Information Sciences 316 (2015) 517–549

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A comprehensive comparison of large scale global optimizers

Antonio LaTorre [a,*], Santiago Muelas [b], José-María Peña [b]

[a] Instituto Cajal, Consejo Superior de Investigaciones Científicas, Madrid, Spain
[b] Dept. of Computer Systems Architecture and Technology, Universidad Politécnica de Madrid, Spain

A R T I C L E   I N F O

Article history:
Received 13 September 2013
Received in revised form 1 September 2014
Accepted 17 September 2014
Available online 2 October 2014

Keywords:
Evolutionary computation
Continuous optimization
Large Scale Global Optimization
Benchmarking

A B S T R A C T

Large Scale Global Optimization is one of the most active research lines in evolutionary and metaheuristic algorithms. In the last five years, several conference sessions and journal special issues have been conducted, and many algorithmic alternatives and hybrid methods, more and more sophisticated, have been proposed. However, most of the proposed algorithms are only evaluated on a particular benchmark of functions and thus its performance in other benchmarks presenting different characteristics remains unknown. In this paper, it is our aim to fill in this gap by evaluating and comparing 10 of the most recently proposed algorithms, in particular, those reporting the best performance in the last major competitions. This paper proposes an evaluation consisting of a broader testbed that considers all the functions of three well-known benchmarks, including a comparative statistical study of the results and the identification of algorithm profiles for those with an equivalent performance. As a part of the comparative analysis this paper also includes three different studies: (1) first, on the complexity of the compared algorithms; (2) then, on the relevance of the comparative statistical tests; and (3) finally, on direct/indirect measures of the exploration/exploitation capabilities of the most representative algorithms in the overall comparison. In addition, this work introduces an open-access web service to perform future analysis and keep trace of new algorithm performances offered to the community of researchers in the field.

1. Introduction

In the last years, there has been an increasing effort in studying the intrinsic characteristics of Large Scale Global Optimization (LSGO) problems in the continuous domain. For this purpose, several special sessions and journal issues, with their corresponding benchmarks, have been proposed. Among all of them, we can highlight the CEC Special Sessions and Competitions held in 2008, 2010, 2012 and 2013, and the Special Issue of the Soft Computing (SOCO) Journal of 2011 on this topic. All these events constitute an important source of information for a LSGO practitioner, as numerous algorithms were tested on each of the benchmarks. Furthermore, for the CEC 2010 and 2012 Special Session and Competition on LSGO the same benchmark of functions was used, allowing the comparison of around 20 different algorithms. However, the existence of several alternative benchmarks increases the fragmentation of the topic and introduces uncertainty on the performance of the algorithms when the problem under consideration changes, even if its dimensionality remains the same.

* Corresponding author.
E-mail addresses: atorre@cajal.csic.es (A. LaTorre), smuelas@fi.upm.es (S. Muelas), jmpena@fi.upm.es (J.-M. Peña).

http://dx.doi.org/10.1016/j.ins.2014.09.031

The purpose of this paper is thus twofold. On the first hand, we aim to provide a comprehensive comparison of the performance of several algorithms evaluated on the different considered benchmarks. For this reason, we have chosen the CEC 2010, CEC 2013 and SOCO 2011 benchmarks and, for each of them, the three algorithms that obtained the best performance. As the CEC 2010 benchmark was also used in the CEC 2012 Special Session and Competition, we decided to also include the best three algorithms of that session. This is important as it was observed that, in general, there was an improvement in the results of the algorithms competing on CEC 2012 special session compared to those of the 2010 edition. This makes an overall of 12 different algorithms. However, two of them could not be executed for different reasons (this will be discussed later). This makes an overall number of 10 algorithms that were exhaustively compared over 3 different benchmarks (with a total number of 54 different 1000-dimension continuous functions), as described further in this paper. On the second hand, a comparative analysis has been performed not only to evaluate the best overall algorithm (from this large set of benchmark functions) but also to analyze performance patterns equivalent among groups of algorithms, studying the common characteristics of the algorithms with similar behavior.

Finally, all the results reported in this paper have been uploaded and classified in an open access website[1] so that they are freely available for all the interested researchers in LSGO. Furthermore, the website allows the users to upload their own results and reports an updated ranking of the performance of all the registered algorithms.

More precisely, the objectives pursued in this work are the following:

- Conduct an extensive study of different algorithms on several LSGO benchmarks. This is especially important as most of the algorithms have not been compared among them nor on such a broad testbed.
- Analyze the performance of the different algorithms both locally (on each independent benchmark) and globally (on all the benchmarks simultaneously).
- Study the generality of the algorithms, i.e., their ability to adapt their performance to different benchmarks without tuning their parameters.
- Provide a common repository for researchers interested in LSGO to retrieve information on this topic as well as to report their own results.

The remainder of this paper is structured as follows. In Section 2, the considered algorithms for each of the benchmarks are briefly described. Section 3 provides all the experimentation details (description of the benchmarks, parameters, stopping criteria, etc.). In Section 4, the experimental results are reported and all the algorithms are compared on the different benchmarks. Finally, Section 5 contains the concluding remarks derived from this study.

2. Preliminaries

This section briefly describes each of the algorithms compared in the present study. To ease the readability and to structure the contents of this section, the algorithms have been grouped according to the benchmark they were first proposed for.

2.1. CEC 2010 Special Session and Competition

In this section we briefly describe the algorithms from the CEC 2010 special session that were selected for comparison. Initially, the three best algorithms were going to be included in the comparative. However, the "Dynamic Multi-Swarm Particle Swarm Optimizer with Sub-Regional Harmony Search" [32] algorithm was finally discarded due to the impossibility of executing the complete set of functions in a reasonable amount of time (some functions took more than twelve hours to run in a state-of-the-art computer such as the one described in Table 4).

2.1.1. MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization (MA-SW-Chains)

The MA-SW-Chains [17] is a Memetic Algorithm for Continuous Optimization that introduces the concept of Local Search Chains to adjust the Local Search (LS) intensity assigned to the particular LS method (the well-known Solis and Wets' algorithm [21] in this case). It uses a Steady State Genetic Algorithm (SSGA) as the population-based algorithm, which implies that individuals may reside in the population for a long time. This situation allows these individuals to become the starting points of subsequent LS invocations. The LS Chains approach used by this algorithm stores, at the end of a LS invocation, the full internal state of the LS method (strategy parameter values, internal variable values) so that a subsequent invocation of the LS method on the same individual may continue at the same point it was interrupted in the previous step. The selection mechanism of the SSGA itself is responsible of maintaining good solutions within the population that would be subject to multiple local improvements. This algorithm obtained the best results in this special session [23]. For a summary of the parameters used by this algorithm and their associated values refer to Table A.18.

[1] http://midas.ctb.upm.es/lab/benchmarks/.

2.1.2. Two-stage based ensemble optimization for Large-Scale Global Optimization (2S-Ensemble)

The algorithm proposed in this work [28] divides the search procedure into two different stages: (1) global shrinking stage and (2) local exploration stage. In the first phase, the algorithms tries to focus the search on a promising area as quickly as possible, whereas the second phase consists in an intense exploitation of the promising area to find the best possible local optimum. The idea behind this approach is to provide explicit balance between exploration and exploitation in the two stages of the algorithm. For this purpose, an EDA based on mixed Gaussian and Cauchy Models (MUEDA) and a Cooperative Co-evolution (CC) algorithm are used for the first and the second stage, respectively. All the parameters and their values can be found in Table A.19. This algorithm was the runner-up algorithm in this special session.

2.2. SOCO 2011 Special Issue

This section introduces the three best performing algorithms of the 2011 Special Issue of the Soft Computing Journal.

2.2.1. A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test (MOS-SOCO2011)

This work [7] describes a MOS-based algorithm combining a Differential Evolution (DE) algorithm and the first of the local searches of the MTS algorithm (MTS-LS1). The proposed approach makes use of the Multiple Offspring (MOS) framework to create the hybrid algorithm. This framework allows the seamless combination of multiple search algorithm in a dynamic way, i.e., adjusting the participation of the considered algorithms as the search process evolves. For this purpose, multiple participation functions and quality measures can be used. In this proposal, a hybrid participation function was used. The idea behind this hybrid function is to quickly adjust the participation of the algorithms if the two considered quality measures agree (i.e., if both measures find one algorithm to be performing better than the other) and, in the case of discrepancies, conduct this adjustment more slowly. For a summary of the parameters of this algorithm and the values used in the experimentation, refer to Table A.20. This algorithm obtained the best overall results among all the algorithms included in this special issue.

2.2.2. Self-adaptive differential evolution algorithm using population size reduction and three strategies (jDElscop)

The jDElscop algorithm [3] is a self-adaptive DE algorithm that uses three different DE strategies (DE/rand/1/bin, DE/rand/1/exp and DE/best/1/bin), a sign-change control mechanism for the F parameter and a new population size reduction mechanism. The three different DE strategies are used alternatively during the search, according to predefined rules. The sign-changing mechanism for the F parameter consists of changing the sign of F at each mutation with a certain probability. Finally, the population size reduction works as follows: the overall number of Fitness Evaluations (FEs) is divided into evenly distributed steps. At the end of each of these steps, the population size is divided by two, keeping the best individuals in a pair-wise comparison of the individuals of the first and second halves of the population. All the parameters used by the different DE strategies, as well as those involved in the adjustment of its parameters, can be checked in Table A.21. This algorithm was the runner-up in this special issue.

2.2.3. Scalability of generalized adaptive differential evolution for large-scale continuous optimization (GaDE)

This paper [31] proposes a generalization of the adaptive DE algorithm called GaDE. This algorithm uses a probability distribution to adapt the value of each of the parameters of the algorithm for each of the individuals of the population. At each generation, the probability distribution is adjusted according to a shift value that is computed from the fitness improvements of each individual of the population in the previous generation. Additionally, this adjustment is weighted according to what the authors call the learning experience and a decreasing factor. A summary of the parameters involved in this algorithm can be found in Table A.22. This algorithm obtained the third place among all the algorithms included in the special issue.

2.3. CEC 2012 Special Session and Competition

In this section the best algorithms of the CEC 2012 Special Session are described. We only include the two best performing algorithms as we were unable to obtain neither the code nor the description of the algorithm with the third best performance (“Octopus Optimization”).

2.3.1. Multiple Offspring Sampling in Large Scale Global Optimization (MOS-CEC2012)

In this work [8], a MOS-based hybrid algorithm is proposed and evaluated. It uses the aforementioned MOS framework to create the algorithm but, in this case, combining two powerful local searches without a population-based algorithm. Furthermore, it uses a simpler participation function that only takes into account a single quality function, the overall fitness increment introduced in the last generation by each of the combined algorithms. Both the parameters of the two local searches as well as those of the hybrid algorithm are reported in Table A.23. This algorithm obtained the best overall results in this special session. Moreover, in a comparison combining the results of CEC 2010 and CEC 2012 sessions (both used the same benchmark) this algorithm also obtained the best results [24].

2.3.2. Self-adaptive differential evolution algorithm with a small and varying population size (jDEsps)

The algorithm presented in this work [2], called *jDEsps*, proposes the use of a self-adaptive DE algorithm, with multiple strategies, a population varying in size and an aging mechanism. This algorithm constitutes an evolution of the algorithm described in Section 2.2.2. The main improvements of this new proposal are:

- The inclusion of two new DE strategies:
  – *jDELS*: A local search procedure applied to the best individual.
  – *jDEw*: A strategy that moves the best individual found so far using a large step movement.
- The variation of the population size adjustment. In this case, the algorithm starts with a small population size, which is increased in the middle of the search to then use the size reduction mechanism described before.

For a summary of all the parameters of this algorithm and their corresponding values refer to Table A.24. This algorithm was the runner-up in this special session.

2.4. CEC 2013 Special Session and Competition

This section briefly describes the three best-performing algorithms of the CEC 2013 Special Session.

2.4.1. Large Scale Global Optimization: experimental results with MOS-based hybrid algorithms (MOS-CEC2013)

This is also a MOS-based hybrid algorithm [9] that combines a population-based search algorithm, a Genetic Algorithm (GA), with two powerful local searches: the Solis Wets' algorithm and a variation of the MTS-LS1 local search, called MTS-LS1-Reduced, specifically proposed for this work. The advantage of this improved local search compared with MTS-LS1 is that it is able to pay more attention to those dimensions of the solutions that contribute the most to the improvement of the quality of the solutions. The parameters of the two local searches, the GA and those of the hybrid algorithm are reported in Table A.25. This was the best performing algorithm in this special session, with a clear difference with regards the other contestants [10].

2.4.2. Large scale evolutionary optimization using cooperative coevolution (DECC-G)

The DECC-G algorithm [30] is a cooperative coevolution algorithm that works dividing large problems into small components that are optimized independently by certain EAs. Since interdependencies may exist between subcomponents, coevolution is a key element of this algorithm to capture such interdependencies during the optimization. Though different EAs can be used to optimize each subcomponent, in this version of the algorithm all the subcomponents are optimized by the same algorithm: a Self-Adaptive with Neighborhood Search Differential Evolution (SaNSDE) algorithm. The SaNSDE algorithm is able to self-adapt the parameters of the DE algorithm and replaces the classic mutation operator of DE by an operator that focuses the search within the neighborhood of the solution being mutated. Although this algorithm has only a few parameters, they have been summarized in Table A.26. The DECC-G algorithm was the runner-up in this special session.

2.4.3. Scaling up Covariance Matrix Adaptation Evolution Strategy using cooperative coevolution (CC-CMA-ES)

The CC-CMA-ES algorithm [12] follows a very similar approach to the DECC-G algorithm. However, the basic optimizer in this case is a Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Analogously, although multiple different EAs could be used to optimize each subcomponent, in this case the CMA-ES algorithm was used to search on every subcomponent. As with the previous algorithm, the few parameters involved in this technique are presented in Table A.27. This algorithm obtained the third position in the CEC 2013 special session.

3. Experimentation

In this experimentation, we have considered three recently proposed benchmarks on Large Scale Global Optimization that were used in the following special issue and conference sessions: Soft Computing Special Issue on "Scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems" [14], CEC 2010 Special Session and Competition on Large-Scale Global Optimization [22] and CEC 2013 Special Session and Competition on Large-Scale Global Optimization [11]. These three benchmarks constitute a good testbed to compare the selected algorithms on a wide number of functions with different design perspectives. In the following subsections, each of the benchmarks will be described. Then, the experimental scenario (number of repetitions, stopping criteria, computer configuration, etc.) will be presented.

3.1. CEC 2010 special session on LSGO benchmark

The CEC 2010 benchmark is composed of 20 continuous optimization functions, divided into four groups:

- Separable functions (f_1–f_3).
- Partially-separable functions, in which a small number of variables are dependent (m-non-separability) while all the remaining ones are independent ($m = 50$) (f_4–f_8).

Table 1
CEC 2010 benchmark functions.

Id	Name
f_1	Shifted Elliptic Function
f_2	Shifted Rastrigin's Function
f_3	Shifted Ackley's Function
f_4	Single-group Shifted and m-rotated Elliptic Function
f_5	Single-group Shifted and m-rotated Rastrigin's Function
f_6	Single-group Shifted and m-rotated Ackley's Function
f_7	Single-group Shifted and m-dimensional Schwefel's Problem 1.2
f_8	Single-group Shifted and m-dimensional Rosenbrock's Function
f_9	$\frac{D}{2m}$-group Shifted and m-rotated Elliptic Function
f_{10}	$\frac{D}{2m}$-group Shifted and m-rotated Rastrigin's Function
f_{11}	$\frac{D}{2m}$-group Shifted and m-rotated Ackley's Function
f_{12}	$\frac{D}{2m}$-group Shifted and m-dimensional Schwefel's Problem 1.2
f_{13}	$\frac{D}{2m}$-group Shifted and m-dimensional Rosenbrock's Function
f_{14}	$\frac{D}{m}$-group Shifted and m-rotated Elliptic Function
f_{15}	$\frac{D}{m}$-group Shifted and m-rotated Rastrigin's Function
f_{16}	$\frac{D}{m}$-group Shifted and m-rotated Ackley's Function
f_{17}	$\frac{D}{m}$-group Shifted and m-rotated Schwefel's Problem 1.2 Function
f_{18}	$\frac{D}{m}$-group Shifted and m-rotated Rosenbrock's Function
f_{19}	Shifted Schwefel's Problem 1.2
f_{20}	Shifted Rosenbrock's Function

- Partially-separable functions that consist of multiple independent subcomponents, each of which is m-non-separable ($m = 50$) (f_9–f_{18}).
- Fully-nonseparable functions (f_{19}–f_{20}).

All the functions of this benchmark are defined for the special case of large dimensionality, $D = 1000$. Table 1 presents the 20 functions making up this benchmark.

3.2. SOCO 2011 special issue benchmark

This benchmark is made up of 19 continuous optimization functions. The first 6 functions were originally proposed for the "*Special Session and Competition on Large Scale Global Optimization*" held at the CEC 2008 Congress [25]. The next 5 functions were proposed for the Workshop on *Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems – A Scalability Test* held at the ISDA 2009 Conference. Finally, the last 7 functions are non-separable functions built by combining two functions belonging to the set of functions f_1–f_{11}. All the functions are completely scalable functions. Table 2 presents the aforementioned 19 functions, whereas a detailed description of the benchmark can be found at the web page of the organizers of the special issue.[2] We did not study the performance of those algorithms only presented at ISDA 2009 Conference [16,18,27], as the authors improved the algorithms for the competition at SOCO 2011.

3.3. CEC 2013 special session on LSGO benchmark

The CEC 2013 benchmark is an evolution of the CEC 2010 one. The main new characteristics compared to the previous benchmark are:

- Nonuniform subcomponent sizes.
- Imbalance in the contribution of subcomponents.
- Functions with overlapping subcomponents.
- New transformations to the base functions:
  - Ill-conditioning.
  - Symmetry breaking.
  - Irregularities.

This benchmark is composed of 15 functions that are divided into four different groups according to their characteristics:

- Fully-Separable functions (f_1–f_3).
- Partially-separable functions.

[2] http://sci2s.ugr.es/eamhco/CFP.php.

Table 2
SOCO 2011 benchmark functions.

