a benchmark to be associated with more than one problem size and results can be studied for any of them, independently.
- Milestones: a benchmark can have a number of mandatory milestones (in large-scale global optimization problems these are typically 1.2E+05, 3.0E+05 and 3.0E+06) as well as other optional milestones. This information is used when parsing the results uploaded by the users of the applications to appropriately store their information.
- Groups or categories of functions: benchmarks are normally made up of functions with different characteristics and it is useful to store this information to allow the analysis of the results per groups of functions (e.g., compare

[1]https://vuejs.org/

[2]http://flask.pocoo.org/

[3]https://www.sqlalchemy.org/

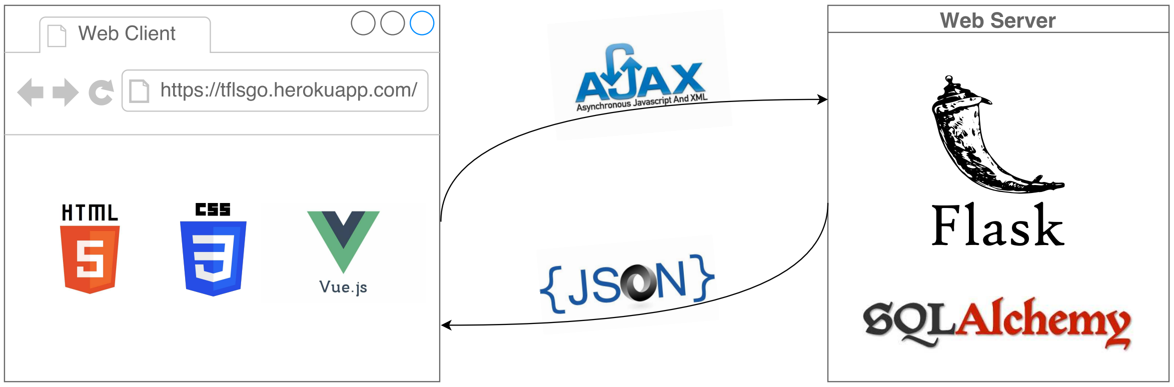

Web Client
https://tflsgo.herokuapp.com/
HTML
CSS
Vue.js
AJAX
Asynchronous Javascript And XML
{JSON}
Web Server
Flask
SQLAlchemy

Fig. 1: Web application architecture and technologies involved

the performance of several algorithms on unimodal and multimodal functions, independently).

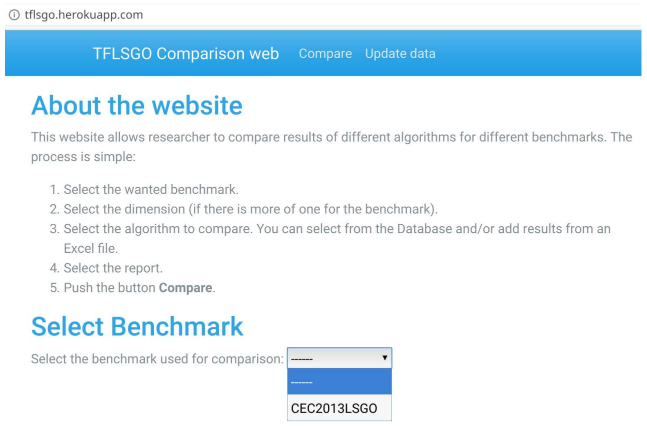
- Reports: another important module of the application is the reporting system (see Section IV for details). One benchmark can be associated to any of the standard pre-defined reports (non-parametric statistical analysis, convergence plots, etc.) or with a specifically designed one for that benchmark (e.g., the F1 ranking that has been used in some CEC special sessions). The reports presented to the user once a benchmark has been selected are those associated to it, regardless of other reports that could be possibly defined in the system.

In Section III we provide an overview of the frontend of the application in which all these characteristics can be observed.

III. Frontend

Figures 2, 3 and 4 summarize the standard workflow of the application. The first information that a user finds when she connects to the application's website is an informative message describing how she should interact with the application. Then, the user needs to select the benchmark for which she wants to conduct the analysis by selecting it in the drop-down list (Figure 2). Once the benchmark has been selected, the application queries the server for the list of available dimensions for the benchmark selected and shows the user a similar drop-down list. However, in the event of the selection of a benchmark with a fixed dimensions number (as it is the case for the CEC2013 benchmark shown in Figure 2), the application will automatically select that value and continue to the next step.

The application will now present the list of algorithms available in the database for the selected benchmark and problem size (Figure 3). The user can select some of the algorithms from this list and (optionally) upload a file with the information of her own algorithms. The application will check that at least one algorithm is selected from the list when no results file is provided or that this file is uploaded when no algorithm is selected. TACO supports results both in *csv* and *xls* formats. To help users to collect their results in the appropriate format, a sample file with the correct format is dynamically generated (it depends on the benchmark and the



About the website

This website allows researcher to compare results of different algorithms for different benchmarks. 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: ------

CEC2013LSGO

Fig. 2: Selection of the benchmark for the comparison. The drop-down list is dynamically generated by querying the database and gathering the available benchmarks.

number of functions and dimensions that it has associated) and can be downloaded from the application (the link at the top of the results upload form).

