Analysis among winners of different IEEE CEC competitions on Real-parameters Optimization: Is there always improvement?

Daniel Molina
Dept. of the Computer Science
University of Cádiz, Spain
Email: daniel.molina@uca.es

Francisco Moreno-García
Dept. of Computer Science and A.I.
University of Granada, Spain
Email: fram@correo.uca.es

Francisco Herrera
Dept. of Computer Science and A.I.
University of Granada, Spain
Email: herrera@decsai.ugr.es

***Abstract*—For years, there have been organized single objective real-parameter optimization competitions on the IEEE Congress on Evolutionary Computation, in which the organizer define a common experimental, the researchers carry out the experiments with their proposals using it, and the obtained results are compared. It is a excellent way to know which algorithms (and ideas) can improve others, creating guidelines to improve the field. However, in several competitions the benchmark can change and the winners of previous benchmarks are not always introduced into the comparisons. Due to that, it could be not clear the improvement that new proposals offer against proposals of previous years. In this paper, we compare the winners in different years among them using the different proposed benchmarks, and we analyse the results obtained by all of them to observe whether there is an real improvement or not by the winner proposals of these competitions through the years.**

***Keywords*—Continuous Optimization, Comparisons, state-of-art.**

I. Introduction

Real-parameter optimization is a research topic of great interest nowadays in the community, due to the wide number of real-world applications in fields that engineering that need to be optimized. In many problems, it is usually not possible to carry out an exhaustive search, so they are tackle with evolutionary algorithms [5], because they are able to obtain good results in a reasonable time, without requiring any particular knowledge of the problems [6].

The rising interest on this type of algorithms is producing along the years a huge number of proposals (and many more proposals arise each year), from more classical proposals as Genetic Algorithms (GAs), Evolutionary Strategy, Estimation of Distribution Algorithms (EDAs) to other evolutionary algorithms like Differential Evolution (DE), many nature-inspired algorithms like Artificial Ant Colony (ACO), Particle Swarm Optimization (PSO), ... and hybrid algorithms like co-evolutionary algorithms [25], or memetic algorithms (MA) [24].

In order to compare the optimization capacities of the different proposals, in international congresses like the IEEE Congress on Evolutionary Computation it is became usual to organize each year a special session on real-parameter optimization. In many of them, organizers have presented a competition with a specific benchmark, ranking the algorithms by its results, and pointing out the algorithms considered *winners*.

These competitions are very useful, because the conclusions obtained help researchers to know with algorithms (and optimization ideas) seem to improve others, creating guidelines to improve the field. We use the term 'seem' because it is recognized that the results are related with the benchmark and experimental conditions, and these results could not be generalizable [19]. In the comparisons of these competitions only participate, in the case of IEEE CEC, the algorithms proposed (and the reference algorithms).

Unfortunately, comparing in each competition the submitted proposals only give information about the current participants in the competition. It is good when these competitions are used to continuously improving an algorithm, but sometimes an winner algorithm (or an improved version) does not compete anymore. In that case, that winner algorithm is not compared against new proposals, and there is not proved that the algorithm obtain worse results than new winner proposals (neither in the previous benchmark or the new one).

That arise a important question: Through the years, the winner algorithms of these competitions obtain best results than previous ones? It is a crucial question if we want to use these competitions as a measure of the evolution of the field. It is commonly accepted that new published proposals should be better than older ones, although sometimes it is not always the case [11]. Observing the results of these IEEE CEC competitions, researcher could guess that a winner proposal is a competitive algorithm that improves previous winners, at least under the same benchmark. However, there is not always a clear improvement from one competition to another one and, if this assumption were not true the researcher will be wrong, and one previous winner algorithms could even be better than new ones in these benchmarks.

In this paper, our goal is to discuss the open question, and to observe if winner algorithms in more recent competitions (based many times in previous ones) improve older winner algorithms from previous competitions. For doing that, we present a complete comparison among the winner algorithms of the competition CEC'2013, CEC'2014 and CEC'2015, and some other older algorithms as reference. These algorithms are going to be compared using the different benchmarks proposed in these years. In this way, we can compare them and observe if there has been actually a positive evolution in the performance of the algorithms through these years.

978-1-5090-4601-0/17/$31.00

This paper is divide as follow: In Section II, we briefly describe the algorithms chosen for the comparisons. In Section III, we introduce describe the differences benchmarks used in the comparisons, remarking the differences among them. In Section IV, we describe the experimental framework used for the comparisons. In Section V, we analyse the obtained results by the different algorithms. Finally, in Section VI, the conclusions and learned lessons are summarized.