Id	Name
f_1	Shifted Sphere Function
f_2	Shifted Schwefel's Problem 2.21
f_3	Shifted Rosenbrock's Function
f_4	Shifted Rastrigin's Function
f_5	Shifted Griewank's Function
f_6	Shifted Ackley's Function
f_7	Schwefel's Problem 2.22
f_8	Schwefel's Problem 1.2
f_9	Extended f_{10}
f_{10}	Bohachevsky
f_{11}	Schaffer
f_{12}	Hybrid f_9 & f_1 (25%, 75%)
f_{13}	Hybrid f_9 & f_3 (25%, 75%)
f_{14}	Hybrid f_9 & f_4 (25%, 75%)
f_{15}	Hybrid f_{10} & f_7 (25%, 75%)
f_{16}	Hybrid f_9 & f_1 (50%, 50%)
f_{17}	Hybrid f_9 & f_3 (75%, 25%)
f_{18}	Hybrid f_9 & f_4 (75%, 25%)
f_{19}	Hybrid f_{10} & f_7 (75%, 25%)

Table 3
CEC 2013 benchmark functions.

Id	Name
f_1	Elliptic Function
f_2	Rastrigin's Function
f_3	Ackley's Function
f_4	Partially Separable Elliptic Function
f_5	Partially Separable Rastrigin's Function
f_6	Partially Separable Ackley's Function
f_7	Partially Separable Schwefel's Problem 1.2
f_8	Non-Separable Elliptic Function
f_9	Non-Separable Rastrigin's Function
f_{10}	Non-Separable Ackley's Function
f_{11}	Non-Separable Schwefel's Problem 1.2
f_{12}	Rosenbrock's Function
f_{13}	Schwefel's Function with Conforming Overlapping Subcomponents
f_{14}	Schwefel's Function with Conflicting Overlapping Subcomponents
f_{15}	Schwefel's Problem 1.2

– Partially separable functions with a set of non-separable subcomponents and one fully-separable subcomponent (f_4–f_7).
– Partially separable functions with only a set of non-separable subcomponents and no fully-separable subcomponent (f_8–f_{11}).

- Functions with overlapping subcomponents.
  - Overlapping functions with conforming subcomponents: the optimization of one subcomponent may improve the value of the other subcomponent due to the optimization of the shared decision variables (f_{12}–f_{13}).
  - Overlapping functions with conflicting subcomponents: the optimization of one subcomponent may have a detrimental effect on the other overlapping subcomponent due to the conflicting nature of the shared decision variables (f_{14}).

- Fully-nonseparable functions (f_{15}).

Table 3 presents the functions making up this benchmark.

3.4. Experimental scenario

The results reported for this work are the average of 25 independent executions conducted on the computer configuration displayed in Table 4. For every algorithm and function, five statistics have been gathered: best, worst, mean, median and standard deviation.

Table 4
Computer configuration.

PC	Intel Xeon 8 cores 1.86 GHz CPU 22 GB RAM
Operating System	Ubuntu Linux 11.10
Prog. Language	C++ & Matlab
C/C++ Compiler	Clang 2.9
Matlab Version	Matlab R2011B

The stopping criteria are different depending on the benchmark. In the case of the SOCO 2011 benchmark, the maximum number of allowed Fitness Evaluations (FEs) is 5M, whereas for both CEC 2010 and 2013 benchmarks it is 3M FEs.

Finally, all the algorithms have been run with the original parameters values reported for the benchmark they were initially proposed. This means that no further parameter tuning has been conducted on the algorithms. There are several reasons for this. First, adjusting the parameters values for every algorithm would have dramatically increased the required computation time, especially for the algorithms implemented in Matlab. Second, tuning the parameters values for the whole set of functions (considering all the three benchmarks simultaneously as a "super-benchmark") would lead to overfitting. Keeping the original parameters values allows us to study the generality and sensibility to differences in the characteristics of the problems of the algorithms under study. For a summary of the parameters of the algorithms compared, the reader should refer to Tables A.18–A.27. or to the original papers.

4. Analysis of the results

4.1. Algorithms complexity analysis

Prior to reporting the results of the considered algorithms on the different selected benchmarks, we provide a complexity analysis of the different algorithms. However, as these algorithms are, in general, complex pieces of software with many components, alternative branches, etc., it is difficult to conduct a complexity analysis from an order of growth (i.e., using Big O notation) perspective. Instead, we provide the following two measures that give a good overall estimation on the complexity of the algorithms taken into account.

The first measure is the cyclomatic complexity (or conditional complexity, or McCabe's complexity) [15], which is a software metric that measures the number of independent paths in a program's source code. The higher the number of independent paths are, the more complex the program is and, thus, a higher complexity value is obtained. To compute these values, the internal "checkkode" Matlab function has been used for the algorithms programmed in that language, and the "lizard" tool for code in the C/C++ programming language.[3]

The second measure is the running time for both the fastest and the slowest functions in the overall "super-benchmark" (considering the three benchmarks simultaneously) for each of the algorithms.

Table 5 contains these two measures for all the compared algorithms. It can be observed that, in general, a low cyclomatic complexity implies lower running times. However, this correlation is not perfect, as algorithms with similar complexity exhibit different scalability properties. For example, MOS-SOCO2011 algorithm presents the lowest cyclomatic complexity and running time for the fastest function but, on the other hand, it does not have the lowest running time for the slowest function. In the case of GaDE, it also has a low cyclomatic complexity value but its running times seem to be more stable with regards to the running time. On the other side, algorithms with higher cyclomatic complexity normally present higher running times. This is the case for DECC-G, CC-CMA-ES and 2S-Ensemble (although the running time for the slowest function is not as high for this algorithm as for the other two). However, this does not seem to be the case for jDEsps, which holds a relatively high cyclomatic complexity value (5.12) while it exhibits very good running times. Finally, the case of the jDElscop algorithm is also interesting, as it has a moderate cyclomatic complexity value (3.60), a good running time for the fastest function but one of the worst running times for the slowest function.

If we pay attention to the programming languages used for the 10 considered algorithms, we can observe that the 3 slowest algorithms for the fastest functions, which are also 3 out of the 4 slowest algorithms for the slowest function, have been programmed in the Matlab programming language. This is an expected result, as Matlab is known to be considerably slower than other lower-level programming languages, but if we observe the cyclomatic complexity values, we can see that again 3 out of the 4 algorithms with higher values have also in common Matlab as their programming language. On the other hand, the 7 remaining algorithms are all programmed in the C++ language and they all share low complexity values (except for jDEsps), fast running times for the fastest function (of course, with some variability, but always within the same order of magnitude) and fast running times for the slowest function (with the exception of the jDElscop algorithm).

So, to summarize, it seems that, at least for this kind of large-scale global optimization problems, a lower-level programming language, such as C++, should be preferred if we want a good performance, although attention should also be paid at the complexity of the algorithm to avoid cases such as that of jDElscop for the slowest function.

[3] https://github.com/terryyin/lizard.

Table 5
Cyclomatic complexity and running times (for both the fastest and the slowest functions) for each of the compared algorithms.

	Cyclomatic complexity	Running time (fastest function) (s)	Running time (slowest function) (s)
MOS-SOCO2011	2.50	72.57	5403.23
MA-SW-Chains	2.85	451.31	3534.46
GaDE	2.86	532.60	2323.20
MOS-CEC2012	3.20	715.07	5971.32
MOS-CEC2013	3.24	290.03	6023.70
jDElscop	3.60	205.97	26423.83
DECC-G	4.91	1344.67	20888.04
jDEsps	5.12	142.37	3301.23
CC-CMA-ES	6.16	1658.18	245111.87
2S-Ensemble	7.01	1322.16	9937.16

Table 6
Average ranking, number of functions for which the algorithm obtains the best results and number of wins (nWins) in pair-wise comparisons with the other algorithms on the CEC 2010 benchmark.

	Ranking	# Best	nWins
MOS-CEC2013	3.42	5	8
jDEsps	3.50	6	5
MA-SW-Chains	4.67	0	3
MOS-SOCO2011	5.00	5	1
MOS-CEC2012	5.08	6	0
2S-Ensemble	5.35	3	0
jDElscop	5.50	1	0
CC-CMA-ES	6.15	0	−1
GaDE	7.88	1	−8
DECC-G	8.45	0	−8

4.2. Individual benchmarks analysis

In this section we report the results of all the analyzed algorithms for each of the three considered benchmarks. As we have already commented, for each of the benchmarks, we report the best, median, worst, mean and standard deviation of the error for every function. Highlighted cells represent the best results for that function. Furthermore, we have used the Friedman test for multiple comparisons to check if there are significant differences among the considered algorithms. If such differences exist, we provide the following values for each algorithm:

- Overall ranking according to the Friedman test: we compute the relative ranking of each algorithm according to its mean performance for each function and report the average ranking computed through all the functions. Given the following mean performance in a benchmark of three functions for algorithms A and B: A = (0.00e+00, 1.27e+01, 3.54e−03), B = (3.72e+01, 0.42e+00, 2.19e−07); their relative ranking would be: $Ranks_A = (1,2,2), Ranks_B = (2,1,1)$; and thus their corresponding average rankings are: $R_A = 1.67$ and $R_B = 1.33$.
- # Best: This is the number of functions for which each algorithm obtains the best results compared to all the other algorithms.
- nWins: This is the number of other algorithms for which each algorithm is statistically better minus the number of algorithms for which each algorithm is statistically worse according to the Wilcoxon Signed Rank Test in a pair-wise comparison [19].

Finally, we also report the *p-values* for two statistical tests (Friedman based *p*-value [5,6] and Wilcoxon [29]) and their corresponding adjusted values to control the familywise error rate according to the Holm procedure [6]. This is the recommended statistical validation procedure used in the past in many other studies [6].

Tables B.28–B.30 show the final results of each algorithm on all the functions of the CEC 2010 benchmark. From these data, the average ranking, the number of functions with best results and the nWins value are computed and reported in Table 6. The algorithm that obtains the best ranking and the nWins value is the MOS-CEC2013 algorithm. However, there are two other algorithms with a higher number of functions for which they obtain the best results: jDEsps and MOS-CEC2012. Nonetheless, the difference is of just one function, which cannot be considered significant.

The Friedman test reported a *p-value* = 2.66E−08 for this benchmark, which is below the significance level, and means that there are statistical differences that must be checked with the two aforementioned tests. Table 7 contains the results of the statistical tests used to compare the algorithms. According to the previously shown ranking, the MOS-CEC2013 algorithm has been used as the control algorithm. The adjusted *p-value* according to the Friedman statistic reports significant differences between the MOS-CEC2013 algorithm and other 3 algorithms, whereas the adjusted *p-value* according to the Wilcoxon test reveals significant differences with two more algorithms, to a total of 5. For the remaining 4 algorithms, the *p-value* of three of them are nonetheless very close to the significance level.

Table 7
Statistical validation for the CEC 2010 benchmark (MOS-CEC2013 is the control algorithm).

MOS-CEC2013 vs.	Friedman p-value		Wilcoxon p-value	
	Raw	Adjusted	Raw	Adjusted
jDEsps	9.38E−01	9.38E−01	5.22E−01	5.22E−01
MA-SW-Chains	1.92E−01	3.83E−01	2.00E−02	8.00E−02
MOS-SOCO2011	1.00E−01	3.39E−01	1.24E−03	8.65E−03[a]
MOS-CEC2012	8.48E−02	3.39E−01	2.21E−02	8.00E−02
2S-Ensemble	4.44E−02	2.22E−01	2.91E−02	8.00E−02
jDElscop	3.02E−02	1.81E−01	2.75E−03	1.65E−02[a]
CC-CMA-ES	4.42E−03	3.10E−02[a]	3.19E−03	1.65E−02[a]
GaDE	3.35E−06	2.68E−05[a]	7.75E−05	6.20E−04[a]
DECC-G	1.53E−07	1.38E−06[a]	1.81E−05	1.63E−04[a]

[a] There are statistical differences with significance level $\alpha = 0.05$.

Analogously, Tables B.31–B.33 show the final results of each algorithm on all the functions of the SOCO 2011 benchmark. The same three statistics are reported in Table 8. In this case, the algorithm with the best average ranking, number of functions with best results and nWins value is the MOS-SOCO2011 algorithm. In this benchmark, the performance differences are more obvious than in the previous one, with one algorithm clearly reporting the best results in most of the functions and, consequently, in the average ranking and the nWins value. The two algorithms that obtained the best results in the CEC 2010 benchmark show now an average performance, ranked 5th and 6th, respectively.

The Friedman test reported a *p-value* = 1.30E−11 this time, which is also below the significance level, and thus the same tests must be used, which yield the results depicted in Table 9. These results confirm the superior performance of the MOS-SOCO2011 algorithm, which is statistically better than 7 out of 9 algorithms when taking into account the adjusted Friedman *p-value* and better than any of the remaining algorithms when using the adjusted Wilcoxon *p-value*. The only algorithms not getting significant results with the first test are jDElscop and GaDE. However, the *p-value* for the GaDE algorithm is very close to the significance level $\alpha = 0.05$.

It should be noted that, in this benchmark, the three algorithms taking the top three rankings are the algorithms originally proposed for this benchmark, as opposed to the other two benchmarks. In contrast, for the CEC 2010 benchmark, the best algorithm was not proposed in that special session, whereas for the CEC 2013 benchmark, two of the three best algorithms were proposed for the CEC 2010 special session, as we will show in the following paragraphs.

Table 8
Average ranking, number of functions for which the algorithm obtains the best results and number of wins (nWins) in pair-wise comparisons with the other algorithms on the SOCO 2011 benchmark.

	Ranking	# Best	nWins
MOS-SOCO2011	2.00	15	9
jDElscop	3.11	8	6
GaDE	4.16	2	4
2S-Ensemble	5.08	4	−1
jDEsps	5.47	0	4
MOS-CEC2013	6.16	4	−4
DECC-G	7.03	3	−4
CC-CMA-ES	7.21	2	−5
MA-SW-Chains	7.24	0	−5
MOS-CEC2012	7.55	2	−4

Table 9
Statistical validation for the SOCO 2011 benchmark (MOS-SOCO2011 is the control algorithm).

MOS-SOCO2011 vs.	Friedman p-value		Wilcoxon p-value	
	Raw	Adjusted	Raw	Adjusted
jDElscop	2.61E−01	2.61E−01	2.52E−02	2.52E−02[a]
GaDE	2.80E−02	5.61E−02	2.43E−03	9.71E−03[a]
2S-Ensemble	1.72E−03	5.17E−03[a]	1.35E−03	8.11E−03[a]
jDEsps	4.06E−04	1.62E−03[a]	1.91E−06	1.72E−05[a]
MOS-CEC2013	2.31E−05	1.15E−04[a]	8.53E−03	1.71E−02[a]
DECC-G	3.11E−07	1.86E−06[a]	4.30E−03	1.29E−02[a]
CC-CMA-ES	1.13E−07	7.91E−07[a]	1.46E−04	1.02E−03[a]
MA-SW-Chains	9.76E−08	7.80E−07[a]	1.91E−06	1.72E−05[a]
MOS-CEC2012	1.58E−08	1.42E−07[a]	1.64E−03	8.21E−03[a]

[a] There are statistical differences with significance level $\alpha = 0.05$.

Finally, Tables B.34–B.36 present the results for the CEC 2013 benchmark. The same analysis as in the previous two benchmarks has been conducted and the results are presented in Table 10. For this benchmark, the best overall algorithm according to the three statistics is the MOS-CEC2013 algorithm. Moreover, the runner-up and the third algorithm are two contributions to the CEC 2010 special session, that were specifically and originally designed for the CEC 2010 benchmark. Furthermore, to find the first of the algorithms presented in the SOCO 2011 special issue we have to go down until the 6th (jDElscop), 8th (MOS-SOCO2011) and 9th (GaDE) positions. This distribution of the algorithms proposed for the different benchmarks proposes that CEC 2010 and 2013 benchmarks are quite similar among them, regardless of the different strategies followed for the combination of the basic functions, whereas the SOCO 2011 benchmark is different by design to these two other benchmarks. In Section 4.3 we will graphically depict this behavior.

Regarding the statistical validation of the results for the CEC 2013 benchmark, the Friedman test reported a *p-value* = 5.29E−04, thus existing significant differences that must be studied with the aforementioned tests (Table 11). The adjusted Friedman *p-value* yields significant differences for 4 of the algorithms, whereas the adjusted Wilcoxon *p-value* increases this value to 6 algorithms. Nonetheless, for the remaining algorithms the results are not statistically significant, although the *p-values* remain relatively low (always below 3.0E−01 for Friedman statistic and below 1.5E−01 for Wilcoxon statistic).

4.3. Overall analysis

In this section we conduct an overall analysis of the results, considering the three sets of functions as one large benchmark. Although some of the benchmarks include similar basic functions, we have decided to keep all the functions in this comparison as these apparently colliding functions do not present always the same perturbing characteristics (rotation, shifting of the global optimum, noise, etc.). Furthermore, the results reported in Tables B.28–B.36 show that some algorithms obtain significantly different error values (sometimes several orders of magnitude) for the same basic function with different perturbations. As a consequence of this, the discussion conducted in this section takes into account the results on a set of 54 functions ($20 + 19 + 15$).

On the first hand, we have carried out the same validation process that we did for the individual benchmarks. Then we compare both statistical tests used (Friedman and Wilcoxon) and conduct a power analysis to decide which one should be preferred for this kind of comparisons. Next, we conduct an exploration/exploitation analysis on four selected algorithms that exhibit similar performance to check if that can explain this similar performance. To continue this thorough analysis, we have divided the overall set of functions into multiple groups according to their inherent characteristics and carried out a new comparison at this level. To conclude this study, we have measured and graphically depicted the correlation in the performance of the algorithms considered in this work.

Table 10
Average ranking, number of functions for which the algorithm obtains the best results and number of wins (nWins) in pair-wise comparisons with the other algorithms on the CEC 2013 benchmark.

	Ranking	# Best	nWins
MOS-CEC2013	3.00	8	7
MA-SW-Chains	4.33	1	5
2S-Ensemble	4.67	1	2
jDEsps	4.93	2	2
MOS-CEC2012	5.20	2	1
jDElscop	5.60	1	−2
CC-CMA-ES	5.93	0	−3
MOS-SOCO2011	6.40	2	−3
GaDE	7.00	2	−4
DECC-G	7.93	0	−5

Table 11
Statistical validation for the CEC 2013 benchmark (MOS-CEC2013 is the control algorithm).