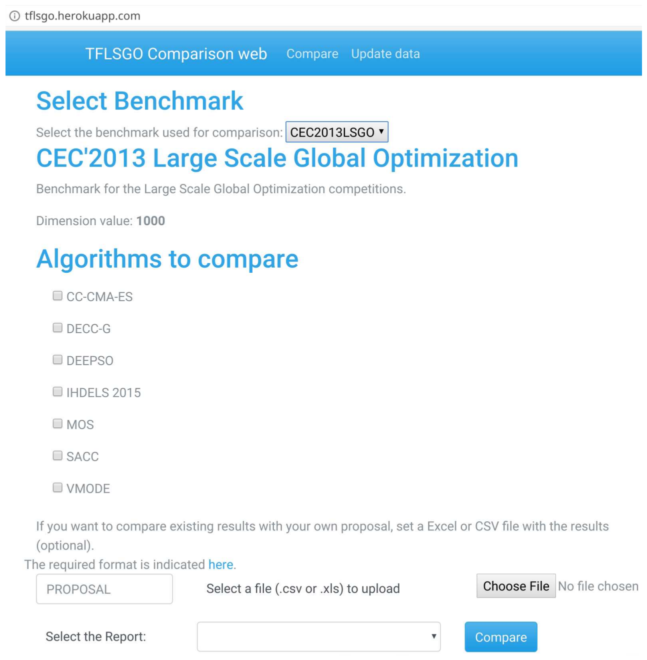
Finally, the user needs to select the type of report that she wants to obtain. As described in Section II, depending on the benchmark selected, the list of available reports can vary.

At this point, everything is ready to launch an analysis by clicking on the *Compare* button. An example of the kind of analyses that the application currently supports is given in Section V. However, before going into that, a detailed description of the reporting system is provided in Section IV.

IV. Reports

As already mentioned, one of the core characteristics of TACO is its modularity and flexibility. One of the best examples of this modular design is the reporting system. As mentioned before, each benchmark included in the database has a variable number of associated reports. These reports can belong to two different categories:

- General purpose reports. Some analyses are independent of the particular benchmark under consideration.


tflsgo.herokuapp.com

TFLSGO Comparison web Compare Update data

Select Benchmark

Select the benchmark used for comparison: CEC2013LSGO

CEC'2013 Large Scale Global Optimization

Benchmark for the Large Scale Global Optimization competitions.

Dimension value: **1000**

Algorithms to compare

- CC-CMA-ES
- DECC-G
- DEEPSO
- IHDELS 2015
- MOS
- SACC
- VMODE

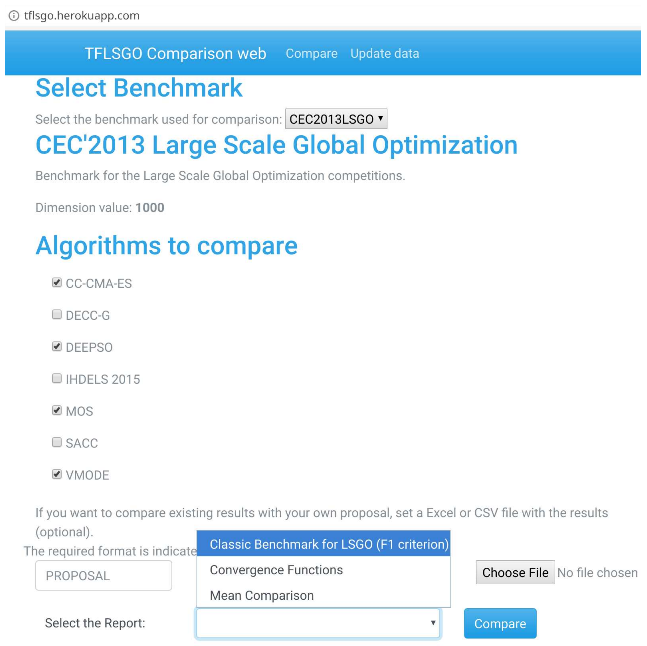
If you want to compare existing results with your own proposal, set a Excel or CSV file with the results (optional).

The required format is indicated here.

PROPOSAL Select a file (.csv or .xls) to upload Choose File No file chosen

Select the Report: Compare

Fig. 3: Selection of the algorithms for the comparison. The user can select one or more algorithms from the list and/or upload a file with her own results.


tflsgo.herokuapp.com

TFLSGO Comparison web Compare Update data

Select Benchmark

Select the benchmark used for comparison: CEC2013LSGO

CEC'2013 Large Scale Global Optimization

Benchmark for the Large Scale Global Optimization competitions.

Dimension value: **1000**

Algorithms to compare

- [✓ CC-CMA-ES]
- DECC-G
- [✓ DEEPSO]
- IHDELS 2015
- [✓ MOS]
- SACC
- [✓ VMODE]

If you want to compare existing results with your own proposal, set a Excel or CSV file with the results (optional).

The required format is indicate

PROPOSAL Choose File No file chosen

Classic Benchmark for LSGO (F1 criterion)
Convergence Functions
Mean Comparison

Select the Report: Compare

Fig. 4: Selection of the reports for the comparison. Depending on the benchmark, the available reports can differ.

Examples of these reports are summary tables with common statistics (minimum, maximum, mean, median, standard deviation), convergence plots (fitness vs. fitness evaluations) or non-parametric statistical tests. All these reports are generic and can be applied to any selection of algorithms.

- Benchmark/Special Session specific reports. On the other hand, some benchmarks may require of special reports that conduct analyses too specific for the problems of that particular set of problems that would not be applicable to problems of a different domain. For example, if the benchmark contains some real-world problems (from the scientific or the engineering domain) specific analyses paying attention to features different from the fitness value itself (physical characteristics of the models adjusted, robustness of the solutions with regards to any additional considerations, factibility of the solutions reported, etc.) might be necessary. Another scenario in which these specific reports are frequent is in the context of a special session. A good example of this (which is actually included in the current version of the application) is the F1 criterion followed by CEC large-scale global optimization sessions. This report sorts algorithms according to the error reported for each function and assigns points to each algorithm according to the F1 criterion (25 points to the first algorithm, 18 to the second, etc.). This report is too specific of a particular community of users and thus it makes little sense to make it available to other benchmarks.