II. PREVIOUS WINNERS IN COMPETITIONS

In this sections, we are going to briefly describe the algorithms considered in this work.

In order to compare the considered most promising algorithms in last years, we have considered the three winners of the IEEE CEC real-parameter optimization competitions from CEC'2013, CEC'2014 and CEC'2015, and two reference algorithms from previous competitions. We have not taken in account the CEC'2016 competition, because in that competition no new benchmark was proposed but the proposals were compared with a previous one, and for each previous considered benchmark the winners, UMOEAII [8] and LSHADE-EpSin [4], actually improved the previous winner, SPS-L-SHADE-EIG [12].

From CEC'2013 the winner was ICMAES-ILS[20]. The second one was NBIPOPaCMA[21]. The third with best results, with worse results than previous ones, is DRMA [15].

From CEC'2014, the clear winner was L-SHADE[31]. The second one was GaAPPADE [22]. The third one was MVMO-SH[9], indicated as MVMO14 in our study to avoid confusing it with other MVMO-SH version in following years.

From CEC'2015, the selected winners were SPS-L-SHADE-EIG [12] and LSHADE-ND[28]. DEsPA [3] is not considered because it was mainly good for dimension 50, obtaining average results for the other dimension values. Unfortunately, the new proposed MVMO algorithm [26] gave us several performance problems with dimension 50 in some problems, so we had to avoid it in the comparisons. However, because it is the third winner of its competition and the two better algorithms are considered, its incorporation, although interesting, it is not important to analyse the veracity of the open question.

Also, as reference algorithms, we compare with the winner (only the first one) of CEC'2005, I-POP-CMAES [2], and the winner of the CEC'2011 competition for real-world numerical optimization, GA-MPC[7].

Thus, we are going to compare ten competitive algorithms, eight from competitions of CEC'2013, CEC'2014 and CEC'2015, and two more as reference algorithms (older winner algorithms). In the following, sorted chronologically, we are going to describe briefly these algorithms:

- IPOP-CMA-ES [2] applies the CMA-ES algorithm [14] in a restart algorithm, and increase the population size generated in each step for each restart.
- GAMPC [7] is a GA that uses a multiple parent crossover, sort them by fitness and generate three offspring, combined with an archive of best solutions to increase diversity.
- ICMAES-ILS[20] combines the IPOP-CMA-ES algorithm with an iterative local search. It applies both algorithms over the same solution a certain number of evaluations, and whose which best results is applied the rest of evaluations.
- NBIPOPaCMA[21], it updates the covariance matrix taking in account also the the worst solutions. Also, it uses two populations that differ in the criterion followed after each restart. One of them uses the original criterion from IPOP-CMA-ES, and the other applied a reduced step-size.
- DRMA [15] is a memetic algorithm that combines GA using CMA-ES as its local search, using local search chaining [23], and dividing the domain search in hypercubes of same size with a decreasing size during the run of the algorithm (to avoid a strong dependency with the region size).
- GaAPPADE [22] divides the population into three subpopulation, each one improved by a different algorithm (GA, DE, and CMA-ES) the half of evaluations. The rest of evaluations is applied to all subpopulations the same algorithm (which have obtained best results until then).
- MVMO14 [9] is an hybrid version of MVMO with an additional improvement method. It performs the search through a population of particles that are classified in good particles or bad particles: good particles evolves toward the current best, and the bad ones are crossed with a multi-parent crossover to create better particles.
- L-SHADE [31], an improving of the SHADE[30] that combines the self-adaptive mechanism to adapt the F and CR parameter, with a continuous reduction of its population size during the search, to incrementally reduce the diversity during the run.
- L-SHADE-ND [28] is an algorithm that iterative chooses, in an adaptive way, to apply a L-SHADE (with some minor changes) or a neuro-dynamic optimization method [32].
- SPS-L-SHADE-EIG [12] modifies the L-SHADE[31] replacing the original crossover operator by a eigen-based crossover operator, and a *successful-parent-selecting* framework [13] to select differently the parents when there a stagnation is detected.

III. CONSIDERED BENCHMARKS

The different IEEE CEC benchmarks for real-parameter competitions have many common features:

- There are several functions, unimodals and multimodals, where more complex functions are composed by other ones.
- All functions are shifted from the center of the domain search, to avoid to favor algorithms bias to

the center of the domain search, and the majority are also rotated.