MOS-CEC2013 vs.	Friedman *p*-value		Wilcoxon *p*-value	
	Raw	Adjusted	Raw	Adjusted
MA-SW-Chains	2.28E−01	2.63E−01	7.57E−02	1.26E−01
2S-Ensemble	1.32E−01	2.63E−01	2.40E−02	9.58E−02
jDEsps	8.03E−02	2.41E−01	6.03E−02	1.26E−01
MOS-CEC2012	4.66E−02	1.86E−01	7.83E−03	3.91E−02[a]
jDElscop	1.87E−02	9.34E−02	4.21E−02	1.26E−01
CC-CMA-ES	7.97E−03	4.78E−02[a]	1.01E−03	7.55E−03[a]
MOS-SOCO2011	2.10E−03	1.47E−02[a]	1.16E−03	7.55E−03[a]
GaDE	2.97E−04	2.37E−03[a]	9.44E−04	7.55E−03[a]
DECC-G	8.11E−06	7.30E−05[a]	3.05E−05	2.75E−04[a]

[a] There are statistical differences with significance level $\alpha = 0.05$.

Table 12
Average ranking, number of functions for which the algorithm obtains the best results and number of wins (nWins) in pair-wise comparisons with the other algorithms on the whole set of benchmarks.

	Ranking	# Best	nWins
MOS-CEC2013	4.27	17	7
MOS-SOCO2011	4.33	22	4
jDEsps	4.59	8	2
jDElscop	4.69	10	2
2S-Ensemble	5.06	8	1
MA-SW-Chains	5.48	1	1
MOS-CEC2012	5.98	10	1
GaDE	6.32	5	−4
CC-CMA-ES	6.46	2	−6
DECC-G	7.81	3	−8

4.3.1. Statistical analysis

Table 12 shows the average ranking, number of functions with best results and nWins value for the whole set of 54 functions. According to this table, the two algorithms with the best overall performance (for all the three statistics) are the MOS-CEC2013 and the MOS-SOCO2011 algorithms, respectively. The MOS-CEC2013 algorithm gets the best overall ranking and nWins value, whereas the MOS-SOCO2011 is the best algorithm for a higher number of functions. In 3rd and 4th places we have jDEsps and jDElscop, which are very similar among them as jDEsps was conceived as an evolution of the jDElscop. Similarly to the case of the two MOS-based algorithms, jDEsps gets a better overall ranking, whereas jDElscop reports the best results for a highest number of functions. It is interesting to note how the two best ranked algorithms share a similar approach (combining a population-based algorithm and a powerful, or two, local searches), whereas the other two algorithms combine several DE strategies and adjust their parameters along with the population size. It is interesting to remark how these two different approaches obtain similar results, especially in the case of the SOCO 2011 benchmark, where the use of a strong local search seemed to be necessary due to the large number of functions that can be easily optimized dimension by dimension. It could be the case that the use of different DE strategies, the self-adjustment of their parameters or the varying population size could replace the exploitation capabilities of the local searches, but it may also be the case that a combination of all these characteristics provides the algorithm with this ability. Further research to isolate each of the composing parts of the jDEsps and jDElscop algorithms seems to be necessary to shed light on this issue.

The Friedman test was run for all the algorithms on the whole set of functions, obtaining a *p-value* of 1.46E−11. The results of the statistical tests considering all the 54 functions are reported in Table 13. The Friedman test reports significant differences in 4 out of 9 algorithms, whereas the Wilcoxon test reports significant differences for 5 algorithms. Furthermore, the adjusted *p-value* according to the Wilcoxon test are very low (close to 5.0E−02) for two of the remaining algorithms, and low (below 1.5E−01) for another algorithm.

4.3.2. Power analysis for Friedman and Wilcoxon tests

To end this statistical analysis, we will discuss the differences in the *p*-values reported by both the Friedman and the Wilcoxon tests. As can be seen in the current and previous sections, both tests exhibit small discrepancies for some comparisons. In general, Wilcoxon test is able to report more statistical differences than Friedman test can. This is due to the way both test work. As described in [1], the Friedman test takes into account just the relative ranking of the algorithms under comparison to compute the statistic. On the other hand, the Wilcoxon algorithm not only considers the relative ranking but also the quantitative differences in the performance of the algorithms and thus is able to find more differences than Friedman's test can. Let us see these differences with a good example, in this case with the MOS-SOCO2011 and MOS-SOCO2013 algorithms on the CEC 2010 benchmark. According to Table 6, MOS-CEC2013 has an average ranking of 3.42, whereas MOS-SOCO2011 has an average ranking of 5.00. A difference of around 1.5 in the average ranking could seem like a small performance dif-

Table 13
Statistical validation for the whole set of benchmarks (MOS-CEC2013 is the control algorithm).

MOS-CEC2013 vs.	Friedman *p*-value		Wilcoxon *p*-value	
	Raw	Adjusted	Raw	Adjusted
MOS-SOCO2011	9.11E−01	1.00E+00	1.07E−03	5.37E−03[a]
jDEsps	5.78E−01	1.00E+00	5.00E−01	5.00E−01
jDElscop	4.75E−01	1.00E+00	7.18E−02	1.44E−01
2S-Ensemble	1.72E−01	6.87E−01	1.55E−02	6.18E−02
MA-SW-Chains	3.74E−02	1.87E−01	2.51E−02	7.52E−02
MOS-CEC2012	3.28E−03	1.97E−02[a]	4.23E−04	2.96E−03[a]
GaDE	4.19E−04	2.93E−03[a]	9.20E−05	7.36E−04[a]
CC-CMA-ES	1.66E−04	1.33E−03[a]	4.58E−04	2.96E−03[a]
DECC-G	1.28E−09	1.15E−08[a]	2.18E−06	1.96E−05[a]

[a] There are statistical differences with significance level $\alpha = 0.05$

ference, which is actually what Friedman test reports. What happens if we take into account not only rankings but also quantitative differences? Table 15 depicts these differences, with a positive number indicating that the MOS-CEC2013 algorithm is better in that number of orders of magnitude, whereas a negative number represents just the opposite thing. A value of 0 represents a tie, i.e., both algorithms reported errors in the same order of magnitude. As can be seen, MOS-CEC2013 obtains differences of one or more orders of magnitude in 10 out of 20 functions. On the other hand, MOS-SOCO2011 gets a difference of one or more orders of magnitude for 3 out of 20 functions. Finally, both algorithms report an average error of the same order of magnitude for 7 of 20 functions. Furthermore, we can observe that not only MOS-CEC2013 exhibits significant better performance on much more functions, but also these differences are significantly higher, especially for functions 6, 7, 8, 12 and 17.

As a complement to this quantitative comparison, we have also conducted a power analysis of both tests, which allows the estimation of the probability to avoid a Type II error (i.e., the probability of correctly rejecting a false null hypothesis). As computing the exact power of the Wilcoxon test is too expensive [4], to carry out this power analysis we have followed one of the bootstrapping methods described in [4] (in particular, the one in which both Xs and Ys are simultaneously sampled to produce the empirical distribution). This procedure allows the estimation of the power of a statistical test by sampling the original distribution of values and computing the proportion of experiments in which the test rejected the null hypothesis. To minimize the noise that may be introduced by the sampling mechanism, this procedure has been repeated 100 times for each algorithm and benchmark and the values reported for each benchmark are the average values of the 100 iterations of the 10 algorithms.

Table 14 shows the estimated power of each test for each benchmark and the average of these three values. As can be seen, Wilcoxon test obtains significantly better estimated power values than Friedman test, especially in the case of CEC 2010 and CEC 2013 benchmarks.

For all these reasons, Wilcoxon test should be preferred to Friedman test due to its ability to better detect significant differences.

4.3.3. Exploration vs. exploitation analysis

Fig. 1 depicts in a radar chart the average ranking of each algorithm on each benchmark and in the whole set of functions. As an average ranking is better when it has a smaller value, in this chart the best algorithms have a smaller area.

To ease the comprehension of the plot, we have divided this chart into two plots that only contain the algorithms without and with significant different performance (compared to the reference algorithm MOS-CEC2013) in Figs. 2 and 3, respectively. In the first plot we find an interesting pattern: the MOS-SOCO2011 and the jDElscop algorithms have a similar shape, which means that they behave similarly on the same benchmarks. This is also the case for the MOS-CEC2013 and the jDEsps algorithms, with a small divergence for the CEC 2013 benchmark. Graphically, it can be seen as if the two first algorithms lay on *Y* axis, whereas the other two algorithms lay on the *X* axis.

To better understand why this behavior emerges, we have conducted a comparative analysis from the perspective of the exploration/exploitation balance [20]. Direct measures to analyze the exploration capabilities of an algorithm barely exist [13]. Thus, carrying out this sort of analysis is a difficult task. At this point, we decided to conduct this study in a similar way to the one described in [13]. In this paper, the authors propose a measure called the "Exploration ratio", which is computed by using an ancestry tree and calculating the percentage of nodes in the tree for which the distance between parent and children individuals is over a threshold. In our case, instead of computing the Exploration ratio, which requires to fix an arbitrary threshold for each function, we have directly compared the distribution of distances between parents and children in the ancestry tree by means of the Wilcoxon non-parametric statistical test. However, as the number of distance values is very large (millions of new individuals) we have carried out the comparisons by sampling from the original distances distributions. To avoid any bias introduced by the sampling method (uniform sampling) we have repeated the process 100 times. Table 16 summarizes the results obtained by means of this procedure. This table shows, for each of the four compar-

Table 14
Average power value for each test and benchmark (in bold, significantly better estimated power values).

	CEC 2010	SOCO 2011	CEC 2013	Average
Friedman	0.37	0.82	0.44	0.54
Wilcoxon	**0.63**	**0.84**	**0.63**	**0.70**

Table 15
Quantitative performance comparison (in orders of magnitude) for MOS-CEC2013 and MOS-SOCO2011 on the CEC 2010 benchmark. Positive numbers represent better performance (in orders of magnitude) of the MOS-CEC2013 algorithm, whereas negative numbers represent better performance of the MOS-SOCO2011 algorithm.

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
0	0	−2	+1	0	+14	+21	+6	+1	0
F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
0	+17	0	+1	+1	0	+10	−1	+1	−1

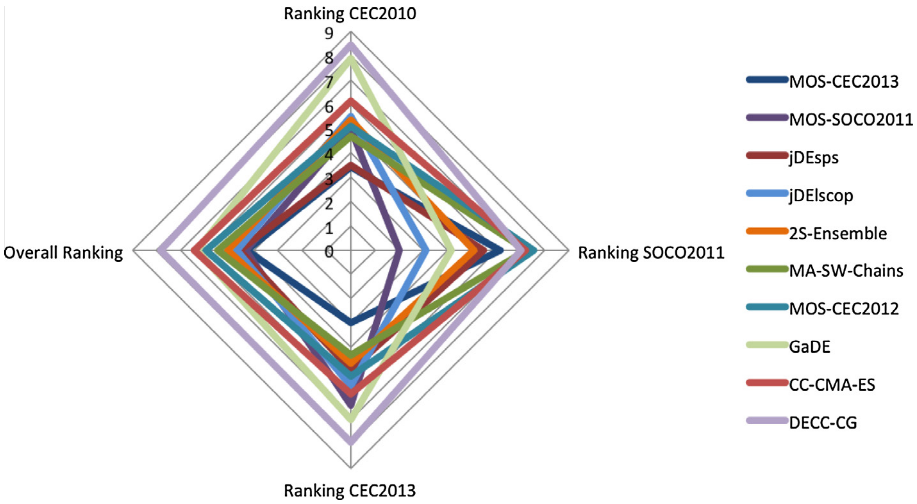

Fig. 1. Radar chart for the average rankings on the different benchmarks and the whole set of functions for all the algorithms.

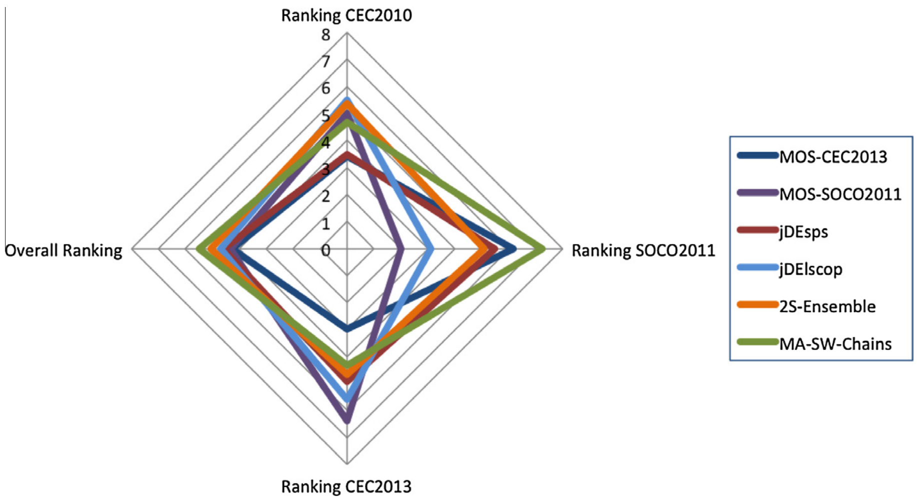

Fig. 2. Radar chart for the average rankings on the different benchmarks and the whole set of functions for the algorithms with no significant difference in performance.

isons, the number of functions for which the differences in the distances distributions (i.e., the exploratory capabilities of each algorithm) are statistically different. As can be observed, the MOS-CEC2013 algorithm has a higher exploration capability than both the jDElscop and the MOS-SOCO2011 algorithms. However, this is not the case for the jDEsps, which presents better exploratory ability than the other two algorithms in a marginal number of functions. Furthermore, if we conduct the tests with the opposite hypothesis, we will obtain significant results for most of the functions with both the jDElscop and MOS-SOCO2011 algorithms. The full results of this statistical comparison can be found in Tables C.37 and C.38.

With the objective of complementing the results obtained from the direct measures we have also compared the influence of the different parts of each of these four algorithms by means of some indirect measures that, to a greater or a lesser extent, can reflect the exploration/exploitation ability of an algorithm. The objective of this complementary study is to clarify those cases in which direct measures do not cast concluding results. The different parts of the compared algorithms are as follows: jDEbest, jDEbin and jDEexp (three DE algorithms with different configurations/operators) for jDElscop; DE and MTS-LS1 for MOS-SOCO2011; jDEbest, jDEbin, jDEexp and jDELS (the same three DE configurations as mentioned before and one local search) for jDEsps; and GA, MTS-LS1 and SW for MOS-CEC2013. Figs. 4 and 5 depict the following three measures for each of the four considered algorithms on a sample function coming from the CEC 2013 benchmark: fitness decrement (as these are all minimization functions), number of improvements to the best so far solution and diversity increment/decrement (measured as the average euclidean distance of all the solutions in the population).

If we analyze these figures, we can observe two interesting patterns. First, there seems to be a clear balance in the exploration/exploitation effort for all the four algorithms. We can conclude this by analyzing the fitness decrement and the num-

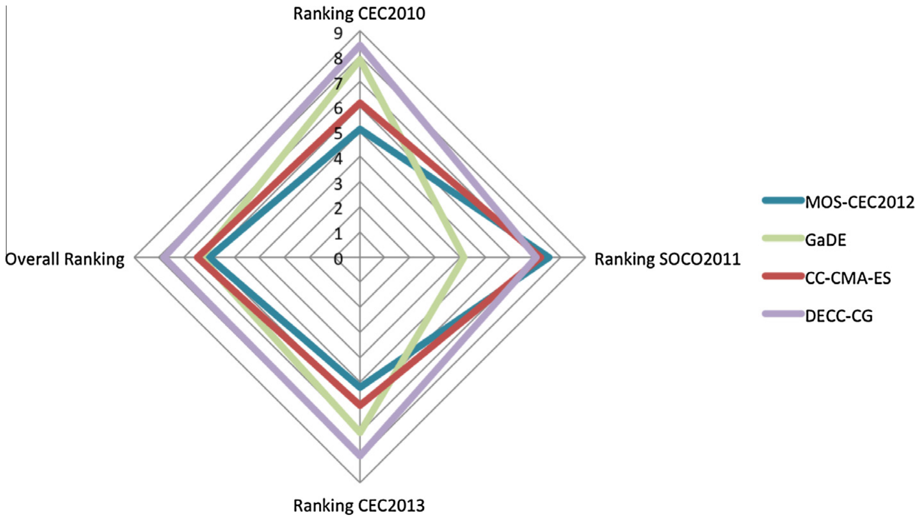

Fig. 3. Radar chart for the average rankings on the different benchmarks and the whole set of functions for the algorithms with significant different performance.

Table 16
Summary of the comparative analysis of the exploration capabilities of the four selected algorithms.

	CEC 2010	SOCO 2011	CEC 2013
jDEsps vs. jDElscop	1/20	1/19	0/15
jDEsps vs. MOS-SOCO2011	0/20	1/19	2/15
MOS-CEC2013 vs. jDElscop	20/20	19/19	15/15
MOS-CEC2013 vs. MOS-SOCO2011	19/20	19/19	15/15

ber of improvements plots. In all the cases, there is at least one component of each algorithm that conducts less improvements but of higher intensity (exploration). Analogously, there is at least another component of each algorithm that behaves in the fully opposite way: it conducts a lot of improvements but of low intensity (exploitation). Now the question is why jDElscop and MOS-SOCO2011, and jDEsps and MOS-SOCO2013 perform similarly between them, respectively, but not between the two groups. One possible explanation to this behavior may be related to the ability of the algorithms to preserve the diversity of the populations they manage [20]. As none of the compared algorithms include an explicit mechanism to increase diversity such as, for example, population reset, it is responsibility of the operators of the algorithms to maintain this diversity. If we take a look at the diversity increment/decrement plot for each algorithm, we can observe that the first two algorithms (jDElscop and MOS-SOCO2011) are unable to increase the diversity of the population after an early initial exploration phase. The effect of the lack of diversity is more clearly appreciated in the jDElscop algorithm as the jDEexp component reduces the diversity of the population, whereas its exploitative counterpart in the MOS-SOCO2011 algorithm, the MTS-LS1 technique, does not have this effect as it is applied only to the best solution in the population thus not significantly affecting the overall diversity. On the other hand, the second group of algorithms (jDEsps and MOS-CEC2013) are capable of increasing the diversity of the population at later stages of the search, which allows these two algorithms to better explore the search space. This is particularly interesting in the case of the MOS-CEC2013, as this algorithm dynamically adjusts the participation of each of its composing algorithms. This means that, after the initial exploration phase, the participation of the GA algorithm, the exploration component of this algorithm, gets its participation reduced, thus decreasing the number of new solutions that it is allowed to create. Nevertheless, the mechanisms provided by the MOS framework (periodical adjustment of participation, minimum participation ratio) allows the algorithm to still allocate a minimum exploration capability to avoid premature convergence.