For this first version of TACO we have included three different reports:

- Two general-purpose report: means comparison and convergence plots.
- A benchmark-specific one: CEC2013 benchmark report using the F1 criterion.

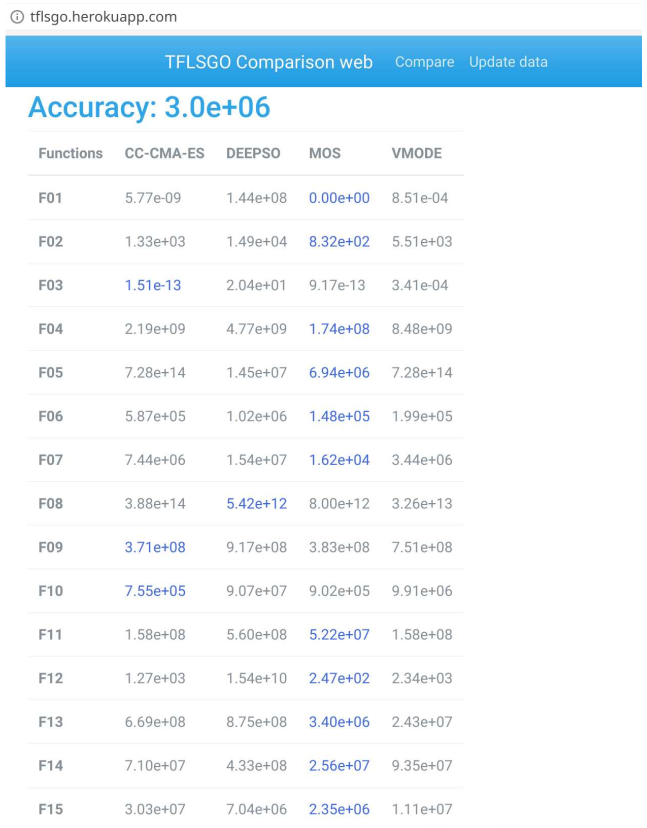
In the remainder of this section we discuss each of these reports and briefly discuss how the application can be expanded by adding new reports (and the support provided by the framework that facilitates this procedure).

A. *Means comparison report*

The first report is the simplest one of the three reports provided. It basically computes average errors for each function and algorithm at each milestone and arranges this information in a table. The smallest error values for each function are highlighted. Figure 5 depicts an example of this table for the CEC2013 benchmark and the last milestone (maximum number of fitness evaluations).

B. *Convergence plots report*

This second report is also generic and can be associated to any benchmark. It plots the evolution of the mean error for each function and algorithm reported at the required milestones. Depending on the benchmark, the number of milestones can differ (which can affect to the detail of the plot: more milestones mean softer convergence plots), but all of them are subject to this kind of analysis. An example of this kind of plots can be seen in Figure 6.


tflsgo.herokuapp.com

TFLSGO Comparison web Compare Update data

Accuracy: 3.0e+06

Functions	CC-CMA-ES	DEEPSO	MOS	VMODE
F01	5.77e-09	1.44e+08	0.00e+00	8.51e-04
F02	1.33e+03	1.49e+04	8.32e+02	5.51e+03
F03	1.51e-13	2.04e+01	9.17e-13	3.41e-04
F04	2.19e+09	4.77e+09	1.74e+08	8.48e+09
F05	7.28e+14	1.45e+07	6.94e+06	7.28e+14
F06	5.87e+05	1.02e+06	1.48e+05	1.99e+05
F07	7.44e+06	1.54e+07	1.62e+04	3.44e+06
F08	3.88e+14	5.42e+12	8.00e+12	3.26e+13
F09	3.71e+08	9.17e+08	3.83e+08	7.51e+08
F10	7.55e+05	9.07e+07	9.02e+05	9.91e+06
F11	1.58e+08	5.60e+08	5.22e+07	1.58e+08
F12	1.27e+03	1.54e+10	2.47e+02	2.34e+03
F13	6.69e+08	8.75e+08	3.40e+06	2.43e+07
F14	7.10e+07	4.33e+08	2.56e+07	9.35e+07
F15	3.03e+07	7.04e+06	2.35e+06	1.11e+07

Fig. 5: Means comparison report for the last milestone of the CEC2013 benchmark for four different algorithms.

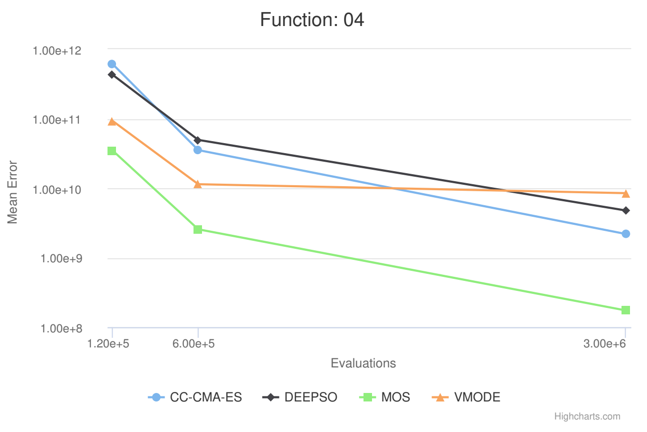


Fig. 6: Convergence plot for Function 4 of the CEC2013 benchmark for four different algorithms.