- The benchmark has been run with different dimension values: 10, 30 and 50. The last benchmarks also use dimension 100, but we have considered 10, 30, and 50 for a homogeneous comparison.
- The stopping criterion is the maximum evaluation number, that depends in the dimension value, increasing linearly with the dimensionality (for instance maxEvals = $10000 \cdot Dim$, where Dim is the dimensionality).

The benchmarks and the experimental conditions are described in detail in the following papers: CEC'2013 [18], CEC'2014 [17], CEC'2015 [16]. However, in the following, we are going to describe the main differences among them:

- The initial benchmark, CEC'2005 [29], proposed 25 functions well-known functions, all shifted against the center of the domain search and several rotated functions, grouping by different levels of modality: 5 unimodal functions, and 20 multimodal functions, 11 of them hybrid functions combinations of previous ones. One of the problem with this benchmark is that the function diversity is actually rather limited due to the fact that more than 40% of functions are combinations of few previous functions, so the global results of an algorithm strongly depend on these ones. CEC'2013 [18] was presented as an improve version over CEC'2005, reducing the number of hybrid functions to 8 functions, and increasing the number of basic multimodal functions to 14. In total, the diversity of functions is clearly increased, with more interesting functions. Also, while in CEC'2005 around half of functions are rotated, in CEC'2013 they are all functions (several include also a non-rotated version). In other issues CEC'2013 is more simple than CEC'2005, because all functions have the same domain search, and the same shift optimum. To summarise, CEC'2013 increase the difficulty by more rotation, and more diversity in the functions to optimize.
- CEC'2014 benchmark [17] mainly differ from CEC'2013 benchmark in the distribution of functions. The original group of unimodal functions and basic functions are rather similar (with a reduction of 2 in each category). The main difference is the following 14 functions: Inspired by the real-world problems with linkables among variables, CEC'2014 propose two groups of functions: 6 hybrid functions, in which the variables are randomly divided into subcomponents (with different percentage of membership), each subcomponent is evaluate with a different function and the result is combined; and 8 composition functions, that combine (with weights) results of several hybrid functions over the complete solution. By the use of hybrid functions to create the composition functions, they have different properties for different variables subcomponents. At difference that CEC'2013 benchmark, each function has its own shift data. To summarise, CEC'2014 change the benchmarks introducing dependencies among group of variables in the functions, to have a more realistic behavior. Also, this competition exclude any surrogate technique, because in that year they have its own competition.
- Finally, CEC'2015 learning-based optimization benchmark [27]. This benchmark is rather similar to previous one, with a clear reduction of functions, from 30 functions to only 15 functions. The main difference against previous competitions is that until now it was mandatory to use the same parameters for each function. In this competition, on the contrary, participant are allowed to optimize the parameters of their proposed (hybrid) optimization algorithm for each problem. Thus, winners in that competition could get very good results, but are too specific to each problems.

IV. Experimental framework

In this section we are going to describe the experimental framework followed into the comparison.

First, we are going to describe how the algorithms were implemented, and tested that the implementations were right. Then, we are going to describe the benchmarks used. Finally, we are going to detail the process followed for the comparisons.

A. Implementation

In order to do the comparisons with the different benchmarks, it is needed to have an implementation of the selected algorithms.

The majority of algorithms have been implemented in Java, in the public repository [1], using different mathematical libraries to simplify the process. Also, to validate the implementation not only was used several automatic tests, but it have been checked the results obtained by our implementation with the results obtained by their authors, using statistical tests to detect if any statistical difference between our implementation and original code was obtained. There was no detected any statistical difference among results obtained by our implementations and the results obtained by the original code. For DRMA and SPS-L-SHADE-EIG, the original source code were used to obtain the results.

The used parameters in each case where the parameter values used for each algorithm in its corresponding comparison. In the case of CEC'2015, algorithms could optimize their parameters for each problem, and the winner of the competition, SPS-L-SHADE-EIG, adapt its parameters for each dimension using a previous optimization process. This optimization process could imply that SPS-L-SHADE-EIG could have a worse result than expected in other benchmarks.

B. Methodology of comparisons

In this study the comparisons are carried out benchmark by benchmark.

[1] github repository: https://github.com/framg/EvolutionaryAlgorithms.git

For each benchmark the experiments are carried out for each algorithm following the experimental conditions defined in the corresponding benchmark, and the average value for function and algorithm is obtained.