This behavior, with small variations, can be observed for all the algorithms on all the functions. This can explain the similar behavior of jDElscop and MOS-SOCO2011, and jDEsps and MOS-CEC2013, respectively. The first two algorithms are especially good in solving problems that can be easily solved dimension by dimension (heavy exploitation after quick exploration, as many of the functions of the SOCO 2011 benchmark), whereas the second two algorithms can better deal with functions that require successive exploration stages (as most of the CEC 2010 and CEC 2013 do).

To conclude, it should be remarked that the different highlights shown by direct and indirect measures may be explained due to not only their ability to produce a more diverse offspring but to the relative quality or survival probabilities due to evolutionary pressure. Whereas producing individuals with significatively different genotypes improves diversity, if these individuals do not contribute to the next generation offspring, this diversity is wasted.

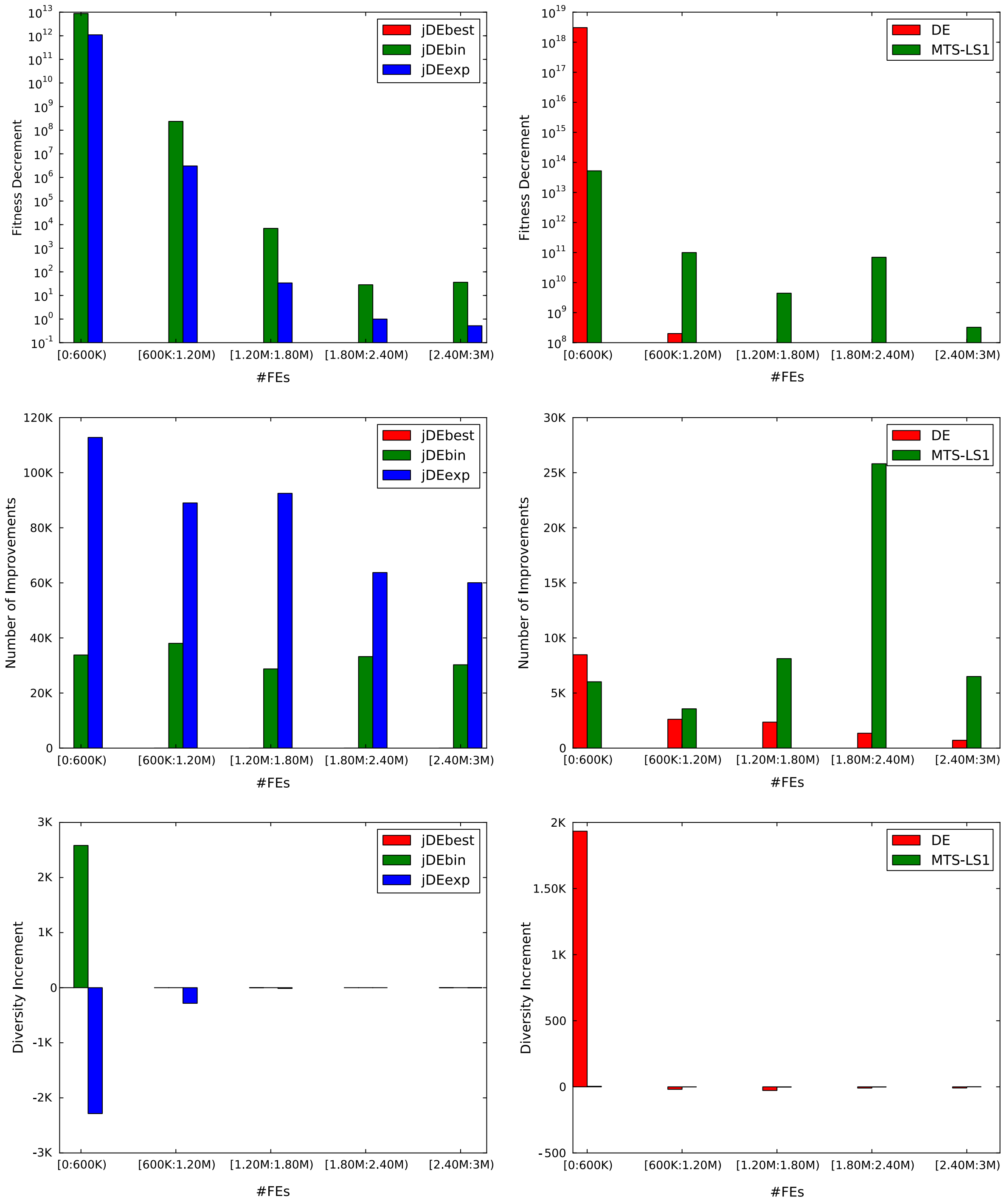

Fig. 4. Fitness increment, number of improvements and population diversity for the jDElscop and MOS-SOCO2011 algorithms on function F_{11} of the CEC2013 benchmark.

4.3.4. Per-group of functions comparison

The next analysis that has been conducted studies how the considered algorithms behave on different groups of functions. These groups of functions have been created according to the characteristics summarized in Table 17.

It should be noted that the groups defined in this way are overlapping (one function may be classified as unimodal and shifted, for example) and that complementary groups may be unbalanced, which will be analyzed later.

Fig. 6 represents the performance of each of the algorithms taking part in the comparison per group of functions. In this radar chart, again, a smaller area means better performance of the algorithm.

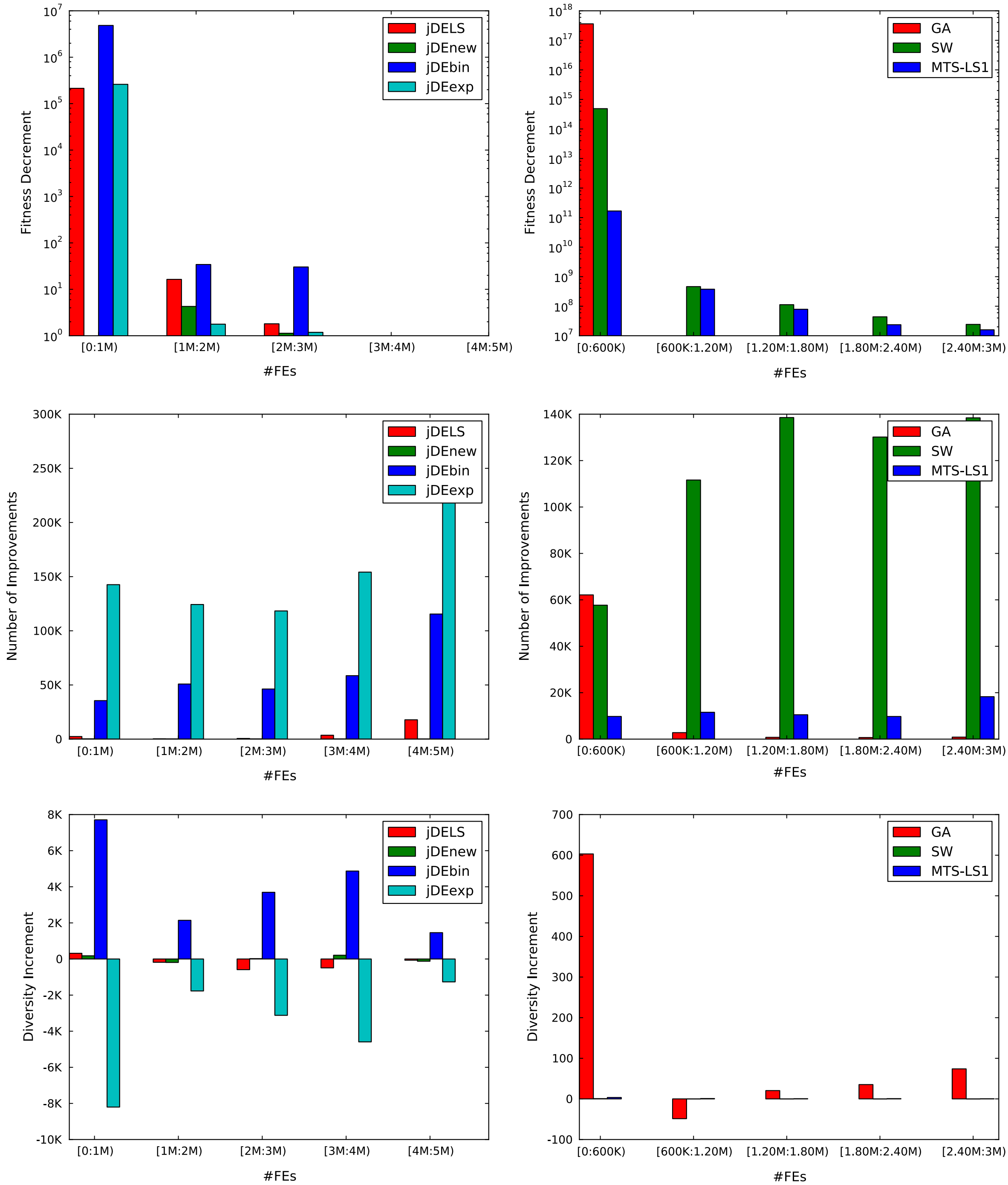

Fig. 5. Fitness increment, number of improvements and population diversity for the jDEsps and the MOS-CEC2013 algorithms on function F_{11} of the CEC2013 benchmark.

As in the previous comparison, we have divided the algorithms into two groups: algorithms without and with significant different performance (Figs. 7 and 8, respectively) to ease the comprehension of the plot. This per-group comparison confirms that the analysis conducted in the previous paragraphs also holds when focusing on different characteristics of the functions: the average ranking of the two identified pairs of algorithms (MOS-CEC2013 & jDEsps vs. MOS-SOCO2011 & jDElscop) behave in a similar manner: when MOS-CEC2013 obtains good results, jDEsps also does and vice versa.

If we pay attention to the performance of each algorithm individually, we can observe that MOS-CEC2013 algorithm has an overall good performance on all the groups of functions, being the best algorithm for the unimodal and the shifted groups of functions and having an average performance for the remaining groups, except for the non-shifted functions. This is a

Table 17
Groups of functions according to several important characteristics of the functions of the different benchmarks.

Modality	Unimodal vs. Multimodal
Separability	Fully separable vs. Partially separable vs. Non-separable
Shifting of the global optima	Shifted vs. Not shifted
Rotation	Rotated vs. Non-rotated

special case of a clearly unbalanced classification, as this group is made up of only the last 13 functions of the SOCO 2011 benchmark. This is in accordance with the general performance of this algorithm on the whole SOCO 2011 benchmark. Reviewing the results reported in Tables B.31–B33, it can be seen that the performance of this algorithm on function f_9 is poor compared to the competing algorithms. This will also explain the poor performance of this algorithm on most of the hybrid functions of the SOCO 2011 benchmark as the 75% of these functions combine f_9 with another function.

The MOS-SOCO2011 algorithm, on the other hand, seems to be a much more specialized algorithm, as it obtains the best results on the Non-shifted, Non-rotated, Fully separable and Non-separable groups, whereas it is the worst algorithm among these four algorithms in the Rotated and Partially-separable groups. This somehow makes sense due to the particular design of this algorithm, with a very small population size (only 15 individuals) and a very strong local search (the first of the local searches of the MTS algorithm [26], MTS-LS1). Fig. 4 confirms the effect of this local search in the fast optimization of functions by the intense exploitation of the search space.

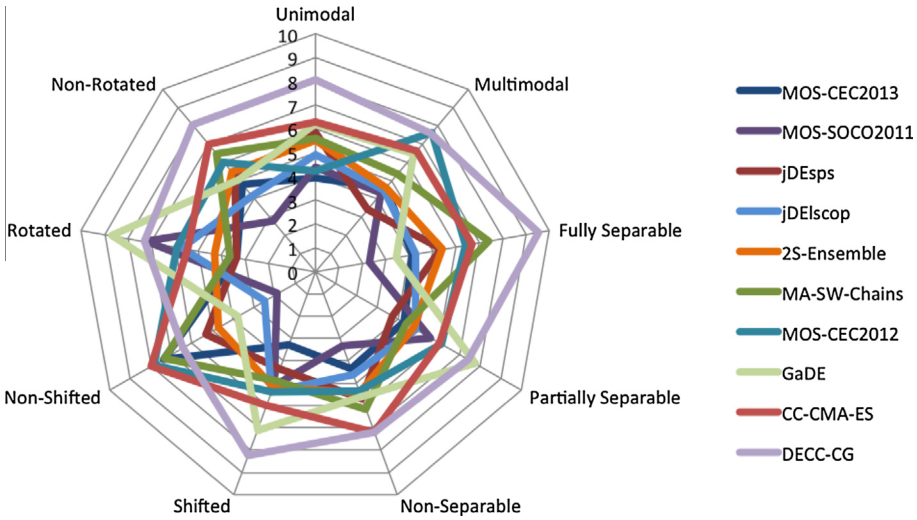

Fig. 6. Radar chart for the average rankings on the different groups of functions for all the algorithms.

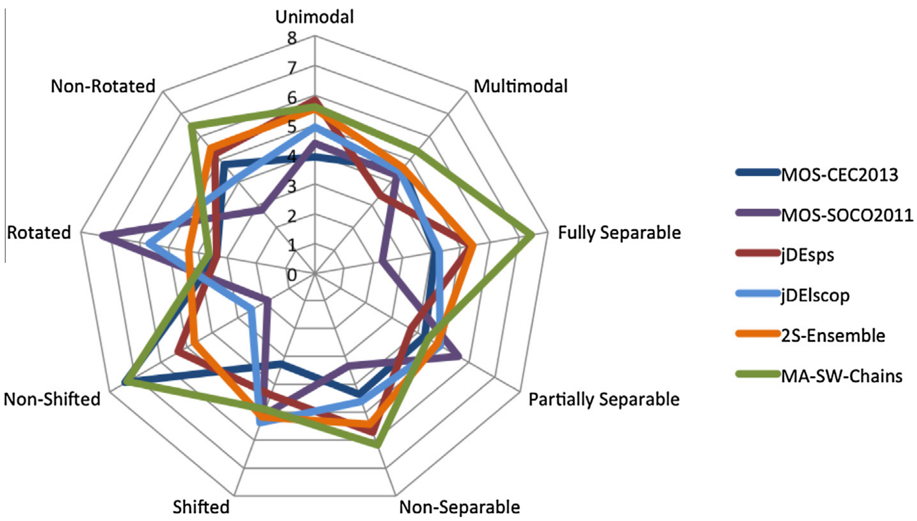

Fig. 7. Radar chart for the average rankings on the different groups of functions for the algorithms with no significant differences.

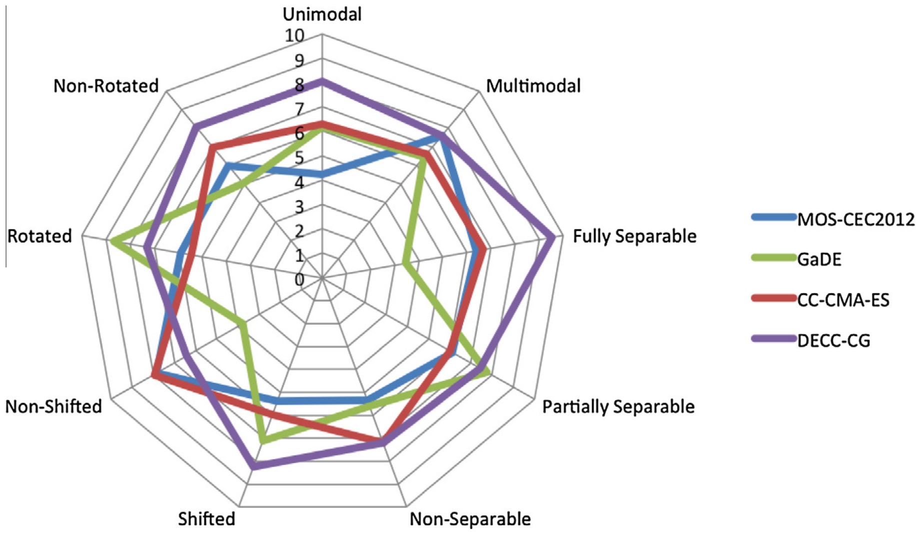

Fig. 8. Radar chart for the average rankings on the different groups of functions for the algorithms with significant differences.

Finally, it is also interesting to observe the relative performance of the jDElscop and jDEsps algorithms. The latter algorithm being an evolution of the first one, it obtains complementary results in most of the studied groups of functions: good performance of the jDElscop is normally accompanied by a not so good performance of the jDEsps and the other way around. Of course, there are some differences between both algorithms, as described in Section 2.3.2, that could explain these differences. It could be the inclusion of two new DE strategies that makes the difference or the change in the approach followed to vary the population size.

4.3.5. Performance correlation analysis

To conclude this analysis, we have measured the Pearson correlation among the mean error of all the pairs of algorithms on the 54 functions under study. Fig. 9 graphically depicts the correlation matrix with an ellipses plot. In this type of plots, the smaller an ellipse is (the more similar to a straight line), the higher the correlation is. On the other hand, the more rounded the ellipse is, the lower the correlation is. Colors represent the type of correlation, direct or inverse, for each pair of algorithms compared.

Based on this analysis, it is possible to distinguish, at least, three different behavior profiles among the algorithms: First, the group conformed by MA-SWChains, MOS-CEC2013, MOS-SOCO2011, jDElscop, MOS-CEC2012, jDEsps and GaDE show a common pattern (being the first two of them a bit different). It does not mean that they performed the same (as we have previously shown), but the error values (not ranking) are equivalent (at similar scales) for the same functions. The final performance, or the precise ability to reach (close-to) optimal values may relay on the particular adjustments or, for instance, the contribution of a local search component. Nonetheless, in general, they perform well on the same functions and stayed at the same orders of magnitude on a similar set of functions. In general this group includes DE-based and GA-based algorithms with/without dynamic population adjustments and with/without local search algorithms.

A second group is the one including CC-CMA-ES and 2S-Ensemble. These two algorithms are cooperative hybrid approaches based either on EDA techniques or other statistically based search (CMA-ES). As both algorithms are not among the best performing of the analyzed alternatives, it seems that when these approaches do not have enough information to identify the patterns of the good solutions driving the search, they fail to escape from low-quality local optima.