C. *CEC2013 benchmark F1 report*

This report is specific for the CEC2013 benchmark (associated to the corresponding special session). Traditionally, in the CEC special sessions on large-scale global optimization, the organizers have used a method to rank participating algorithms inspired by the Formula 1 points system, in which cars are assigned a different number of points depending on their final position: 25 points for the winner, 18 for the runner-up, etc. Analogously, in these competitions algorithms are ranked according to the error reported for each function and they are given the corresponding number of points following the aforementioned system. Results can then be aggregated taking into account the characteristics of the functions (unimodals vs. multimodals, separable vs. non-separable, etc.) or globally. Figure 7 depicts the case where the results have been aggregated for the non-separable group of functions. As can be seen, results are provided for the three mandatory milestones required by the benchmark. Similarly, Figure 8 represents the overall results for the same milestones. The color of each of the parts of the bars encodes the subgroup of functions from which those points come from. The plots include a tooltip to show the exact value and name of each category moving the mouse over each part.

D. *Inclusion of new reports*

The reporting system relies on a set of classes that facilitate the inclusion of new reports. In particular, every report extends a class that defines an API in which one function must be implemented. This function is in charge of generating the tables and figures that will be part of the analysis. For each element (either a table or a figure) the report needs to return a structure that contains the following information:

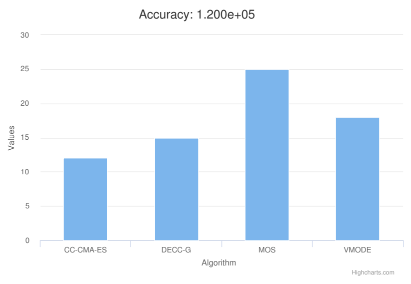
- The object itself.
- The type of the object (table or figure).
- The order in which the element should be inserted. This allows to have reports that consist of only tables, only figures or a combination of both, in any order.

For the generation of tables, the application makes use of standard *Pandas data frames*. On the other hand, for figures it provides an abstraction layer to easily generate frequently used plots: line plots, bar and stacked bars plots, etc. This way, new reports can be agnostic of the plotting library being used and changes in the backend do not imply to modify all the existing reports.

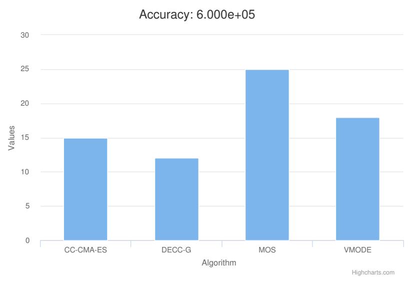
V. USAGE EXAMPLE

To illustrate the type of analyses that can be conducted with TACO, we have considered the results of the algorithms submitted to the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization [7], freely available at the website of the organizers of the special session. In particular, our analysis has focused on the following 8 algorithms:

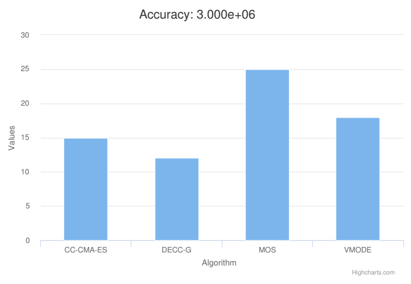
- DYYPO [8].
- LSHADE_SPACMA [9].
- MM_OED [10].
- MOS [11].
- PPSO [12].
- RB-IPOP-CMA-ES [13].
- TLBO-FL [14].
- jSO [15].



(a) Results at 1.2E+5 FEs

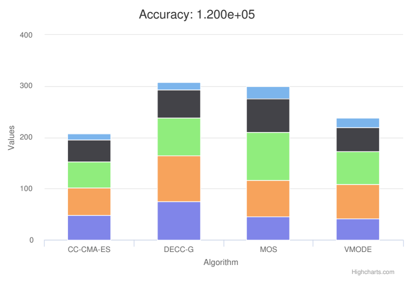


(b) Results at 3.0E+5 FEs

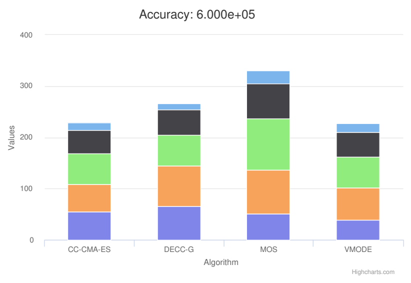


(c) Results at 3.0E+6 FEs

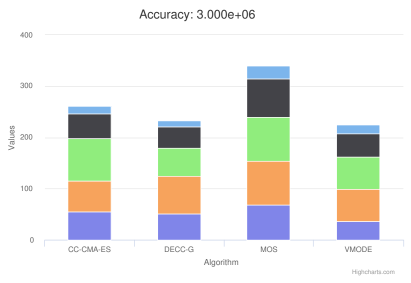
Fig. 7: Results for the F1 criterion at different milestones for the non-separable group of functions. The y axis represents the cumulated points for each of the algorithms.



(a) Results at 1.2E+5 FEs



(b) Results at 3.0E+5 FEs



(c) Results at 3.0E+6 FEs

Fig. 8: Results for the F1 criterion at different milestones for all the functions in the benchmark. The y axis represents the cumulated points for each of the algorithms.

For this analysis, we have only taken into account the Means comparison and the Convergence analysis reports as no specific report was proposed for this special session.

In the case of the Means comparison, this is analyzed for all the milestones defined in [7]: 1%, 2%, 3%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the available fitness evaluations. Figures 9-11 depict the output of this report for 100 dimensions and three different milestones: the first one (at 1% of the fitness evaluations), an intermediate one (30% of the fitness evaluations) and the last one (when all the fitness evaluations have been exhausted).