The methodology used to rank the algorithms is the followed by the organizers of the competitions: First, for each function and dimension the algorithms are sorted by its average error. Then, the average ranking for the different functions are calculated by each algorithm. As a conclusion, we show the average ranking for algorithms in a table. To visualize better which algorithms obtain best results, it is generated another table with the final sorted position of each algorithm (from previous table).

In these two tables, the algorithms are listed in a chronological order, to make easier to reader to observe if new algorithms improve previous ones. Algorithms proposed in different competitions are separated by a horizontal line. The algorithms proposed in the same competition are sorted by its ranking position. In each table, the best algorithms and their results are remarked in bold.

We also have applied statistical testing. The statistical tests are been done using the KEEL software [1], in particular there are been used non-parametric tests like Friedman and Hochberg, because it is proven that parametric-tests are not adequate for these benchmarks [10]. First, the friedman test is applied, and then it is applied, as *post-hoc* method, the Holm/Hochberg. This test compares for each case the best algorithm (reference algorithm) against the rest ones, maintaining controlled the accumulated error. The tables generated by KEEL are included. The first column is the original p-value, and the other column is the adjusted *p-value* got by the Hochberg method to maintain controlled the comparison error. We have put an horizontal line dividing the algorithms identified as statistically worse than the best one (with a *p-value* of 0.05 or lower).

V. Analysis of the Algorithms by Benchmark

In this section we are going to show and analyse the ranking obtained by the selected algorithms in each one of the benchmarks. For each benchmark we compare then using the average ranking, and apply statistical tests to detect significant differences among them.

In the following, we are going to analyse the results benchmark by benchmark, and then we present several general conclusions.

A. *Results in CEC'2013 benchmark*

Table I shows the average ranking for the CEC'2013 benchmark, and Table II shows the final ranking of the algorithms.

Tables I and Table II show that the best algorithms for different dimension values are L-SHADE (dimension 10), and L-SHADE-ND (dimension 30 and 50). ICMAES-ILS, the winner of that competition maintains an average position for all dimension values, being overcome for newer algorithms. A part of L-SHADE and L-SHADE-ND, results from SPS-L-SHADE-EIG show it as a robust and competitive algorithm. GAAPPADE also obtains good results for dimension 10 and 30.

TABLE I: Average Ranking for algorithms in CEC'2013 benchmark

Algorithm	D10	D30	D50
IPOP-CMAES	5.52	6.18	6.23
GAMPC	8.11	8.04	7.79
ICMAES-ILS	4.71	5.04	4.95
NBIPOPaCMA	6.68	6.77	6.80
DRMA	5.12	5.30	5.64
L-SHADE	**4.32**	**4.27**	**4.16**
GAAPPADE	**4.64**	**4.27**	**5.12**
MVMO14	6.12	6.41	6.07
SPS-L-SHADE-EIG	**4.52**	**4.70**	**4.34**
L-SHADE-ND	**5.25**	**4.04**	**3.89**

TABLE II: Position of algorithms based on their average ranking in CEC'2013 benchmark

Algorithm	D10	D30	D50
IPOP-CMAES	7	7	8
GAMPC	10	10	10
ICMAES-ILS	4	5	4
NBIPOPaCMA	9	9	9
DRMA	5	6	6
L-SHADE	**1**	**2.5**	**2**
GAAPPADE	**3**	**2.5**	**5**
MVMO14	8	8	7
SPS-L-SHADE-EIG	**2**	**4**	**3**
L-SHADE-ND	**6**	**1**	**1**

In Tables III, IV and V there are shown the statistical test results for dimension 10, 30, and 50, respectively. For dimension 10 only are considered as statistically worse GAMPC and NBIPOPaCMA. For dimension 30 and 50, there is a group of algorithms: DRMA, ICMAES-ILS, GAAPPADE, L-SHADE, and L-SHADE-ND among which there is not detected any statistical difference.

TABLE III: Post Hoc comparison Hochberg vs L-SHADE for D=10, CEC'2013

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000003	0.000026
2	NBIPOPaCMA	0.003579	0.028636
3	MVMO14	0.02582	0.180741
4	IPOP-CMAES	0.139252	0.808197
5	L-SHADE-ND	0.251152	0.808197
6	DRMA	0.320673	0.808197
7	ICMAES-ILS	0.627319	0.808197
8	GAAPPADE	0.691197	0.808197
9	SPS-L-SHADE-EIG	0.808197	0.808197

B. *Results in CEC'2014 benchmark*

Table VI shows the average ranking for the CEC'2014 benchmark, and Table VII shows the final ranking of the algorithms.