Finally, the only algorithm showing a performance profile different from the others is the DECC-G algorithm, an adaptive cooperative DE. This algorithm shows the worst performance in the selected benchmarks and considering the rest of the algorithms. Its results are below average in fully-separable, unimodal and non-rotated, for which many of the algorithms are able to reach the global optimum or at least being very close to it, while DECC-G stays several orders of magnitude away from these solutions.

4.4. LSGO results repository

One of the objectives of this contribution is to provide the LSGO community with a central repository in which find up-to-date information on the performance of novel algorithms on the well-known existing benchmarks. This work aims to be the initial source of information for this repository as all the results obtained during the experimentation for this paper has been uploaded to this repository and can be freely accessed and downloaded.

Keeping up-to-date this kind of repository can be a difficult task as much work is being carried out in this field. For this reason, the web site allows LSGO practitioners around the world to upload their own results to this central database. To avoid spam and fake results, all the uploads will be moderated, requiring a proof of authenticity of the results, normally in the form

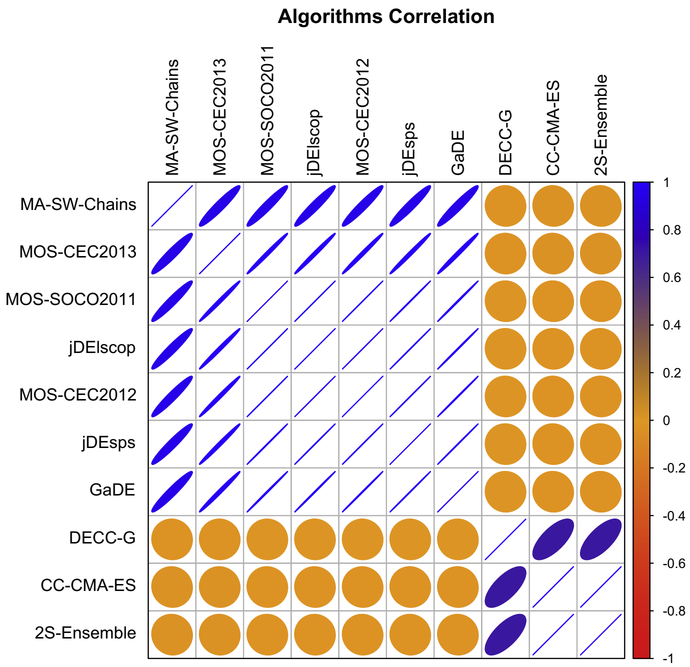

Fig. 9. Correlation of the mean results for the different algorithms.

of a peer-reviewed paper that supports the provided results or even the source code of the algorithm to locally test the program.

In its current form, the web site already allows much of the functionality that we plan to provide, including many of the comparative tools used in this paper. In particular, the site already allows checking and uploading results, computing average and F1 rankings [10,24], as well as comparisons by means of multiple statistical tools (Friedman and Wilcoxon tests, Bonferroni, Holm and Hochberg adjustments), per group of functions analysis (radar charts), etc. Other new characteristics will be included in the near future as we receive feedback from the LSGO community. We hope all these tools will ease the hard process of conducting an exhaustive and rigorous comparison prior to the publication of a new LSGO algorithm. Moreover, it will also help referees to evaluate new algorithms with the aid of this centralized automatic comparison service.[4]

5. Conclusions

In this paper we have conducted a thorough experimentation with some of the current best LSGO algorithms on a broad testbed including the CEC 2010, SOCO 2011 and CEC 2013 benchmarks. The results obtained have been analyzed from different perspectives (per benchmark, group of functions, overall, etc.) to offer a comprehensive comparison of great value for the community. Such comparison unveils the MOS-CEC2013 algorithm as the best overall performing approach, with a small advantage over the MOS-SOCO2011, jDEsps and jDElscop algorithms. A second insight shown by this study is that several algorithms that showed a great performance on a particular benchmark, do not perform as good when new functions are considered. This is remarkably true among those competing in the SOCO 2011 benchmark. Another way to interpret these results is that this particular benchmark is, indeed, composed by problems of a slightly different nature. Moreover, the analysis of equivalent performing algorithms (those solving or being very close on the same functions and performing worse in others) show an homogeneous group including those with the best performance including some common features, such as being based on the DE algorithm in most of the cases, including dynamic adaptability of population sizes or generation of offspring and the inclusion of specialized local search components. Finally, all the results have been collected and uploaded to an on-line repository to ease their use by LSGO researchers. Furthermore, the mechanisms to allow other researchers to upload their own results to the web site or use the available information in their own research may help LSGO practitioners to ease the task of comparing their own results with those of state-of-the-art algorithms.

Acknowledgments

This work was financed by the Spanish Ministry of Science (TIN2010-21289-C02-02) and supported by the Cajal Blue Brain Project. The authors thankfully acknowledge the computer resources, technical expertise and assistance provided

4 http://midas.ctb.upm.es/lab/benchmarks/.

by the Supercomputing and Visualization Center of Madrid (CeSViMa). A. LaTorre gratefully acknowledges the support of the Spanish Ministry of Science and Innovation (MICINN) for its funding throughout the Juan de la Cierva program. We also acknowledge the authors of the papers in which all the compared algorithms were originally presented for providing us with the source code of their algorithms, without which this study would have not been possible.

Appendix A. Summary of the parameters used by each algorithm

See Tables A.18–A.27.

Table A.18
Parameters values for the MA-SW-Chains algorithm.

Steady-State MA	
cr operator	BLX-α ($alpha = 0.5$)
mut operator	BGA
selector	Negative Assortative
p_{CR}	1.0
p_{mut}	0.125
n_{ass}	3
Solis and Wets	
maxSuccesses	5
maxFailures	3
adjustSuccess	2
adjustFailure	0.5
delta	0.2
MA-SW Chains	
NP	60
I_{str}	500
$r_{L/G}$	0.5
δ^{LS}_{min}	0

Table A.19
Parameters values for the 2S-Ensemble algorithm.

MUEDA	
Probabilistic model	Hybrid Gaussian and Lévy
Selector	Truncation Strategy (top 20%)
Probability-Based CC	
G_s	2
α	0.15
Meansize	30
Stdsize	10
Suboptimizers	SaDE, GA and GA-DE
EOEA	
NP	30
Phase trigger	10%

Table A.20
Parameters values for the MOS-SOCO2011 algorithm.

Differential evolution	
F	0.5
CR	0.5
CR operator	Exponential
Selector	Tournament (size 2)
Model	Classic
MTS-LS1	
(initial) SR	40% of the search space
(min) SR	1e−14
adjustFailure	2
adjustMin	0.4
moveLeft	1
moveRight	0.5
Hybrid algorithm	
minPart	5%
stepFactor	60,000

Table A.21
Parameters values for the jDElscop algorithm.

Common DE parameters	
F_{ini}	0.5
CR_{ini}	0.9
jDEbin strategy	
Scheme	DE/rand/1/bin
F interval	$[0.1 + sqrt(\frac{1.0}{NP}), 1.0]$
CR interval	$[0.0, 1.0]$
probability	NP/2 if not jDEbest
jDEexp strategy	
Scheme	DE/rand/1/exp
F interval	$[0.5, 1.0]$
CR interval	$[0.3, 1.0]$
Probability	NP/2 if not jDEbest
jDEbest strategy	
Scheme	DE/best/1/bin
F interval	$[0.4, 1.0]$
CR interval	$[0.7, 0.95]$
Probability	0.1 after MaxFEs/2
Population size reduction	
Initial NP	100
pmax (NP reduction)	4

Table A.22
Parameters values for the GaDE algorithm.

DE scheme 1	
Scheme	DE/rand/1/exp
Probability	0.8
DE scheme 2	
Scheme	DE/current-to-best/1/exp
Probability	0.2
DE parameters adaptation	
F_m	0.5
CR_m	0.9
F adaptation	Cauchy(F_m, 0.2)
CR adaptation	Gaussian(CR_m, 0.1)
Other parameters	
NP	60
Decreasing factor c	0.1

Table A.23
Parameters values for the MOS-CEC2012 algorithm.

Solis and Wets	
maxSuccesses	2
maxFailures	5
adjustSuccess	2
adjustFailure	0.75
delta	5
MTS-LS1	
(initial) SR	50% of the search space
(min) SR	$1e{-14}$
adjustFailure	5
adjustMin	2.5
moveLeft	0.5
moveRight	0.25
Hybrid algorithm	
minPart	5%
stepFactor	18,000

Table A.24
Parameters values for the jDEsps algorithm.

Common DE parameters	
F_{ini}	0.5
CR_{ini}	0.9
F_l	0.1
F_u	0.1
τ_1	0.1
τ_2	0.9
jDELS strategy	
Scheme	DE/rand/1/bin
Probability	0.1
jDEw strategy	
Scheme	DE/rand/1/bin
Probability	0.1 after MaxFEs/2
jDEbin strategy	
Scheme	DE/rand/1/bin
Probability	NP/2 if not jDELS nor jDEw
jDEexp strategy	
Scheme	DE/rand/1/exp
Probability	NP/2 if not jDELS nor jDEw
Population size reduction	
Initial NP	100
pmax (NP reduction)	3

Table A.25
Parameters values for the MOS-CEC2013 algorithm.

Genetic Algorithm	
NP	400
p_{CR}	0.9
p_{mut}	0.01
Solis and Wets	
maxSuccesses	5
maxFailures	5
adjustSuccess	4
adjustFailure	0.75
delta	2.4
MTS-LS1-Reduced	
(initial) SR	50% of the search space
(min) SR	$1e-14$
adjustFailure	2
adjustMin	10
moveLeft	0.25
moveRight	0.5
improvePerc	0.9
minPerc	0.025
Hybrid algorithm	
minPart	20%
stepFactor	36,000

Table A.26
Parameters values for the DECC-G algorithm.

Cooperative CoEvolution	
NP	100
Cycles	50
subSize	100
Internal optimizer	SaNSDE

Table A.27
Parameters values for the CC-CMA-ES algorithm.

Cooperative CoEvolution	
NP	200
historyWindow	5
subSize	50
Internal optimizer	CMA-ES

Appendix B. Full results of all the algorithms

See Tables B.28–B.36.

Table B.28
Comparison of all the algorithms on the CEC 2010 benchmark (F_1–F_7). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
MA-SW-Chains	Best	1.28E-15	7.40E+02	2.74E-13	7.49E+10	7.86E+07	9.51E-07	1.35E-03
	Median	2.67E-14	8.47E+02	5.16E-13	3.10E+11	2.30E+08	2.45E+00	7.94E-03
	Worst	2.38E-13	9.12E+02	1.76E-12	3.65E+11	2.89E+08	1.37E+06	9.80E+02
	Mean	3.80E-14	8.40E+02	5.76E-13	2.97E+11	2.18E+08	1.42E+05	1.17E+02
	Std	4.91E-14	4.88E+01	2.73E-13	6.19E+10	5.75E+07	3.96E+05	2.37E+02
2S-Ensemble	Best	0.00E+00	0.00E+00	3.13E-13	1.92E+11	1.69E+07	2.63E+06	2.77E-02
	Median	0.00E+00	1.87E-13	3.87E-13	1.13E+12	2.29E+07	3.51E+06	6.03E+00
	Worst	3.55E-24	3.98E+00	9.31E-13	3.08E+12	3.58E+07	4.47E+06	1.09E+03
	Mean	1.42E-25	4.86E-01	4.39E-13	1.24E+12	2.38E+07	3.56E+06	1.52E+02
	Std	7.10E-25	9.19E-01	1.31E-13	5.87E+11	4.99E+06	5.33E+05	2.78E+02
MOS-SOCO2011	Best	0.00E+00	0.00E+00	0.00E+00	2.36E+11	3.54E+08	1.94E+07	2.29E+06
	Median	0.00E+00	0.00E+00	0.00E+00	4.94E+11	5.00E+08	1.97E+07	2.87E+07
	Worst	0.00E+00	0.00E+00	0.00E+00	1.01E+12	6.25E+08	1.99E+07	1.39E+08
	Mean	0.00E+00	0.00E+00	0.00E+00	5.16E+11	4.93E+08	1.97E+07	3.54E+07
	Std	0.00E+00	0.00E+00	0.00E+00	1.85E+11	6.93E+07	1.15E+05	3.22E+07
jDElscop	Best	0.00E+00	0.00E+00	2.56E-13	2.85E+11	8.18E+07	7.55E-09	1.70E+03
	Median	0.00E+00	9.96E-01	2.78E-13	7.60E+11	1.34E+08	7.55E-09	7.67E+03
	Worst	0.00E+00	6.02E+00	2.99E-13	1.40E+12	1.65E+08	7.55E-09	2.97E+04
	Mean	0.00E+00	1.74E+00	2.78E-13	7.71E+11	1.30E+08	7.55E-09	1.08E+04
	Std	0.00E+00	1.81E+00	1.10E-14	3.09E+11	1.85E+07	4.11E-14	8.93E+03
GaDE	Best	0.00E+00	0.00E+00	2.17E-13	5.63E+12	2.83E+08	8.79E+06	2.11E+06
	Median	0.00E+00	1.27E-04	2.56E-13	8.29E+12	3.56E+08	2.04E+07	6.52E+06
	Worst	0.00E+00	1.20E-01	3.17E-13	1.34E+13	4.49E+08	2.06E+07	3.67E+07
	Mean	0.00E+00	1.08E-02	2.65E-13	8.81E+12	3.57E+08	1.98E+07	8.82E+06
	Std	0.00E+00	2.93E-02	2.88E-14	2.21E+12	4.90E+07	2.31E+06	7.33E+06
MOS-CEC2012	Best	0.00E+00	1.77E+02	1.89E-12	2.34E+09	4.57E+08	1.98E+07	0.00E+00
	Median	0.00E+00	1.95E+02	2.44E-12	7.27E+09	6.60E+08	1.99E+07	0.00E+00
	Worst	0.00E+00	2.60E+02	2.59E+00	1.61E+10	8.89E+08	2.00E+07	0.00E+00
	Mean	0.00E+00	1.98E+02	8.53E-01	7.80E+09	6.77E+08	1.99E+07	0.00E+00
	Std	0.00E+00	1.80E+01	9.93E-01	3.90E+09	1.23E+08	6.46E+04	0.00E+00
jDEsps	Best	0.00E+00	2.51E+01	4.66E-14	1.37E+11	6.23E+07	4.00E-09	2.32E+02
	Median	2.77E-23	7.85E+01	9.64E-14	2.51E+11	8.05E+07	4.02E-09	3.54E+04
	Worst	6.35E-21	2.41E+02	2.04E-12	1.70E+12	9.66E+07	1.47E-08	3.13E+06
	Mean	3.08E-22	9.80E+01	1.82E-13	3.80E+11	7.91E+07	5.71E-09	2.26E+05
	Std	1.26E-21	6.18E+01	3.89E-13	3.64E+11	1.14E+07	2.53E-09	6.23E+05
MOS-CEC2013	Best	0.00E+00	0.00E+00	1.50E-12	6.37E+09	6.67E+07	6.79E-08	0.00E+00
	Median	0.00E+00	0.00E+00	1.67E-12	1.63E+10	1.08E+08	9.28E-08	0.00E+00
	Worst	0.00E+00	0.00E+00	1.77E-12	2.94E+10	1.70E+08	2.92E-07	0.00E+00
	Mean	0.00E+00	0.00E+00	1.65E-12	1.70E+10	1.07E+08	1.11E-07	0.00E+00
	Std	0.00E+00	0.00E+00	6.44E-14	6.39E+09	2.49E+07	5.88E-08	0.00E+00
DECC-G	Best	1.55E-07	1.28E+03	2.65E-04	9.65E+12	1.71E+08	3.03E+06	1.59E+08
	Median	2.81E-07	1.31E+03	1.16E+00	2.40E+13	2.93E+08	4.52E+06	6.99E+08
	Worst	5.63E-07	1.43E+03	1.39E+00	5.48E+13	5.16E+08	6.34E+06	1.96E+09
	Mean	3.07E-07	1.32E+03	1.11E+00	2.58E+13	3.02E+08	4.53E+06	7.60E+08
	Std	1.02E-07	4.13E+01	2.68E-01	1.13E+13	7.41E+07	7.97E+05	4.57E+08
CC-CMA-ES	Best	1.36E-10	1.17E+03	1.35E-13	4.14E+11	6.45E+07	6.08E-05	2.53E-07
	Median	5.14E-09	1.41E+03	1.46E-13	9.82E+11	1.32E+08	1.98E+01	7.46E-04
	Worst	7.64E-09	2.20E+03	1.56E-13	2.63E+12	3.99E+08	4.62E+06	1.88E-01
	Mean	4.84E-09	1.46E+03	1.47E-13	1.08E+12	1.78E+08	1.03E+06	1.60E-02
	Std	1.80E-09	2.43E+02	5.32E-15	5.37E+11	9.92E+07	1.42E+06	4.41E-02