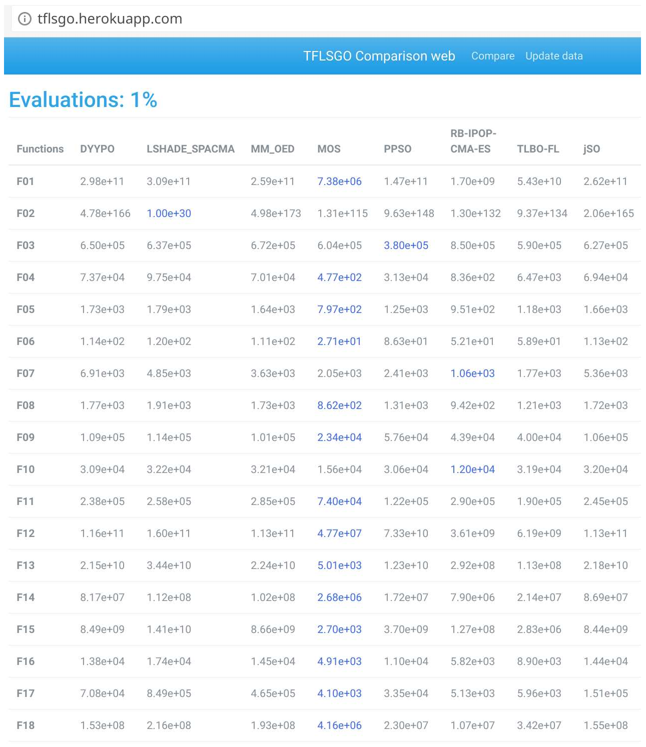
As can be seen on these plots, different algorithms perform differently at different stages of the search: MOS is the fastest algorithm and quickly decreases error values for most of the functions. On the other hand, LSHADE_SPACMA converges more slowly but to better regions of the search space, finding the best solutions for many of the considered problems.

If we look at the convergence plots instead in 10 dimensions, we can observe that several different patterns arise, depending on the function. For some problems, there is an algorithm that outperforms all the others for all their execution (Figure 12). For others, algorithms exhibit different performance at different stages of the search (Figure 13). There are also some functions for which there is not a clear difference and most of the algorithms quickly converge to similar solutions (Figure 14). Finally, there is an interesting pattern in which it takes a different amount of time for the algorithms to locate the region of the global optimum but, once located, all of them quickly converge towards it (Figure 15).

As can be seen, TACO is a powerful and flexible tool to compare different algorithms.

VI. Conclusions

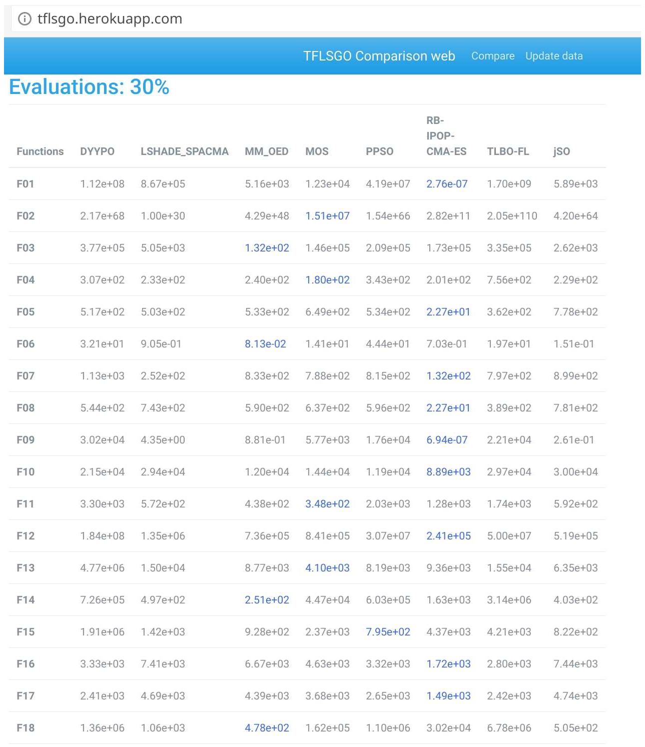
In this paper, we have presented TACO, a new tool for the comparison of large-scale global optimizers. As discussed in the paper, the application is currently able to handle multiple benchmarks and their associated information (number of functions, possible number of dimensions, required milestones, groups of functions, etc). Moreover, for each of these benchmarks, different types of analyses can be conducted. Some of them will be general and shared by multiple benchmarks, whereas others will be specific for that particular benchmark. The application is currently focused on large-scale global optimization benchmarks due to the background of the authors. Nonetheless, the modular design of TACO makes it straightforward to incorporate new benchmarks: it only requires of the registration of the benchmark with all its information and the desired specific reports, if any. Despite being mature enough and ready to use in a production environment, there are a number of ways in which TACO can be extended and improved. First, as mentioned before, the database will be enriched by incorporating the information of more benchmarks and algorithms. This process will not be limited to large-scale benchmarks but will also cover other types of problems and different domains. Second, more reports will be added, both general (e.g., non-parametric statistical tests) and benchmark-