TABLE IV: Post Hoc comparison Hochberg vs L-SHADE-ND for D=30, CEC'2013

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000001	0.000007
2	NBIPOPaCMA	0.000734	0.005874
3	MVMO14	0.003334	0.023341
4	IPOP-CMAES	0.008092	0.048552
5	DRMA	0.117149	0.585744
6	ICMAES-ILS	0.216522	0.774197
7	SPS-L-SHADE-EIG	0.414197	0.774197
8	GAAPPADE	0.774197	0.774197
9	L-SHADE	0.774197	0.774197

TABLE V: Post Hoc comparison Hochberg vs L-SHADE-ND for D=50, CEC'2013

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000002	0.000014
2	NBIPOPaCMA	0.000322	0.002574
3	IPOP-CMAES	0.003841	0.026885
4	MVMO14	0.007095	0.042571
5	DRMA	0.030564	0.152821
6	GAAPPADE	0.127829	0.511318
7	ICMAES-ILS	0.192905	0.578714
8	SPS-L-SHADE-EIG	0.581148	0.740625
9	L-SHADE	0.740625	0.740625

TABLE VI: Average Ranking for algorithms in CEC'2014 benchmark

Algorithm	D10	D30	D50
IPOP-CMAES	5.62	5.48	5.18
GAMP	7.67	7.50	8.03
ICMAESILS	**5.17**	**4.43**	**4.07**
NBIPOPaCMA	6.88	6.35	6.02
DRMA	5.23	5.53	5.65
L-SHADE	**4.18**	**3.93**	**4.27**
GAAPPADE	6.18	5.82	5.53
MVMO14	5.45	7.08	7.05
SPS-L-SHADE-EIG	**4.57**	**4.65**	**4.57**
L-SHADE-ND	**4.05**	**4.22**	**4.63**

TABLE VII: Position of algorithms based on their average ranking in CEC'2014 benchmark

Algorithm	D10	D30	D50
IPOP-CMAES	7	5	5
GAMP	10	10	10
ICMAESILS	**4**	**3**	**1**
NBIPOPaCMA	9	8	8
DRMA	5	6	7
L-SHADE	**2**	**1**	**2**
GAAPPADE	8	7	6
MVMO14	6	9	9
SPS-L-SHADE-EIG	**3**	**4**	**3**
L-SHADE-ND	**1**	**2**	**4**

From Tables VI and VII it can be observed that the best algorithms are L-SHADE, L-SHADE-ND and ICMAES-ILS. L-SHADE, the winner of that competition is still the best algorithm in average, but L-SHADE-ND obtains best results for dimension 10. While L-SHADE and L-SHADE-ND are better for dimensions 10 and 30, ICMAES-ILS get the best results for the higher dimension, 50. SPS-L-SHADE-EIG have also good results.

TABLE VIII: Post Hoc comparison Hochberg vs L-SHADE-ND for D=10, CEC'2014

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000004	0.000033
2	NBIPOPaCMA	0.00029	0.002317
3	GAAPPADE	0.006353	0.044473
4	IPOP-CMAES	0.045061	0.270365
5	MVMO14	0.073312	0.366558
6	DRMA	0.130096	0.459492
7	ICMAES-ILS	0.153164	0.459492
8	SPS-L-SHADE-EIG	0.508662	0.864569
9	L-SHADE	0.864569	0.864569

TABLE IX: Post Hoc comparison Hochberg vs L-SHADE for D=30, CEC'2014

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000004	0.000033
2	MVMO14	0.000056	0.000447
3	NBIPOPaCMA	0.001384	0.009687
4	SPS-L-SHADE-EIG	0.003737	0.022423
5	GAAPPADE	0.020132	0.100658
6	IPOP-CMAES	0.036675	0.115907
7	DRMA	0.038636	0.115907
8	ICMAES-ILS	0.405701	0.749119
9	L-SHADE-ND	0.749119	0.749119

TABLE X: Post Hoc comparison Hochberg vs ICMAES-ILS for D=50, CEC'2014

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000005	0.000045
2	MVMO14	0.000056	0.000447
3	NBIPOPaCMA	0.001992	0.013945
4	GAAPPADE	0.015989	0.095934
5	DRMA	0.040685	0.189574
6	IPOP-CMAES	0.047393	0.189574
7	SPS-L-SHADE-EIG	0.359267	0.717022
8	ICMAES-ILS	0.522431	0.717022
9	L-SHADE-ND	0.717022	0.717022

In Tables VIII, IX and X there are shown the statistical-test results for dimension 10, 30, and 50, respectively. For dimension 10, the best one is L-SHADE-ND and improve significantly three algorithms: GAMP, NBIPOPaCMA, and GAAPPADE. For dimension 30, the best one is L-SHADE and statistically improves GAMPC, MVMO14, NBIPOPaCMA, and SPS-L-SHADE-EIG. For dimension 50, the best one is ICMAES-ILS, that improves GAMPC, MVMO14 and NBIPOPaCMA.