Table B.29
Comparison of all the algorithms on the CEC 2010 benchmark (F_8–F_{14}). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
MA-SW-Chains	Best	2.26E+05	1.25E+07	1.63E+03	2.35E+01	2.53E-06	3.18E+02	2.91E+07
	Median	2.76E+06	1.48E+07	2.02E+03	3.77E+01	3.09E-06	8.61E+02	3.23E+07
	Worst	9.78E+07	1.93E+07	2.33E+03	6.05E+01	4.56E-06	2.70E+03	3.73E+07
	Mean	6.90E+06	1.49E+07	2.01E+03	3.86E+01	3.24E-06	9.83E+02	3.25E+07
	Std	1.90E+07	1.61E+06	1.59E+02	8.06E+00	5.78E-07	5.66E+02	2.46E+06
2S-Ensemble	Best	5.07E+02	3.47E+07	9.02E+02	2.46E+01	2.91E+03	6.46E+02	1.34E+08
	Median	6.39E+06	4.18E+07	1.02E+03	2.84E+01	5.44E+03	1.01E+03	1.46E+08
	Worst	6.08E+07	4.60E+07	1.18E+03	4.91E+01	7.39E+03	2.14E+03	1.62E+08
	Mean	1.02E+07	4.07E+07	1.03E+03	3.08E+01	5.28E+03	1.09E+03	1.46E+08
	Std	1.28E+07	3.28E+06	6.82E+01	6.07E+00	1.08E+03	3.60E+02	8.43E+06
MOS-SOCO2011	Best	5.17E+00	7.82E+06	5.55E+03	1.98E+02	1.30E+03	1.52E+02	1.33E+07
	Median	2.14E+06	1.18E+07	6.35E+03	1.99E+02	3.46E+03	3.19E+02	2.04E+07
	Worst	1.42E+07	1.38E+07	6.86E+03	1.99E+02	1.35E+04	6.01E+02	2.62E+07
	Mean	3.75E+06	1.13E+07	6.28E+03	1.99E+02	4.39E+03	3.32E+02	2.05E+07
	Std	4.40E+06	1.61E+06	3.12E+02	3.06E-01	2.92E+03	1.19E+02	3.60E+06
jDElscop	Best	4.04E-03	6.09E+07	2.66E+03	1.88E-08	4.03E+04	6.99E+02	2.11E+08
	Median	7.12E+06	7.71E+07	3.38E+03	4.67E+01	5.51E+04	1.01E+03	2.75E+08
	Worst	1.40E+07	8.99E+07	4.48E+03	1.27E+02	7.45E+04	1.20E+03	3.41E+08
	Mean	5.84E+06	7.60E+07	3.42E+03	5.51E+01	5.46E+04	1.01E+03	2.79E+08
	Std	4.75E+06	8.42E+06	4.79E+02	3.05E+01	8.74E+03	1.28E+02	3.32E+07
GaDE	Best	6.70E+06	8.97E+08	6.24E+03	1.99E+02	6.49E+05	8.61E+02	2.33E+09
	Median	4.50E+07	1.00E+09	6.45E+03	2.08E+02	7.48E+05	1.15E+03	2.51E+09
	Worst	9.24E+07	1.11E+09	6.66E+03	2.09E+02	8.08E+05	2.95E+03	2.63E+09
	Mean	5.09E+07	1.01E+09	6.47E+03	2.07E+02	7.43E+05	1.23E+03	2.49E+09
	Std	2.56E+07	5.71E+07	1.17E+02	3.04E+00	4.80E+04	4.10E+02	8.74E+07
MOS-CEC2012	Best	9.35E-03	4.02E+06	6.76E+03	1.98E+02	0.00E+00	9.39E+01	9.19E+06
	Median	2.57E-01	5.06E+06	7.79E+03	1.99E+02	0.00E+00	7.23E+02	1.10E+07
	Worst	3.99E+06	6.08E+06	1.08E+04	2.01E+02	0.00E+00	2.87E+03	1.25E+07
	Mean	7.97E+05	5.12E+06	7.88E+03	1.99E+02	0.00E+00	9.35E+02	1.10E+07
	Std	1.63E+06	5.02E+05	7.43E+02	6.64E-01	0.00E+00	7.40E+02	8.78E+05
jDEsps	Best	7.05E-23	1.23E+04	1.86E+03	5.01E-07	4.72E+02	2.00E+01	3.79E+04
	Median	4.49E-16	3.13E+05	1.86E+03	1.56E+00	5.41E+03	2.53E+02	1.21E+05
	Worst	1.62E+07	4.21E+06	2.50E+03	1.56E+00	1.96E+05	8.99E+02	6.70E+05
	Mean	6.50E+05	5.37E+05	1.89E+03	1.50E+00	1.59E+04	3.13E+02	1.84E+05
	Std	3.25E+06	8.54E+05	1.28E+02	3.12E-01	3.96E+04	2.21E+02	1.68E+05
MOS-CEC2013	Best	1.51E-09	2.70E+06	3.52E+03	1.90E+02	0.00E+00	5.04E+00	8.36E+06
	Median	1.38E-07	3.47E+06	3.82E+03	1.91E+02	0.00E+00	5.69E+02	9.77E+06
	Worst	3.51E+01	4.59E+06	4.03E+03	1.92E+02	0.00E+00	2.50E+03	1.09E+07
	Mean	1.40E+00	3.59E+06	3.81E+03	1.91E+02	0.00E+00	8.23E+02	9.69E+06
	Std	7.01E+00	4.89E+05	1.62E+02	4.01E-01	0.00E+00	6.77E+02	6.71E+05
DECC-G	Best	1.70E+07	3.51E+08	9.70E+03	2.21E+01	8.33E+04	1.66E+03	8.45E+08
	Median	8.90E+07	4.24E+08	1.02E+04	2.57E+01	9.66E+04	3.37E+03	9.85E+08
	Worst	9.31E+07	5.96E+08	1.10E+04	2.85E+01	1.16E+05	3.23E+04	1.14E+09
	Mean	8.01E+07	4.39E+08	1.03E+04	2.56E+01	9.79E+04	6.51E+03	9.88E+08
	Std	2.14E+07	6.20E+07	2.75E+02	1.55E+00	8.27E+03	7.34E+03	6.16E+07
CC-CMA-ES	Best	3.92E+03	2.03E+07	3.13E+03	5.05E+01	2.65E-01	6.01E+02	5.84E+07
	Median	4.83E+07	2.82E+07	3.47E+03	8.81E+01	4.97E-01	1.48E+03	6.68E+07
	Worst	3.88E+08	3.21E+07	3.95E+03	2.34E+02	1.25E+00	5.73E+03	7.74E+07
	Mean	6.93E+07	2.75E+07	3.54E+03	1.07E+02	5.33E-01	1.83E+03	6.70E+07
	Std	8.28E+07	2.86E+06	2.49E+02	5.22E+01	2.29E-01	1.16E+03	5.11E+06

Table B.30
Comparison of all the algorithms on the CEC 2010 benchmark (F_{15}–F_{20}). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
MA-SW-Chains	Best	2.53E+03	8.10E+01	9.72E-01	7.88E+02	2.70E+05	9.49E+02
	Median	2.67E+03	9.32E+01	1.28E+00	1.41E+03	2.95E+05	1.04E+03
	Worst	2.91E+03	1.26E+02	1.53E+00	3.05E+03	3.20E+05	1.34E+03
	Mean	2.68E+03	9.95E+01	1.27E+00	1.57E+03	2.95E+05	1.06E+03
	Std	9.95E+01	1.53E+01	1.24E-01	6.73E+02	1.34E+04	9.38E+01
2S-Ensemble	Best	1.94E+03	6.82E+01	3.33E+04	1.47E+03	8.88E+05	1.31E+03
	Median	2.17E+03	8.03E+01	3.88E+04	3.05E+03	9.95E+05	1.74E+03
	Worst	2.33E+03	9.83E+01	6.17E+04	4.93E+03	1.16E+06	2.13E+03
	Mean	2.15E+03	8.13E+01	4.22E+04	2.96E+03	9.98E+05	1.76E+03
	Std	9.96E+01	9.15E+00	8.72E+03	7.76E+02	6.72E+04	2.19E+02
MOS-SOCO2011	Best	1.23E+04	3.83E+02	3.26E+03	4.95E+02	3.76E+05	5.86E-05
	Median	1.29E+04	3.97E+02	7.30E+03	7.78E+02	5.71E+05	7.40E+01
	Worst	1.34E+04	3.98E+02	2.74E+04	2.30E+03	6.79E+05	3.87E+02
	Mean	1.29E+04	3.96E+02	8.45E+03	8.96E+02	5.49E+05	9.23E+01
	Std	3.48E+02	3.47E+00	5.04E+03	4.03E+02	8.38E+04	8.99E+01
jDElscop	Best	5.08E+03	1.10E+02	2.45E+05	2.45E+03	4.10E+05	9.12E+02
	Median	7.31E+03	2.04E+02	2.74E+05	3.31E+03	5.54E+05	9.69E+02
	Worst	9.21E+03	3.29E+02	3.14E+05	4.00E+03	6.50E+05	9.80E+02
	Mean	7.37E+03	1.97E+02	2.75E+05	3.23E+03	5.36E+05	9.47E+02
	Std	8.21E+02	5.24E+01	1.84E+04	4.00E+02	6.62E+04	2.93E+01
GaDE	Best	1.21E+04	4.07E+02	1.86E+06	4.60E+03	2.61E+05	9.68E+02
	Median	1.35E+04	4.08E+02	1.93E+06	1.23E+04	3.17E+05	1.06E+03
	Worst	1.38E+04	4.09E+02	2.01E+06	2.30E+04	3.56E+05	1.15E+03
	Mean	1.34E+04	4.08E+02	1.93E+06	1.33E+04	3.16E+05	1.06E+03
	Std	4.61E+02	5.26E-01	4.52E+04	5.77E+03	2.27E+04	5.21E+01
MOS-CEC2012	Best	1.42E+04	3.96E+02	4.58E-12	4.66E+02	3.49E+03	3.10E-02
	Median	1.54E+04	3.97E+02	7.35E-12	2.01E+03	3.89E+03	1.63E+02
	Worst	1.64E+04	3.98E+02	8.93E-12	6.78E+03	4.56E+03	1.22E+03
	Mean	1.54E+04	3.97E+02	7.19E-12	2.35E+03	3.91E+03	2.43E+02
	Std	5.51E+02	3.81E-01	1.15E-12	1.21E+03	2.96E+02	3.04E+02
jDEsps	Best	4.55E+03	1.16E+00	2.10E+03	5.78E+02	1.07E+05	2.08E+02
	Median	5.23E+03	1.16E+00	1.97E+04	1.52E+03	9.03E+05	9.92E+02
	Worst	6.09E+03	1.16E+00	6.19E+05	2.17E+03	1.49E+06	1.73E+03
	Mean	5.27E+03	1.16E+00	6.85E+04	1.37E+03	9.56E+05	8.95E+02
	Std	4.45E+02	1.02E-06	1.36E+05	5.00E+02	2.97E+05	4.25E+02
MOS-CEC2013	Best	7.14E+03	3.27E+02	1.71E-07	5.08E+02	2.48E+04	1.05E-02
	Median	7.45E+03	3.87E+02	2.67E-07	1.55E+03	2.87E+04	1.81E+02
	Worst	7.85E+03	3.88E+02	4.89E-07	4.25E+03	3.47E+04	1.17E+03
	Mean	7.44E+03	3.79E+02	2.73E-07	1.77E+03	2.92E+04	2.93E+02
	Std	1.90E+02	1.83E+01	7.67E-08	9.57E+02	2.29E+03	3.99E+02
DECC-G	Best	1.13E+04	4.70E+01	2.71E+05	7.27E+03	1.00E+06	3.98E+03
	Median	1.21E+04	6.62E+01	3.18E+05	3.98E+04	1.11E+06	4.28E+03
	Worst	1.33E+04	8.67E+01	3.48E+05	6.00E+04	1.27E+06	8.00E+03
	Mean	1.22E+04	6.61E+01	3.14E+05	3.69E+04	1.12E+06	4.82E+03
	Std	5.96E+02	8.85E+00	1.70E+04	1.36E+04	6.85E+04	1.30E+03
CC-CMA-ES	Best	3.95E+03	1.44E+02	3.12E+01	3.34E+03	8.33E+04	9.79E+02
	Median	4.94E+03	2.29E+02	4.12E+01	7.86E+03	5.45E+06	9.80E+02
	Worst	5.95E+03	4.27E+02	6.05E+01	1.62E+04	6.73E+06	1.07E+03
	Mean	4.99E+03	2.33E+02	4.19E+01	8.29E+03	5.26E+06	9.86E+02
	Std	4.98E+02	7.02E+01	7.12E+00	3.79E+03	1.26E+06	2.10E+01

Table B.31
Comparison of all the algorithms on the SOCO 2011 benchmark (F_1–F_7). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_1	F_2	F_3	F_4	F_5	F_6	F_7
MA-SW-Chains	Best	2.50E-16	9.34E+01	8.94E+02	1.49E+02	1.39E-16	6.54E-13	2.08E-13
	Median	2.78E-16	9.65E+01	9.56E+02	1.89E+02	1.53E-16	7.11E-13	2.51E-13
	Worst	3.05E-16	9.91E+01	1.07E+03	2.30E+02	7.40E-03	7.39E-13	4.14E-13
	Mean	2.75E-16	9.64E+01	9.72E+02	1.90E+02	2.96E-04	7.07E-13	2.52E-13
	Std	1.59E-17	1.74E+00	4.78E+01	2.15E+01	1.48E-03	2.99E-14	3.83E-14
2S-Ensemble	Best	0.00E+00	3.60E+01	7.51E+02	0.00E+00	0.00E+00	0.00E+00	Inf
	Median	0.00E+00	4.26E+01	1.33E+03	9.95E-01	0.00E+00	0.00E+00	Inf
	Worst	0.00E+00	5.63E+01	1.74E+03	3.98E+00	1.97E-02	0.00E+00	Inf
	Mean	0.00E+00	4.31E+01	1.34E+03	8.58E-01	3.00E-03	0.00E+00	Inf
	Std	0.00E+00	4.17E+00	2.25E+02	9.48E-01	1.80E+02	0.00E+00	NaN
MOS-SOCO2011	Best	0.00E+00	6.48E-02	3.56E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Median	0.00E+00	1.24E-01	6.77E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Worst	0.00E+00	1.14E+01	2.88E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	0.00E+00	5.88E-01	7.09E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	4.52E+02	8.11E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
jDElscop	Best	0.00E+00	2.07E+01	8.35E+02	0.00E+00	0.00E+00	1.11E-12	0.00E+00
	Median	0.00E+00	2.33E+01	8.44E+02	0.00E+00	0.00E+00	1.17E-12	0.00E+00
	Worst	0.00E+00	3.25E+01	9.09E+02	1.99E+00	0.00E+00	1.19E-12	0.00E+00
	Mean	0.00E+00	2.46E+01	8.51E+02	2.39E-01	0.00E+00	1.16E-12	0.00E+00
	Std	0.00E+00	2.86E+00	1.91E+01	5.20E-01	0.00E+00	2.50E-14	0.00E+00
GaDE	Best	0.00E+00	2.03E+01	8.68E+02	0.00E+00	0.00E+00	1.47E-14	0.00E+00
	Median	0.00E+00	5.41E+01	9.48E+02	0.00E+00	0.00E+00	1.82E-14	0.00E+00
	Worst	0.00E+00	7.76E+01	1.03E+03	7.85E-01	1.48E-02	1.21E-13	0.00E+00
	Mean	0.00E+00	5.46E+01	9.47E+02	3.79E-02	5.91E-04	2.55E-14	0.00E+00
	Std	0.00E+00	1.35E+01	3.65E+01	1.57E-01	2.95E-03	2.46E-14	0.00E+00
MOS-CEC2012	Best	0.00E+00	8.39E+01	7.80E+02	1.64E+02	6.00E-12	1.70E-11	8.00E-14
	Median	0.00E+00	1.05E+02	9.04E+02	1.89E+02	6.99E-12	1.22E+00	1.10E-13
	Worst	0.00E+00	1.24E+02	1.51E+03	2.26E+02	1.48E-02	2.45E+00	1.20E-13
	Mean	0.00E+00	1.04E+02	9.38E+02	1.90E+02	1.18E-03	1.03E+00	1.04E-13
	Std	0.00E+00	4.61E+02	5.46E+02	3.43E+02	1.80E+02	1.41E+02	1.04E-14
jDEsps	Best	5.55E-17	5.51E+00	8.33E-17	3.14E+01	2.78E-17	2.56E-13	0.00E+00
	Median	1.11E-16	1.31E+01	2.71E+02	8.04E+01	6.94E-17	3.69E-13	0.00E+00
	Worst	3.05E-16	2.65E+01	1.28E+03	3.06E+02	6.85E-02	7.96E-13	3.39E-15
	Mean	1.40E-16	1.37E+01	4.34E+02	1.07E+02	2.74E-03	4.34E-13	2.80E-16
	Std	6.77E-17	5.54E+00	4.93E+02	7.64E+01	1.37E-02	1.72E-13	8.01E-16
MOS-CEC2013	Best	0.00E+00	1.03E+02	2.62E-04	0.00E+00	0.00E+00	0.00E+00	2.46E-12
	Median	0.00E+00	1.10E+02	3.99E+00	0.00E+00	0.00E+00	0.00E+00	2.55E-12
	Worst	0.00E+00	1.13E+02	1.05E+02	0.00E+00	0.00E+00	0.00E+00	2.71E-12
	Mean	0.00E+00	1.10E+02	7.39E+00	0.00E+00	0.00E+00	0.00E+00	2.56E-12
	Std	0.00E+00	4.53E+02	2.07E+01	0.00E+00	0.00E+00	0.00E+00	6.86E-14
DECC-G	Best	8.47E-07	1.24E+03	1.05E-04	8.22E+10	5.73E+06	0.00E+00	Inf
	Median	2.34E-06	1.31E+03	1.19E+00	2.08E+11	8.19E+06	1.14E+00	Inf
	Worst	1.26E-05	1.38E+03	1.42E+00	4.13E+11	1.24E+07	1.41E+00	Inf
	Mean	3.26E-06	1.31E+03	1.09E+00	2.16E+11	8.30E+06	9.63E-01	Inf
	Std	2.93E-06	3.42E+01	3.54E-01	7.78E+10	1.60E+06	4.47E-01	Inf
CC-CMA-ES	Best	0.00e+00	8.74E+00	9.70E+02	1.11E+03	0.00E+00	0.00e+00	Inf
	Median	0.00e+00	1.21E+01	9.74E+02	1.52E+03	0.00E+00	0.00e+00	Inf
	Worst	0.00e+00	1.40E+01	9.80E+02	1.81E+03	1.97E-02	0.00e+00	Inf
	Mean	0.00e+00	1.20E+01	9.74E+02	1.47E+03	2.30E-03	0.00e+00	Inf
	Std	0.00e+00	1.11E+00	1.66E+00	1.78E+02	5.64E-03	0.00e+00	Inf