Evaluations: 1%

Functions	DYYPO	LSHADE_SPACMA	MM_OED	MOS	PPSO	RB-IPOP-CMA-ES	TLBO-FL	jSO
F01	2.98e+11	3.09e+11	2.59e+11	7.38e+06	1.47e+11	1.70e+09	5.43e+10	2.62e+11
F02	4.78e+166	1.00e+30	4.98e+173	1.31e+115	9.63e+148	1.30e+132	9.37e+134	2.06e+165
F03	6.50e+05	6.37e+05	6.72e+05	6.04e+05	3.80e+05	8.50e+05	5.90e+05	6.27e+05
F04	7.37e+04	9.75e+04	7.01e+04	4.77e+02	3.13e+04	8.36e+02	6.47e+03	6.94e+04
F05	1.73e+03	1.79e+03	1.64e+03	7.97e+02	1.25e+03	9.51e+02	1.18e+03	1.66e+03
F06	1.14e+02	1.20e+02	1.11e+02	2.71e+01	8.63e+01	5.21e+01	5.89e+01	1.13e+02
F07	6.91e+03	4.85e+03	3.63e+03	2.05e+03	2.41e+03	1.06e+03	1.77e+03	5.36e+03
F08	1.77e+03	1.91e+03	1.73e+03	8.62e+02	1.31e+03	9.42e+02	1.21e+03	1.72e+03
F09	1.09e+05	1.14e+05	1.01e+05	2.34e+04	5.76e+04	4.39e+04	4.00e+04	1.06e+05
F10	3.09e+04	3.22e+04	3.21e+04	1.56e+04	3.06e+04	1.20e+04	3.19e+04	3.20e+04
F11	2.38e+05	2.58e+05	2.85e+05	7.40e+04	1.22e+05	2.90e+05	1.90e+05	2.45e+05
F12	1.16e+11	1.60e+11	1.13e+11	4.77e+07	7.33e+10	3.61e+09	6.19e+09	1.13e+11
F13	2.15e+10	3.44e+10	2.24e+10	5.01e+03	1.23e+10	2.92e+08	1.13e+08	2.18e+10
F14	8.17e+07	1.12e+08	1.02e+08	2.68e+06	1.72e+07	7.90e+06	2.14e+07	8.69e+07
F15	8.49e+09	1.41e+10	8.66e+09	2.70e+03	3.70e+09	1.27e+08	2.83e+06	8.44e+09
F16	1.38e+04	1.74e+04	1.45e+04	4.91e+03	1.10e+04	5.82e+03	8.90e+03	1.44e+04
F17	7.08e+04	8.49e+05	4.65e+05	4.10e+03	3.35e+04	5.13e+03	5.96e+03	1.51e+05
F18	1.53e+08	2.16e+08	1.93e+08	4.16e+06	2.30e+07	1.07e+07	3.42e+07	1.55e+08

Fig. 9: Means comparison report for the first milestone (1% of fitness evaluations) of the CEC2017 benchmark for eight different algorithms and the first 18 functions. The MOS algorithm is the best overall algorithm at this milestone.



Evaluations: 30%

Functions	DYYPO	LSHADE_SPACMA	MM_OED	MOS	PPSO	RB-IPOP-CMA-ES	TLBO-FL	jSO
F01	1.12e+08	8.67e+05	5.16e+03	1.23e+04	4.19e+07	2.76e-07	1.70e+09	5.89e+03
F02	2.17e+68	1.00e+30	4.29e+48	1.51e+07	1.54e+66	2.82e+11	2.05e+110	4.20e+64
F03	3.77e+05	5.05e+03	1.32e+02	1.46e+05	2.09e+05	1.73e+05	3.35e+05	2.62e+03
F04	3.07e+02	2.33e+02	2.40e+02	1.80e+02	3.43e+02	2.01e+02	7.56e+02	2.29e+02
F05	5.17e+02	5.03e+02	5.33e+02	6.49e+02	5.34e+02	2.27e+01	3.62e+02	7.78e+02
F06	3.21e+01	9.05e-01	8.13e-02	1.41e+01	4.44e+01	7.03e-01	1.97e+01	1.51e-01
F07	1.13e+03	2.52e+02	8.33e+02	7.88e+02	8.15e+02	1.32e+02	7.97e+02	8.99e+02
F08	5.44e+02	7.43e+02	5.90e+02	6.37e+02	5.96e+02	2.27e+01	3.89e+02	7.81e+02
F09	3.02e+04	4.35e+00	8.81e-01	5.77e+03	1.76e+04	6.94e-07	2.21e+04	2.61e-01
F10	2.15e+04	2.94e+04	1.20e+04	1.44e+04	1.19e+04	8.89e+03	2.97e+04	3.00e+04
F11	3.30e+03	5.72e+02	4.38e+02	3.48e+02	2.03e+03	1.28e+03	1.74e+03	5.92e+02
F12	1.84e+08	1.35e+06	7.36e+05	8.41e+05	3.07e+07	2.41e+05	5.00e+07	5.19e+05
F13	4.77e+06	1.50e+04	8.77e+03	4.10e+03	8.19e+03	9.36e+03	1.55e+04	6.35e+03
F14	7.26e+05	4.97e+02	2.51e+02	4.47e+04	6.03e+05	1.63e+03	3.14e+06	4.03e+02
F15	1.91e+06	1.42e+03	9.28e+02	2.37e+03	7.95e+02	4.37e+03	4.21e+03	8.22e+02
F16	3.33e+03	7.41e+03	6.67e+03	4.63e+03	3.32e+03	1.72e+03	2.80e+03	7.44e+03
F17	2.41e+03	4.69e+03	4.39e+03	3.68e+03	2.65e+03	1.49e+03	2.42e+03	4.74e+03
F18	1.36e+06	1.06e+03	4.78e+02	1.62e+05	1.10e+06	3.02e+04	6.78e+06	5.05e+02

Fig. 10: Means comparison report for the seventh milestone (30% of fitness evaluations) of the CEC2017 benchmark for eight different algorithms and the first 18 functions. Now the algorithm with best results is RB-IPOP-CMA-ES.