C. Results in CEC'2015 benchmark

Table XI shows the average ranking for the CEC'2015 benchmark, and Table XII shows the final ranking of the compared algorithms.

TABLE XI: Ranking of algorithms by its average ranking in CEC'2015 benchmark

Algorithms	D10	D30	D50
IPOP-CMAES	7.68	7.00	6.07
GAMPC	7.73	7.47	8.60
ICMAES-ILS	5.93	5.40	5.27
NBIPOPaCMA	7.67	7.20	5.93
DRMA	**5.10**	**5.30**	**4.67**
L-SHADE	5.17	5.23	5.23
GAAPPADE	**3.00**	**2.60**	**2.93**
MVMO14	5.80	7.27	7.80
SPS-L-SHADE-EIG	**3.30**	**2.37**	**3.27**
L-SHADE-ND	**4.23**	**5.17**	**5.23**

TABLE XII: Position of algorithms based on their average ranking in CEC'2015 benchmark

Algorithm	D10	D30	D50
IPOP-CMAES	9	7	8
GAMPC	10	10	10
ICMAES-ILS	7	6	6
NBIPOPaCMA	8	8	7
DRMA	4	5	3
L-SHADE	5	4	4.5
GAAPPADE	**1**	**2**	**1**
MVMO14	6	9	9
SPS-L-SHADE-EIG	**2**	**1**	**2**
L-SHADE-ND	**3**	**3**	**4.5**

From Tables XI and XII it can be observed that the best algorithm is GAAPPADE, and then the following algorithms are SPS-L-SHADE-EIG and L-SHADE-ND. More in detail, for dimension 10 and 50 the best one was GAAPPADE, and only for dimension 30 it was SPS-L-SHADE-EIG, the winner of the competition. This is remarkable, not only because GAAPPADE is older than SPS-L-SHADE-EIG but also because the SPS-L-SHADE-EIG parameter values were specially optimized for the CEC'2015 benchmark, contrary to GAAPPADE.

In Tables XIII, XIV and XV there are shown the statistical-test results for dimension 10, 30, and 50, respectively. It can be observed the clear difference among SPS-L-SHADE-EIG and GAAPPADE in dimension 30 against the rest algorithms. For the other dimension values, the algorithms clearly improved by the best one is similar to obtained with the other benchmarks.

D. General Results

Once we have detected which algorithms obtain the best results for benchmark, we can observe also several interesting conclusions, in no particular order:

TABLE XIII: Post Hoc comparison Hochberg vs GAAPPADE for D=10, CEC'2015

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000019	0.000167
2	NBIPOPaCMA	0.000024	0.000194
3	IPOP-CMAES	0.000235	0.001643
4	ICMAES-ILS	0.007971	0.047825
5	MVMO14	0.011319	0.056595
6	L-SHADE	0.050016	0.172488
7	DRMA	0.057496	0.172488
8	L-SHADE-ND	0.264597	0.529194
9	SPS-L-SHADE-EIG	0.786114	0.786114

TABLE XIV: Post Hoc comparison Hochberg vs SPS-L-SHADE-EIG for D=30, CEC'2015

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0.000004	0.000036
2	MVMO14	0.000009	0.000075
3	NBIPOPaCMA	0.000012	0.000086
4	IPOP-CMAES	0.000028	0.000167
5	ICMAES-ILS	0.006074	0.022638
6	DRMA	0.007971	0.022638
7	L-SHADE	0.009514	0.022638
8	L-SHADE-ND	0.011319	0.022638
9	GAAPPADE	0.832842	0.832842

TABLE XV: Post Hoc comparison Hochberg vs GAAPPADE, CEC'2015

i	Algorithm	unadjusted p	$p_{Hochberg}$
1	GAMPC	0	0.000003
2	MVMO14	0.000011	0.000086
3	IPOP-CMAES	0.004594	0.032158
4	NBIPOPaCMA	0.006656	0.039934
5	ICMAES-ILS	0.034808	0.112459
6	L-SHADE	0.037486	0.112459
7	L-SHADE-ND	0.037486	0.112459
8	DRMA	0.116914	0.233828
9	SPS-L-SHADE-EIG	0.763025	0.763025

The algorithm GAMP obtained in the majority of cases the worse results. One possible reason is that the CEC'2011 benchmark, that consider a group of real-word problems, is very different to the rest ones. Thus, algorithm with a good results in that benchmark does not imply to be competitive in the others.