Table B.32
Comparison of all the algorithms on the SOCO 2011 benchmark (F_8–F_{13}). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
MA-SW-Chains	Best	4.88E+03	2.17E+03	0.00E+00	2.13E+03	8.00E+02	9.99E+02
	Median	5.82E+03	2.50E+03	0.00E+00	2.55E+03	1.01E+03	1.18E+03
	Worst	9.30E+03	3.10E+03	1.22E+01	3.15E+03	1.21E+03	1.93E+03
	Mean	5.98E+03	2.51E+03	1.58E+00	2.54E+03	9.99E+02	1.23E+03
	Std	8.40E+02	1.85E+02	3.57E+00	1.87E+02	1.02E+02	2.17E+02
2S-Ensemble	Best	1.68E+05	4.14E-01	0.00E+00	7.79E-01	0.00E+00	1.09E+03
	Median	1.89E+05	2.46E+00	0.00E+00	2.59E+00	2.52E-29	1.25E+03
	Worst	2.27E+05	6.30E+00	8.22E-15	1.04E+01	8.08E-28	1.45E+03
	Mean	1.93E+05	2.68E+00	5.91E-16	3.23E+00	1.46E-28	1.25E+03
	Std	1.33E+04	1.68E+00	1.76E-15	2.21E+00	2.20E-28	9.96E+01
MOS-SOCO2011	Best	1.17E+05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.23E-01
	Median	1.59E+05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.44E+02
	Worst	2.43E+05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.60E+02
	Mean	1.66E+05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.69E+02
	Std	3.35E+04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.55E+02
jDElscop	Best	2.32E+04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.35E+02
	Median	3.14E+04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.81E+02
	Worst	5.69E+04	5.09E-07	0.00E+00	2.89E-07	0.00E+00	7.06E+02
	Mean	3.17E+04	9.21E-08	0.00E+00	4.98E-08	0.00E+00	6.67E+02
	Std	6.67E+03	1.43E-07	0.00E+00	7.66E-08	0.00E+00	2.36E+01
GaDE	Best	1.24E+04	0.00E+00	0.00E+00	0.00E+00	1.60E-12	6.96E+02
	Median	1.56E+04	0.00E+00	0.00E+00	0.00E+00	2.78E-12	7.11E+02
	Worst	1.85E+04	1.07E-02	2.10E+00	1.07E-02	5.29E-12	7.64E+02
	Mean	1.55E+04	4.29E-04	3.36E-01	8.58E-04	2.90E-12	7.19E+02
	Std	1.60E+03	2.15E-03	5.84E-01	2.97E-03	8.97E-13	1.89E+01
MOS-CEC2012	Best	9.94E+02	5.61E+03	1.53E+02	5.46E+03	9.58E+02	1.47E+03
	Median	1.08E+03	5.96E+03	1.76E+02	5.91E+03	1.11E+03	1.97E+03
	Worst	1.18E+03	6.31E+03	2.19E+02	6.24E+03	1.37E+03	3.22E+03
	Mean	1.09E+03	5.96E+03	1.79E+02	5.88E+03	1.12E+03	2.04E+03
	Std	5.02E+01	1.97E+02	1.68E+01	2.03E+02	9.75E+01	4.27E+02
jDEsps	Best	1.13E+05	0.00E+00	0.00E+00	0.00E+00	8.33E-17	1.10E+02
	Median	2.73E+05	0.00E+00	4.20E+00	0.00E+00	2.54E-16	6.78E+02
	Worst	4.62E+05	1.22E+02	3.25E+01	1.96E+00	4.72E+02	1.25E+03
	Mean	2.72E+05	9.50E+00	7.89E+00	1.54E-01	5.72E+01	6.28E+02
	Std	1.19E+05	2.44E+01	7.73E+00	5.30E-01	1.22E+02	3.34E+02
MOS-CEC2013	Best	4.88E+03	2.17E+03	0.00E+00	2.13E+03	8.00E+02	9.99E+02
	Median	5.82E+03	2.50E+03	0.00E+00	2.55E+03	1.01E+03	1.18E+03
	Worst	9.30E+03	3.10E+03	1.22E+01	3.15E+03	1.21E+03	1.93E+03
	Mean	5.98E+03	2.51E+03	1.58E+00	2.54E+03	9.99E+02	1.23E+03
	Std	8.40E+02	1.85E+02	3.57E+00	1.87E+02	1.02E+02	2.17E+02
DECC-G	Best	1.01E+05	1.62E+01	1.62E+02	1.62E+01	0.00E+00	2.57E+03
	Median	1.10E+05	1.76E+01	1.93E+02	1.75E+01	0.00E+00	3.41E+03
	Worst	1.23E+05	1.94E+01	2.22E+02	1.90E+01	0.00E+00	1.50E+04
	Mean	1.11E+05	1.78E+01	1.94E+02	1.76E+01	0.00E+00	3.86E+03
	Std	6.14E+03	7.52E-01	1.48E+01	7.77E-01	0.00E+00	2.36E+03
CC-CMA-ES	Best	1.69E+06	6.33E+02	2.88E+01	7.57E+02	3.88E-24	2.75E+03
	Median	1.87E+06	9.18E+02	4.15E+01	8.98E+02	1.16E-02	2.83E+03
	Worst	2.18E+06	1.05E+03	6.24E+01	1.08E+03	9.11E-01	3.78E+03
	Mean	1.89E+06	8.86E+02	4.31E+01	8.90E+02	7.90E-02	2.91E+03
	Std	1.40E+05	1.21E+02	9.62E+00	9.17E+01	1.84E-01	2.25E+02

Table B.33
Comparison of all the algorithms on the SOCO 2011 benchmark (F_{14}–F_{19}). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}
MA-SW-Chains	Best	2.96E+03	1.76E-12	8.02E+03	3.55E+11	1.77E+03	1.67E+03
	Median	3.36E+03	1.92E-12	8.02E+03	3.55E+11	2.03E+03	2.03E+03
	Worst	3.83E+03	2.07E-12	8.02E+03	3.55E+11	2.30E+03	2.55E+03
	Mean	3.37E+03	1.93E-12	8.02E+03	3.55E+11	2.03E+03	2.05E+03
	Std	1.69E+02	7.52E-14	7.42E-12	1.52E+05	1.53E+02	1.84E+02
2S-Ensemble	Best	7.11E-15	Inf	0.00E+00	9.34E-01	1.38E-01	0.00E+00
	Median	3.22E-02	Inf	0.00E+00	1.02E+01	4.73E-01	0.00E+00
	Worst	1.68E-01	Inf	0.00E+00	1.42E+02	1.21E+00	1.55E-15
	Mean	4.40E-02	Inf	0.00E+00	3.39E+01	5.51E-01	7.99E-17
	Std	4.25E-02	NaN	0.00E+00	4.23E+01	2.54E-01	3.13E-16
MOS-SOCO2011	Best	0.00E+00	0.00E+00	0.00E+00	1.55E-01	0.00E+00	0.00E+00
	Median	0.00E+00	0.00E+00	0.00E+00	6.60E+01	0.00E+00	0.00E+00
	Worst	0.00E+00	0.00E+00	0.00E+00	2.66E+02	0.00E+00	0.00E+00
	Mean	0.00E+00	0.00E+00	0.00E+00	6.71E+01	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	7.49E+01	0.00E+00	0.00E+00
jDElscop	Best	0.00E+00	0.00E+00	0.00E+00	1.56E+02	1.49E-12	0.00E+00
	Median	0.00E+00	0.00E+00	0.00E+00	1.65E+02	2.70E-12	0.00E+00
	Worst	2.00E+00	0.00E+00	0.00E+00	2.20E+02	8.30E-12	0.00E+00
	Mean	4.03E-01	0.00E+00	0.00E+00	1.71E+02	3.28E-12	0.00E+00
	Std	6.48E-01	0.00E+00	0.00E+00	1.89E+01	1.91E-12	0.00E+00
GaDE	Best	4.07E-11	0.00E+00	1.21E-12	2.13E+02	8.25E-08	0.00E+00
	Median	7.18E-11	0.00E+00	1.69E-12	2.19E+02	1.24E-07	0.00E+00
	Worst	1.26E-01	1.05E+00	2.36E-12	2.25E+02	2.29E-07	2.10E+00
	Mean	7.72E-03	8.40E-02	1.67E-12	2.18E+02	1.31E-07	2.10E-01
	Std	2.49E-02	2.91E-01	3.14E-13	3.14E+00	2.99E-08	5.25E-01
MOS-CEC2012	Best	4.07E+03	1.15E+01	2.15E+03	3.45E+03	2.01E+03	4.04E+01
	Median	4.31E+03	1.99E+01	2.32E+03	3.69E+03	2.29E+03	5.25E+01
	Worst	4.72E+03	3.04E+01	2.51E+03	4.04E+03	2.55E+03	6.30E+01
	Mean	4.32E+03	2.04E+01	2.33E+03	3.71E+03	2.29E+03	5.25E+01
	Std	1.67E+02	5.08E+00	8.86E+01	1.29E+02	1.24E+02	6.42E+00
jDEsps	Best	1.69E+01	0.00E+00	1.73E-16	5.77E-14	9.95E-01	0.00E+00
	Median	3.42E+01	0.00E+00	3.09E+01	1.03E+00	1.99E+00	2.10E+00
	Worst	1.48E+02	8.40E+00	6.84E+02	7.46E+02	2.62E+01	2.20E+01
	Mean	4.66E+01	1.81E+00	1.05E+02	9.01E+01	4.03E+00	3.53E+00
	Std	3.35E+01	2.62E+00	1.63E+02	1.76E+02	5.84E+00	5.27E+00
MOS-CEC2013	Best	2.96E+03	1.76E-12	8.02E+03	3.55E+11	1.77E+03	1.67E+03
	Median	3.36E+03	1.92E-12	8.02E+03	3.55E+11	2.03E+03	2.03E+03
	Worst	3.83E+03	2.07E-12	8.02E+03	3.55E+11	2.30E+03	2.55E+03
	Mean	3.37E+03	1.93E-12	8.02E+03	3.55E+11	2.03E+03	2.05E+03
	Std	1.69E+02	7.52E-14	7.42E-12	1.52E+05	1.53E+02	1.84E+02
DECC-G	Best	1.39E+02	5.25E+00	0.00E+00	2.29E+01	7.40E+00	6.51E+01
	Median	1.61E+02	1.89E+01	0.00E+00	1.91E+02	8.39E+00	1.15E+02
	Worst	1.82E+02	3.25E+01	0.00E+00	3.84E+02	9.42E+00	1.44E+02
	Mean	1.59E+02	1.84E+01	0.00E+00	1.98E+02	8.43E+00	1.12E+02
	Std	1.41E+01	6.89E+00	0.00E+00	8.41E+01	5.56E-01	1.63E+01
CC-CMA-ES	Best	1.22E+03	Inf	1.96E-25	4.04E+03	8.17E+02	4.14E-01
	Median	1.35E+03	Inf	1.58E-14	5.07E+03	1.29E+03	1.26E+01
	Worst	1.80E+03	Inf	1.47E-03	6.03E+03	1.56E+03	2.21E+01
	Mean	1.38E+03	Inf	1.67E-04	5.00E+03	1.29E+03	1.18E+01
	Std	1.31E+02	Inf	4.07E-04	6.17E+02	1.47E+02	5.42E+00

Table B.34
Comparison of all the algorithms on the CEC 2013 benchmark (F_1–F_5). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_1	F_2	F_3	F_4	F_5
MA-SW-Chains	Best	3.03E-14	9.52E+02	2.63E-13	7.20E+08	1.07E+06
	Median	6.15E-13	1.13E+03	6.79E-13	2.70E+09	1.98E+06
	Worst	4.97E-12	1.50E+03	1.24E-12	9.06E+09	7.59E+06
	Mean	1.14E-12	1.18E+03	6.78E-13	3.80E+09	2.26E+06
	Std	1.28E-12	1.84E+02	2.28E-13	2.70E+09	1.36E+06
2S-Ensemble	Best	0.00E+00	1.04E-26	2.81E-13	4.18E+09	5.89E+05
	Median	0.00E+00	9.95E-01	3.98E-13	1.13E+10	1.28E+06
	Worst	4.55E-23	5.97E+00	5.72E-13	1.68E+10	1.79E+06
	Mean	2.35E-24	1.39E+00	4.20E-13	1.10E+10	1.33E+06
	Std	9.16E-24	1.62E+00	7.82E-14	3.44E+09	2.50E+05
MOS-SOCO2011	Best	0.00E+00	1.10E+01	0.00E+00	3.14E+09	8.51E+06
	Median	0.00E+00	1.96E+01	0.00E+00	1.20E+10	1.09E+07
	Worst	0.00E+00	2.72E+01	0.00E+00	3.24E+10	1.44E+07
	Mean	0.00E+00	1.93E+01	0.00E+00	1.34E+10	1.11E+07
	Std	0.00E+00	4.16E+00	0.00E+00	7.69E+09	1.79E+06
jDElscop	Best	0.00E+00	0.00E+00	2.56E-13	1.80E+09	3.24E+06
	Median	0.00E+00	9.95E-01	2.77E-13	7.18E+09	4.83E+06
	Worst	0.00E+00	1.10E+01	2.91E-13	2.05E+10	6.17E+06
	Mean	0.00E+00	1.92E+00	2.78E-13	8.37E+09	4.78E+06
	Std	0.00E+00	2.67E+00	8.41E-15	4.15E+09	7.60E+05
GaDE	Best	0.00E+00	0.00E+00	2.17E-13	4.68E+09	8.72E+06
	Median	0.00E+00	0.00E+00	2.56E-13	1.01E+10	1.02E+07
	Worst	0.00E+00	1.72E-02	3.16E-13	2.79E+10	1.18E+07
	Mean	0.00E+00	6.88E-04	2.61E-13	1.12E+10	1.02E+07
	Std	0.00E+00	3.44E-03	2.95E-14	5.17E+09	9.06E+05
MOS-CEC2012	Best	0.00E+00	1.57E+03	2.09E-12	9.45E+07	1.69E+07
	Median	0.00E+00	1.75E+03	2.35E-12	3.41E+08	2.37E+07
	Worst	0.00E+00	1.94E+03	2.39E+00	5.65E+08	3.43E+07
	Mean	0.00E+00	1.75E+03	6.02E-01	3.07E+08	2.41E+07
	Std	0.00E+00	1.06E+02	8.77E-01	1.47E+08	3.93E+06
jDEsps	Best	1.95E-27	4.26E+01	4.97E-14	8.78E+08	2.05E+06
	Median	2.88E-23	9.15E+01	1.10E-13	2.56E+09	3.11E+06
	Worst	4.90E-22	7.65E+02	1.31E-12	2.72E+10	4.78E+06
	Mean	9.17E-23	1.69E+02	1.73E-13	4.41E+09	3.14E+06
	Std	1.37E-22	1.83E+02	2.55E-13	5.63E+09	6.38E+05
MOS-CEC2013	Best	0.00E+00	7.06E+02	1.54E-12	4.77E+07	5.34E+06
	Median	0.00E+00	8.24E+02	1.69E-12	7.80E+07	6.95E+06
	Worst	0.00E+00	9.29E+02	1.95E-12	1.60E+08	9.40E+06
	Mean	0.00E+00	8.23E+02	1.69E-12	8.73E+07	6.89E+06
	Std	0.00E+00	4.69E+01	9.16E-14	3.11E+07	9.16E+05
DECC-G	Best	8.47E-07	1.24E+03	1.05E-04	8.22E+10	5.73E+06
	Median	2.33E-06	1.31E+03	1.19E+00	2.08E+11	8.19E+06
	Worst	1.22E-05	1.38E+03	1.42E+00	4.12E+11	1.24E+07
	Mean	3.22E-06	1.31E+03	1.09E+00	2.16E+11	8.30E+06
	Std	2.83E-06	3.42E+01	3.54E-01	7.76E+10	1.60E+06
CC-CMA-ES	Best	1.72E-09	1.15E+03	1.39E-13	9.86E+08	7.28E+14
	Median	5.56E-09	1.34E+03	1.49E-13	1.88E+09	7.28E+14
	Worst	7.52E-09	1.69E+03	1.63E-13	6.76E+09	7.28E+14
	Mean	5.36E-09	1.37E+03	1.50E-13	2.82E+09	7.28E+14
	Std	1.32E-09	1.38E+02	6.87E-15	1.84E+09	5.18E+06

Table B.35
Comparison of all the algorithms on the CEC 2013 benchmark (F_6–F_{10}). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_6	F_7	F_8	F_9	F_{10}
MA-SW-Chains	Best	9.54E-01	3.36E+05	3.14E+13	8.09E+07	8.63E+01
	Median	6.24E+02	3.99E+06	4.65E+13	1.16E+08	3.32E+02
	Worst	5.79E+04	4.40E+06	6.42E+13	1.54E+08	1.74E+05
	Mean	1.07E+04	3.78E+06	4.63E+13	1.14E+08	3.66E+04
	Std	2.09E+04	8.46E+05	9.18E+12	2.05E+07	6.17E+04
2S-Ensemble	Best	1.59E+05	6.88E+05	1.21E+14	1.05E+08	9.22E+06
	Median	2.00E+05	1.54E+06	3.94E+14	1.34E+08	1.44E+07
	Worst	2.25E+05	4.17E+06	6.82E+14	1.55E+08	2.20E+07
	Mean	1.94E+05	1.90E+06	3.85E+14	1.31E+08	1.43E+07
	Std	1.64E+04	1.14E+06	1.39E+14	1.51E+07	2.87E+06
MOS-SOCO2011	Best	9.80E+05	9.05E+05	3.92E+14	5.02E+08	1.80E+07
	Median	9.85E+05	1.23E+07	1.13E+15	9.16E+08	8.78E+07
	Worst	9.91E+05	2.20E+08	8.82E+15	1.09E+09	8.97E+07
	Mean	9.85E+05	2.31E+07	1.64E+15	8.97E+08	6.65E+07
	Std	3.22E+03	4.42E+07	1.66E+15	1.39E+08	2.91E+07
jDElscop	Best	3.00E-07	9.32E+08	3.51E+13	2.02E+08	5.61E+02
	Median	1.99E-01	3.17E+09	1.05E+14	3.67E+08	1.01E+03
	Worst	5.66E+03	5.89E+09	2.44E+14	4.46E+08	1.30E+03
	Mean	2.42E+02	3.29E+09	1.18E+14	3.66E+08	1.06E+03
	Std	1.13E+03	1.24E+09	5.28E+13	5.57E+07	1.72E+02
GaDE	Best	1.02E+06	1.97E+09	4.87E+13	4.84E+08	1.52E+07
	Median	1.03E+06	2.68E+09	1.57E+14	6.78E+08	2.29E+07
	Worst	1.03E+06	3.23E+09	8.15E+14	7.74E+08	3.58E+07
	Mean	1.03E+06	2.67E+09	2.08E+14	6.62E+08	2.35E+07
	Std	2.03E+03	3.90E+08	1.75E+14	7.44E+07	4.59E+06
MOS-CEC2012	Best	9.85E+05	2.50E+03	1.00E+11	1.00E+09	8.81E+07
	Median	9.88E+05	1.78E+05	5.31E+11	1.65E+09	9.01E+07
	Worst	9.96E+05	3.06E+05	1.83E+12	2.33E+09	9.07E+07
	Mean	9.89E+05	1.46E+05	7.18E+11	1.65E+09	9.00E+07
	Std	2.67E+03	1.12E+05	5.42E+11	3.47E+08	5.04E+05
jDEsps	Best	3.37E-01	6.98E+08	3.92E+12	2.21E+08	8.71E+02
	Median	7.07E-01	1.14E+09	4.76E+12	2.74E+08	9.21E+02
	Worst	2.24E+01	1.95E+09	2.27E+14	3.21E+08	1.91E+03
	Mean	4.55E+00	1.24E+09	1.76E+13	2.71E+08	1.05E+03
	Std	7.90E+00	4.06E+08	4.44E+13	2.94E+07	2.43E+02
MOS-CEC2013	Best	6.16E+00	6.23E+01	5.49E+11	2.33E+08	4.44E+02
	Median	1.39E+05	1.10E+03	2.82E+12	4.18E+08	1.17E+06
	Worst	2.20E+05	5.30E+04	7.95E+12	5.01E+08	1.23E+06
	Mean	1.43E+05	4.65E+03	2.85E+12	3.99E+08	9.38E+05
	Std	6.86E+04	1.06E+04	1.44E+12	6.26E+07	4.79E+05
DECC-G	Best	1.43E+05	4.55E+08	1.44E+15	3.95E+08	7.87E+06
	Median	1.77E+05	9.10E+08	6.10E+15	5.32E+08	2.15E+07
	Worst	2.28E+05	2.20E+09	1.63E+16	7.82E+08	5.19E+07
	Mean	1.74E+05	1.02E+09	6.94E+15	5.47E+08	2.43E+07
	Std	2.09E+04	4.89E+08	3.37E+15	8.94E+07	9.75E+06
CC-CMA-ES	Best	1.18E+03	8.14E+04	6.83E+13	1.76E+08	1.32E+05
	Median	2.06E+05	1.36E+06	2.92E+14	3.57E+08	7.48E+05
	Worst	1.00E+06	1.04E+07	6.86E+14	6.77E+08	9.03E+07
	Mean	4.56E+05	2.26E+06	3.32E+14	3.82E+08	4.51E+06
	Std	4.56E+05	2.87E+06	1.74E+14	1.48E+08	1.79E+07

Table B.36
Comparison of all the algorithms on the CEC 2013 benchmark (F_{11}–F_{15}). The algorithms with the best results are highlighted (in gray) for each of the functions.