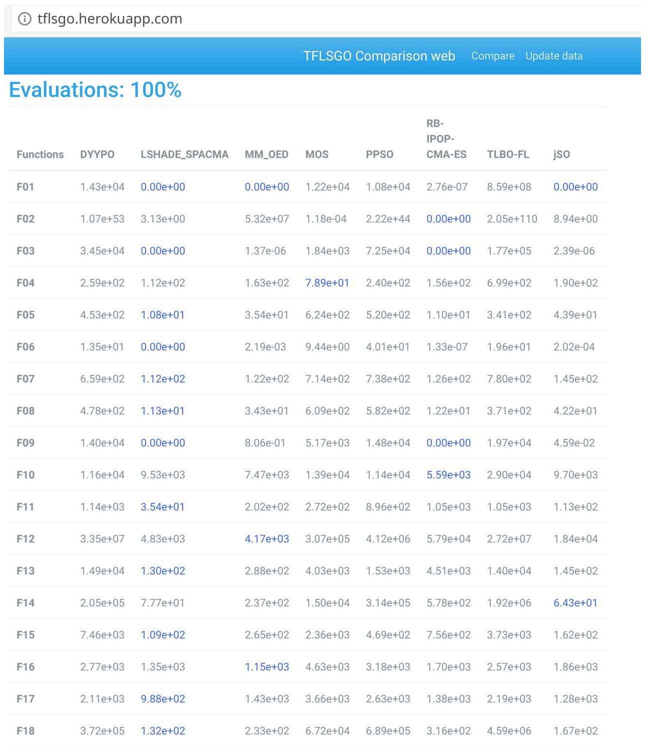
specific. Third, to facilitate the adoption of the application in the publication workflow of potential users, an option to download all the information of the analyses (tables and figures) will be added. Finally, we will add support for competitions to the application. Instead of having benchmarks as the sole element for grouping results of algorithms, we will allow users to define special sessions/competitions regardless of the benchmark used to group the results of their participants. This will hopefully help special sessions organizers in the process of comparing the results of the papers participating in those sessions.

ACKNOWLEDGMENTS

This work was supported by grants from the Spanish Ministry of Science (TIN2014-57481-C2-2-R, TIN2016-8113-R, TIN2017-83132-C2-2-R and TIN2017-89517-P) and Regional Government (P12-TIC-2958).

REFERENCES

[1] A. LaTorre, S. Muelas, and J. M. Peña, "A Comprehensive Comparison of Large Scale Global Optimizers," *Information Sciences*, vol. 316, pp. 517–549, 2015.
[2] 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.
[3] L. Di Gaspero and A. Schaerf, "EASYLOCAL++: an object-oriented framework for the flexible design of local-search algorithms," *Software: Practice and Experience*, vol. 33, no. 8, pp. 733–765, 2003.
[4] J. A. Parejo, J. Racero, F. Guerrero, T. Kwok, and K. A. Smith, "Fom: A framework for metaheuristic optimization," in *Computational Science — ICCS 2003*, P. M. A. Sloot, D. Abramson, A. V. Bogdanov, Y. E. Gorbachev, J. J. Dongarra, and A. Y. Zomaya, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 886–895.
[5] J. Brownlee, "Oat: The optimization algorithm toolkit," Swinburne University of Technology, Tech. Rep., 2007.
[6] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás, "Jclec: a java framework for evolutionary computation," *Soft Computing*, vol. 12, no. 4, pp. 381–392, 2008.
[7] 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.
[8] D. Maharana, R. Kommadath, and P. Kotecha, "Dynamic yin-yang pair optimization and its performance on single objective real parameter problems of cec 2017," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 2390–2396.
[9] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 145–152.
[10] K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-method based orthogonal experimental design algorithm for solving cec2017 competition problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1350–1357.
[11] A. LaTorre and J. M. Pea, "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.
[12] A. Tangherloni, L. Rundo, and M. S. Nobile, "Proactive particles in swarm optimization: A settings-free algorithm for real-parameter single objective optimization problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1940–1947.


tflsgo.herokuapp.com

TFLSGO Comparison web Compare Update data

Evaluations: 100%

Functions	DYYPO	LSHADE_SPACMA	MM_OED	MOS	PPSO	RB-IPOP-CMA-ES	TLBO-FL	jSO
F01	1.43e+04	0.00e+00	0.00e+00	1.22e+04	1.08e+04	2.76e-07	8.59e+08	0.00e+00
F02	1.07e+53	3.13e+00	5.32e+07	1.18e-04	2.22e+44	0.00e+00	2.05e+110	8.94e+00
F03	3.45e+04	0.00e+00	1.37e-06	1.84e+03	7.25e+04	0.00e+00	1.77e+05	2.39e-06
F04	2.59e+02	1.12e+02	1.63e+02	7.89e+01	2.40e+02	1.56e+02	6.99e+02	1.90e+02
F05	4.53e+02	1.08e+01	3.54e+01	6.24e+02	5.20e+02	1.10e+01	3.41e+02	4.39e+01
F06	1.35e+01	0.00e+00	2.19e-03	9.44e+00	4.01e+01	1.33e-07	1.96e+01	2.02e-04
F07	6.59e+02	1.12e+02	1.22e+02	7.14e+02	7.38e+02	1.26e+02	7.80e+02	1.45e+02
F08	4.78e+02	1.13e+01	3.43e+01	6.09e+02	5.82e+02	1.22e+01	3.71e+02	4.22e+01
F09	1.40e+04	0.00e+00	8.06e-01	5.17e+03	1.48e+04	0.00e+00	1.97e+04	4.59e-02
F10	1.16e+04	9.53e+03	7.47e+03	1.39e+04	1.14e+04	5.59e+03	2.90e+04	9.70e+03
F11	1.14e+03	3.54e+01	2.02e+02	2.72e+02	8.96e+02	1.05e+03	1.05e+03	1.13e+02
F12	3.35e+07	4.83e+03	4.17e+03	3.07e+05	4.12e+06	5.79e+04	2.72e+07	1.84e+04
F13	1.49e+04	1.30e+02	2.88e+02	4.03e+03	1.53e+03	4.51e+03	1.40e+04	1.45e+02
F14	2.05e+05	7.77e+01	2.37e+02	1.50e+04	3.14e+05	5.78e+02	1.92e+06	6.43e+01
F15	7.46e+03	1.09e+02	2.65e+02	2.36e+03	4.69e+02	7.56e+02	3.73e+03	1.62e+02
F16	2.77e+03	1.35e+03	1.15e+03	4.63e+03	3.18e+03	1.70e+03	2.57e+03	1.86e+03
F17	2.11e+03	9.88e+02	1.43e+03	3.66e+03	2.63e+03	1.38e+03	2.19e+03	1.28e+03
F18	3.72e+05	1.32e+02	2.33e+02	6.72e+04	6.89e+05	3.16e+02	4.59e+06	1.67e+02