L-SHADE and L-SHADE-ND are algorithms very robust, obtaining the best results in more benchmarks.

Algorithms IPOP-CMA-ES and NBIPOPaCMA have been clearly improved by newer proposals in all the benchmarks, showing a clear improvement in the last ten years. However, CMA-ES is used as component in competitive algorithms like ICMAES-ILS, DRMA, or GAAPPADE. ICMAES-ILS and DRMA still give competitive results, with no statistically different among them and the best one in the majority of cases. GAAPPADE is the algorithm with best results in the CEC'2015 benchmarks, being for a previous competition.

SPS-L-SHADE-EIG obtains the best results in CEC'2015 but it is shown as a robust algorithm but not the best in the other comparisons. Its optimization of parameters allowed it to obtain the best results in that competitions, but not in the other ones.

From a statistical point, there was not detected any significant improvement among L-SHADE, SPS-L-SHADE-EIG, L-SHADE-ND, ICMAES-ILS and DRMA in any of the comparatives, so they could be considered rather robust algorithms. The other algorithms were detected as statistically worse in at least one case. Thus, algorithms like IPOP-CMAES or NBIPOPaCMA could be considered improved in the last years.

ICMAES-ILS, winner of the CEC'2013 competition have obtained the best results in the CEC'2014 benchmark for dimension 10, and GAAPPADE in the CEC'2015, showing that they are competitive algorithm that should still been taken in account.

VI. Learned Lessons and Conclusions

Through the years, several competitions on real-parameter optimization have been proposed. In these competitions each proposal include the obtained results with a proposed benchmark, and the organizer rank the participants by their results. However, each year only the proposals are considered, and the previous winners are not taking in account, so that new winner algorithms are more competitive (or better) is a general assumption but it is not tested.

We are compared the winner of the IEEE CEC competitions on real-parameter optimization from 2013, 2014, and 2015 in these benchmarks.

We have learned that, while the results obtained by one benchmark cannot be generalizable, comparing with different benchmarks remark the existence of several rather robust algorithms (like L-SHADE, L-SHADE-ND, SPS-L-SHADE-EIG). Also, although there is a group of algorithms that are worse significantly in the majority of cases, and there is another one (composed by L-SHADE, L-SHADE-ND, GAAPPADE, DRMA) among which there is not detected significant differences with the best one. Several algorithms like IPOP-CMAES o NBIPOPaCMA have been clearly improved but there are anothers, like GAAPPADE, ICMAES-ILS o DRMA that can be still considered rather competitive.

As learned lesssons we observe that not always the algorithms with best results are the proposed by that benchmark. Thus, answering to the question: *Through the years, the winner algorithms of these competitions obtain best results than previous ones?* The results show that it is not always true. Results from ICMAES-ILS and GAAPPADE show that previous winner can improve new ones in new competitions. These algorithms are unfairly ignored while there are very competitive in modern benchmarks (no algorithm significantly improve them for any benchmark or dimension, and while ICMAES-ILS obtains the best results in CEC'2014 benchmark in dimension 50, GAAPPADE is the best algorithm for the CEC'2015 benchmark).

For future competitions, as a learned lesson, we encourage the organizers to include the winner (or winners) of previous competitions and use them as reference algorithms for the new competitions. In that way, it could be checked if the new proposals improve previous ones in the new benchmark. In several competitions it is already done, using a previous winner as a algorithm reference, but this good practice is not as widely adopted as it should be.

Acknowledgment

This work was supported by Research National Projects TIN2014-57251-P and TIN2016-81113-R.

References

[1] J. Alcal-Fdez, A. Fernndez, J. Luengo, J. Derrac, and S. Garca. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.

[2] A Auger and N Hansen. A Restart CMA Evolution Strategy with Increasing Population Size. In *2005 IEEE Congress on Evolutionary Computation*, pages 1769–1776, 2005.

[3] N. Awad, M. Z. Ali, and R. G. Reynolds. A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1098–1105, May 2015.

[4] N.H. Awad, M.Z. Ali, P.N. Suganthan, and R.G. Reynolds. An Ensemble Sinusoidal Parameter Adaptation incorporated with L-SHADE for Solving CEC2014 Benchmark Problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2958–2965, 2016.