		F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
MA-SW-Chains	Best	1.59E+08	1.08E+03	1.64E+07	1.11E+08	4.57E+06
	Median	2.10E+08	1.24E+03	1.91E+07	1.47E+08	5.76E+06
	Worst	2.42E+08	1.35E+03	2.55E+07	1.70E+08	1.16E+07
	Mean	2.10E+08	1.23E+03	1.98E+07	1.45E+08	5.90E+06
	Std	2.43E+07	8.32E+01	2.30E+06	1.69E+07	1.36E+06
2S-Ensemble	Best	8.26E+07	1.07E+03	6.51E+07	3.61E+07	1.87E+06
	Median	2.45E+08	1.66E+03	2.14E+08	5.68E+07	2.37E+06
	Worst	3.35E+08	2.02E+03	7.31E+08	9.73E+07	2.61E+06
	Mean	2.38E+08	1.70E+03	2.80E+08	6.09E+07	2.33E+06
	Std	6.45E+07	2.20E+02	1.75E+08	1.75E+07	1.74E+05
MOS-SOCO2011	Best	3.83E+08	1.91E-02	2.97E+08	1.29E+08	3.40E+07
	Median	1.35E+09	7.01E+01	8.37E+08	2.30E+09	1.21E+08
	Worst	5.74E+11	2.97E+02	3.65E+09	6.55E+10	2.66E+08
	Mean	4.01E+10	8.63E+01	1.13E+09	6.89E+09	1.31E+08
	Std	1.23E+11	7.74E+01	7.74E+08	1.41E+10	6.02E+07
jDElscop	Best	1.81E+10	9.70E+02	2.17E+10	2.53E+11	2.58E+07
	Median	1.22E+11	9.74E+02	3.48E+10	4.89E+11	3.43E+07
	Worst	2.15E+11	1.03E+03	4.86E+10	6.58E+11	7.28E+07
	Mean	1.15E+11	9.91E+02	3.49E+10	4.59E+11	3.98E+07
	Std	5.86E+10	2.70E+01	5.58E+09	1.16E+11	1.50E+07
GaDE	Best	1.32E+11	1.03E+03	2.12E+10	3.13E+11	7.76E+07
	Median	2.41E+11	1.11E+03	2.87E+10	4.19E+11	1.09E+08
	Worst	3.51E+11	1.33E+03	3.54E+10	6.57E+11	1.33E+08
	Mean	2.40E+11	1.12E+03	2.88E+10	4.21E+11	1.07E+08
	Std	5.96E+10	6.90E+01	3.41E+09	7.13E+10	1.43E+07
MOS-CEC2012	Best	1.85E+07	1.50E-02	4.70E+06	7.96E+06	4.89E+06
	Median	2.53E+07	1.68E+02	5.90E+06	4.80E+07	7.25E+06
	Worst	3.89E+07	8.19E+02	6.84E+06	6.48E+07	1.04E+07
	Mean	2.71E+07	2.21E+02	5.84E+06	3.78E+07	7.63E+06
	Std	5.19E+06	2.29E+02	6.31E+05	2.01E+07	1.53E+06
jDEsps	Best	1.03E+10	3.34E+02	2.29E+10	7.54E+10	2.94E+07
	Median	2.74E+10	7.89E+02	3.12E+10	1.80E+11	4.59E+07
	Worst	5.20E+10	1.71E+03	3.82E+10	4.08E+11	7.49E+07
	Mean	2.90E+10	8.51E+02	3.07E+10	1.93E+11	4.58E+07
	Std	1.13E+10	3.64E+02	4.16E+09	8.24E+10	1.12E+07
MOS-CEC2013	Best	8.89E+06	4.40E-03	3.08E+05	7.83E+06	1.42E+06
	Median	1.71E+07	1.56E+01	1.02E+06	1.28E+07	1.68E+06
	Worst	2.85E+07	7.48E+02	2.22E+06	2.09E+07	2.03E+06
	Mean	1.73E+07	8.13E+01	1.00E+06	1.24E+07	1.71E+06
	Std	5.04E+06	1.57E+02	5.53E+05	2.86E+06	1.44E+05
DECC-G	Best	7.96E+09	3.59E+03	5.15E+09	6.77E+10	9.54E+06
	Median	1.25E+11	4.20E+03	8.66E+09	1.23E+11	1.19E+07
	Worst	2.32E+11	7.24E+03	1.75E+10	2.32E+11	1.42E+07
	Mean	1.21E+11	4.53E+03	9.40E+09	1.36E+11	1.17E+07
	Std	6.03E+10	9.48E+02	3.15E+09	4.79E+10	1.18E+06
CC-CMA-ES	Best	4.17E+07	9.78E+02	1.79E+07	2.27E+07	2.31E+07
	Median	9.79E+07	1.03E+03	4.12E+08	3.66E+07	3.05E+07
	Worst	5.04E+08	2.23E+03	1.90E+10	5.78E+09	4.29E+07
	Mean	1.24E+08	1.33E+03	1.80E+09	3.58E+08	3.13E+07
	Std	9.88E+07	4.78E+02	4.02E+09	1.15E+09	5.30E+06

Appendix C. Full comparison of the exploration capabilities of the four selected algorithms

See Tables C.37 and C.38.

Table C.37
Results of the analysis of the exploratory capabilities of the jDEsps algorithm compared with the jDElscop and MOS-SOCO2011 algorithms, respectively. We report the average p-values in a pairwise comparison. Significant p-values are highlighted in this table.

	jDEsps					
	jDElscop			MOS-SOCO2011		
	CEC 2010	SOCO 2011	CEC 2013	CEC 2010	SOCO 2011	CEC 2013
F_1	9.97E-01	9.54E-01	9.96E-01	1.00E+00	1.00E+00	1.00E+00
F_2	9.84E-01	9.88E-01	9.94E-01	1.00E+00	3.83E-02	1.00E+00
F_3	9.84E-01	9.38E-01	9.89E-01	1.00E+00	1.00E+00	1.00E+00
F_4	9.91E-01	7.98E-01	9.95E-01	9.99E-01	1.00E+00	9.98E-01
F_5	1.00E+00	9.86E-01	9.97E-01	1.00E+00	1.00E+00	9.89E-01
F_6	9.22E-01	9.98E-01	8.02E-01	1.00E+00	1.00E+00	9.92E-01
F_7	8.88E-01	9.98E-01	1.00E+00	9.92E-01	1.00E+00	9.99E-01
F_8	9.97E-01	3.30E-06	7.37E-01	1.00E+00	3.01E-01	3.37E-02
F_9	1.00E+00	9.99E-01	9.91E-01	1.00E+00	1.00E+00	5.08E-01
F_{10}	1.00E+00	9.99E-01	9.10E-01	1.00E+00	1.00E+00	7.52E-01
F_{11}	8.09E-01	9.98E-01	9.08E-01	1.00E+00	1.00E+00	1.83E-01
F_{12}	9.21E-01	9.74E-01	9.52E-01	1.00E+00	1.00E+00	1.00E+00
F_{13}	1.00E+00	9.26E-01	9.91E-01	1.00E+00	1.00E+00	3.41E-01
F_{14}	1.00E+00	9.34E-01	9.57E-01	1.00E+00	1.00E+00	2.68E-01
F_{15}	5.94E-01	9.99E-01	9.32E-01	1.00E+00	1.00E+00	4.92E-02
F_{16}	1.00E+00	9.62E-01		1.00E+00	1.00E+00	
F_{17}	7.12E-01	9.79E-01		9.87E-01	1.00E+00	
F_{18}	1.00E+00	9.86E-01		1.00E+00	1.00E+00	
F_{19}	1.41E-02	1.00E+00		4.31E-01	1.00E+00	
F_{20}	9.83E-01			1.00E+00		

Table C.38
Results of the analysis of the exploratory capabilities of the MOS-CEC2013 algorithm compared with the jDElscop and MOS-SOCO2011 algorithms, respectively. We report the average p-values in a pairwise comparison. Significant p-values are highlighted in this table.

	MOS-CEC2013					
	jDElscop			MOS-SOCO2011		
	CEC 2010	SOCO 2011	CEC 2013	CEC 2010	SOCO 2011	CEC 2013
F_1	3.36E-20	1.02E-29	4.06E-22	2.08E-11	9.00E-25	5.20E-16
F_2	4.23E-21	9.32E-25	2.54E-14	1.00E-11	2.16E-32	3.32E-13
F_3	1.37E-23	2.42E-26	5.63E-24	1.69E-22	3.65E-30	3.62E-23
F_4	1.51E-25	2.28E-24	5.08E-19	2.21E-26	8.04E-10	1.27E-20
F_5	7.95E-09	1.25E-27	6.98E-10	1.25E-13	2.93E-23	7.43E-16
F_6	5.74E-14	2.18E-26	9.61E-09	2.89E-14	8.34E-20	4.28E-07
F_7	1.69E-11	1.49E-22	3.65E-12	1.02E-14	2.64E-05	2.09E-18
F_8	2.20E-13	1.32E-25	1.90E-15	2.24E-21	4.67E-23	2.87E-20
F_9	3.34E-23	2.57E-25	1.38E-18	5.56E-19	8.23E-17	3.12E-26
F_{10}	7.18E-21	7.91E-26	1.05E-13	7.73E-20	1.48E-13	9.79E-21
F_{11}	7.43E-09	7.29E-24	3.90E-12	2.43E-01	4.27E-17	3.34E-15
F_{12}	1.48E-14	3.09E-27	3.56E-25	6.39E-13	3.43E-26	4.25E-29
F_{13}	2.31E-20	1.65E-25	1.36E-05	2.39E-27	9.62E-32	4.47E-11
F_{14}	1.19E-20	7.56E-26	2.36E-07	1.05E-14	5.34E-28	4.26E-10
F_{15}	9.85E-14	2.40E-23	2.64E-07	2.17E-07	2.57E-10	1.11E-13
F_{16}	4.70E-15	4.06E-29		9.55E-07	4.02E-27	
F_{17}	3.05E-12	3.41E-08		1.79E-11	6.51E-10	
F_{18}	6.61E-20	3.38E-34		1.14E-26	7.99E-33	
F_{19}	1.93E-20	2.34E-33		6.29E-23	3.35E-30	
F_{20}	1.70E-24			4.73E-30		

References

[1] Thomas Baguley, Serious Stats: A Guide to Advanced Statistics for the Behavioral Sciences, Palgrave Macmillan, 2012.

[2] J. Brest, B. Boškovic, A. Zamuda, I. Fister, M.S Maucec, Self-adaptive differential evolution algorithm with a small and varying population size, in: 2012 IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, 2012, pp. 1–8.

[3] J. Brest, M.S. Maucec, Self-adaptive differential evolution algorithm using population size reduction and three strategies, Soft Comput. – Fusion Found., Methodol. Appl. 15 (11) (2011) 2157–2174.

[4] B.J. Collings, M.A. Hamilton, Estimating the power of the two-sample Wilcoxon test for location shift, Biometrics 4 (3) (1988) 847–860.
[5] W.W. Daniel, Applied Nonparametric Statistics, Duxbury Thomson Learning, 1990.
[6] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617–644.
[7] A. LaTorre, S. Muelas, J.M. Peña, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test, Soft Comput. – A Fusion Found., Methodol. Appl. 15 (11) (2011) 2187–2199.
[8] A. LaTorre, S. Muelas, J.M. Pen̈a, Multiple offspring sampling in large scale global optimization, in: Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June, 2012, pp. 964–971.
[9] A. LaTorre, S. Muelas, J.M Pen̈a, Large scale global optimization: experimental results with MOS-based hybrid algorithms, in: 2013 IEEE Congress on Evolutionary Computation, CEC 2013, Cancún, Mexico, 2013, pp. 1–8.
[10] X. Li, M.N Omidvar, K. Qin, K. Tang, Z. Yang, 2013 IEEE congress on evolutionary computation competition on: large scale global optimization, in: 2013 IEEE Congress on Evolutionary Computation, CEC 2013, Cancún, 2013, p. 19.
[11] X. Li, K. Tang, M.N Omidvar, Z. Yang, K. Qin, Benchmark Functions for the CEC'2013 Special Session and Competition on Large-Scale Global Optimization, Technical Report, 2013.
[12] J. Liu, K. Tang, Scaling Up Covariance Matrix Adaptation Evolution Strategy Using Cooperative Coevolution, Technical Report, Birmingham, 2013.
[13] S.H. Liu, M. Mernik, D. Hrnčič, M. Crepinšek, A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model, Appl. Soft Comput. 13 (2013) 3792–3805.
[14] M. Lozano, D. Molina, F. Herrera, Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems, Soft Comput. – Fusion Found., Methodol. Appl. 15 (2011) 2085–2087.
[15] T.J. McCabe, A complexity measure, IEEE Trans. Softw. Eng. (4) (1976) 308–320.
[16] D. Molina, M. Lozano, F. Herrera, Memetic algorithm with local search chaining for continuous optimization problems: a scalability test, in: Proceedings of the 9th International Conference on Intelligent Systems Design and Applications, ISDA 2009, Pisa, 2009, pp. 1068–1073.
[17] D. Molina, M. Lozano, F. Herrera, MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization, in: 2010 IEEE Congress on Evolutionary Computation, CEC 2010, 2010, pp. 1–8.
[18] S. Muelas, A. LaTorre, J. M Pen̈a, A memetic differential evolution algorithm for continuous optimization, in: Proceedings of the 9th International Conference on Intelligent Systems Design and Applications, ISDA 2009, IEEE Press, 2009, pp. 1080–1084.
[19] S. Muelas, J.M. Pen̈a, V. Robles, A. LaTorre, P. de Miguel, Machine learning methods to analyze migration parameters in parallel genetic algorithms, in: E. Corchado, J.M. Corchado, A. Abraham (Eds.), Proceedings of the International Workshop on Hybrid Artificial Intelligence Systems 2007, Springer Verlag, Salamanca, Spain, 2007, pp. 199–206.
[20] M. Crepinšek, S.H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. 45 (3) (2013) 1–33.
[21] F.J. Solis, R.J.B. Wets, Minimization by random search techniques, Math. Opera. Res. 6 (1) (1981) 19–30.
[22] K. Tang, X. Li, P.N. Suganthan, Z. Yang, T. Weise, Benchmark Functions for the CEC'2010 Special Session and Competition on Large Scale Global Optimization, Technical Report, 2010.
[23] K. Tang, T. Weise, Z. Yang, X. Li, P.N. Suganthan, Results of the competition on high-dimensional global optimization at WCCI2010, in: 2010 IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, 2010, p. 19.
[24] K. Tang, Z. Yang, T. Weise, Results of the large scale global optimization challenge at 2012 IEEE world congress on computational intelligence, in: 2012 IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, 2012, p. 29.
[25] K. Tang, X. Yao, P.N. Suganthan, C. MacNish, Y.P Chen, C.M Chen, Z. Yang, Benchmark Functions for the CEC 2008 Special Session and Competition on Large Scale Global Optimization, Technical Report, China, 2007.
[26] L.Y. Tseng, C. Chen, Multiple trajectory search for large scale global optimization, in: 2008 IEEE Congress on Evolutionary Computation, CEC 2008, IEEE Press, 2008, pp. 3052–3059.
[27] H. Wang, Z. Wu, S. Rahnamayan, L. Kang, A scalability test for accelerated DE using generalized opposition-based learning, in: Proceedings of the 9th International Conference on Intelligent Systems Design and Applications, ISDA 2009, 2009, pp. 1090–1095.
[28] Y. Wang, B. Li, Two-stage based ensemble optimization for large-scale global optimization, in: IEEE Congress on Evolutionary Computation, 2010, CEC 2010, 2010, pp. 1–8.
[29] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (6) (1945) 80–83.
[30] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Inform. Sci. 178 (15) (2008) 2985–2999.
[31] Z. Yang, K. Tang, X. Yao, Scalability of generalized adaptive differential evolution for large-scale continuous optimization, Soft Comput. – Fusion Found., Methodol. Appl. 15 (2011) 2141–2155.
[32] S.Z. Zhao, P.N. Suganthan, S. Das, Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search, in: IEEE Congress on Evolutionary Computation, 2010, CEC 2010, 2010, pp. 1983–1990.