Fig. 11: Means comparison report for the last milestone (100% of fitness evaluations) of the CEC2017 benchmark for eight different algorithms and the first 18 functions. In the end, the LSHADE_SPACMA algorithm obtained the best results.

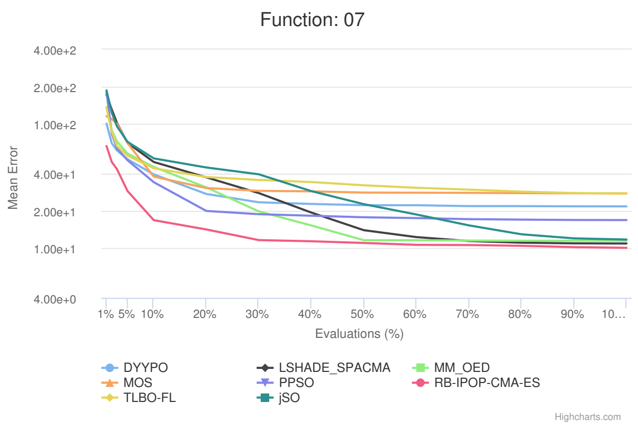


Fig. 12: Convergence plot for Function 7 of the CEC2017 benchmark on 100 dimensions.

[13] 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.

[14] R. Kommadath and P. Kotecha, "Teaching learning based optimization with focused learning and its performance on cec2017 functions," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 2397–2403.

[15] J. Brest, M. S. Mauec, and B. Bokovi, "Single objective real-parameter optimization: Algorithm jso," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1311–1318.

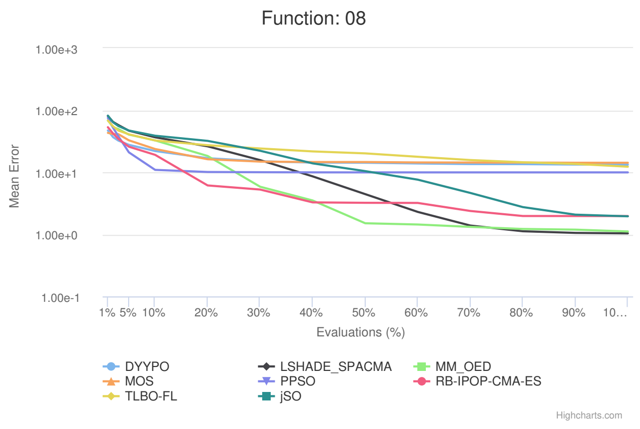


Fig. 13: Convergence plot for Function 8 of the CEC2017 benchmark on 100 dimensions.

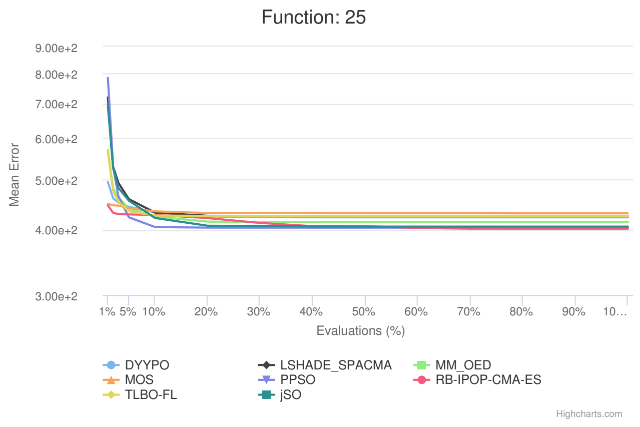


Fig. 14: Convergence plot for Function 25 of the CEC2017 benchmark on 100 dimensions.

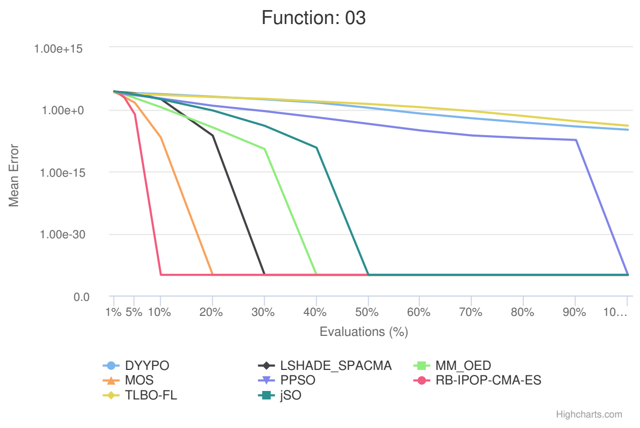


Fig. 15: Convergence plot for Function 3 of the CEC2017 benchmark on 100 dimensions.