[5] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, 1997.

[6] H.G. Beyer and K. Deb. On Self-adaptive Features in Real-parameter Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3(5):250–270, 2001.

[7] S. M. Elsayed, R. A. Sarker, and D. L. Essam. Ga with a new multi-parent crossover for solving ieee-cec2011 competition problems. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1034–1040, June 2011.

[8] S. M. Elsayed, R. A. Sarker, D. L. Essam, and N. M. Hamza. Testing united multi-operator evolutionary algorithms on the cec2014 real-parameter numerical optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1650–1657, July 2014.

[9] I. Erlich, J. L. Rueda, S. Wildenhues, and F. Shewarega. Evaluating the mean-variance mapping optimization on the ieee-cec 2014 test suite. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1625–1632, July 2014.

[10] S. Garcia, D. Molina, M. Lozano, and 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. *Journal of Heuristics*, 15(6):617–644, 2009.

[11] C. García-Martínez, P.D. Gutiérrez, D. Molina, and F. Herrera. Since cec 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis's weakness. *Soft Computing*, pages 1–11, 2017.

[12] S. M. Guo, J. S. H. Tsai, C. C. Yang, and P. H. Hsu. A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1003–1010, May 2015.

[13] S.M. Guo, C.C. Yang, P.H. Hsu, and J.S.H. Tsai. Improving differential evolution with successful-parent-selecting framework. *IEEE Transaction on Evolutionary Computation*, 19(5), 2015.

[14] N Hansen, S D Müller, and P Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 1(11):1–18, 2003.

[15] B. Lacroix, D. Molina, and F. Herrera. Dynamically updated region based memetic algorithm for the 2013 cec special session and competition on real parameter single objective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1945–1951, June 2013.

[16] J.J. Liang, Q. Chen, B.Y. Qu, B. Liu, P.N. Suganthan, and Q. Chen. Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Nov 2014.

[17] J.J. Liang, B-Y. Qu, and P.N. Suganthan. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, 2013.

[18] J.J. Liang, B-Y. Qu, P.N. Suganthan, and A.G. Hernndez-Daz. Problem definitions and evaluation criteria for the cec 2013 special session and competition on real-parameter optimization. Technical report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.

[19] T. Liao, D. Molina, and T. Stützle. Performance evaluation of automatically tuned continuous optimizers on different benchmark sets. *Applied Soft Computing Journal*, 27:490–503, 2015.

[20] T. Liao and T. Sttzle. Benchmark results for a simple hybrid algorithm on the cec 2013 benchmark set for real-parameter optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1938–1944, June 2013.

[21] I. Loshchilov. Cma-es with restarts for solving cec 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 369–376, June 2013.

[22] R. Mallipeddi, G. Wu, M. Lee, and P. N. Suganthan. Gaussian adaptation based parameter adaptation for differential evolution. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1760–1767, July 2014.

[23] D. Molina, M. Lozano, C. Garca-Martnez, and F. Herrera. Memetic algorithms for continuous optimisation based on local search chains. *Evolutionary Computation*, 18(1):27–63, 2010.

[24] P. Moscato and C. Cotta. *Handbook of Metaheuristics*, chapter A Gentle Introduction to Memetic Algorithms, pages 105–144. Kluwer Academic Publishers, Boston MA, 2003.

[25] Jan Paredis. Coevolutionary computation. *Artificial Life*, 2(4):355–375, 1995.

[26] J. L. Rueda and I. Erlich. Testing MVMO on learning-based real-parameter single objective benchmark optimization problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1025–1032, May 2015.

[27] P.N. Suganthan S. Das. Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems. Technical report, Jadavpur University, India and Nanyang, 2010.

[28] K. M. Sallam, R. A. Sarker, D. L. Essam, and S. M. Elsayed. Neurodynamic differential evolution algorithm and solving CEC2015 competition problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1033–1040, May 2015.

[29] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical report, Nanyang Technological University, Singapore, 2005.

[30] R. Tanabe and A. Fukunaga. Evaluating the performance of shade on cec 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 1952–1959, June 2013.

[31] R. Tanabe and A. S. Fukunaga. Improving the search performance of shade using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1658–1665, July 2014.

[32] Youshen Xia and Jun Wang. A general projection neural network for solving monotone variational inequalities and related optimization problems. *IEEE Transactions on Neural Networks*, 15(2):318–328, March 2004.