Cognitive Computation
https://doi.org/10.1007/s12559-018-9554-0

An Insight into Bio-inspired and Evolutionary Algorithms for Global Optimization: Review, Analysis, and Lessons Learnt over a Decade of Competitions

Daniel Molina [1] · Antonio LaTorre [2] · Francisco Herrera [1]

Received: 7 August 2017 / Accepted: 3 April 2018

Abstract

Over the recent years, continuous optimization has significantly evolved to become the mature research field it is nowadays. Through this process, evolutionary algorithms had an important role, as they are able to obtain good results with limited resources. Among them, bio-inspired algorithms, which mimic cooperative and competitive behaviors observed in animals, are a very active field, with more proposals every year. This increment in the number of optimization algorithms is apparent in the many competitions held at corresponding special sessions in the last 10 years. In these competitions, several algorithms or ideas have become points of reference, and used as starting points for more advanced algorithms in following competitions. In this paper, we have obtained, for different real-parameter competitions, their benchmarks, participants, and winners (from the competitions' website) and we review the most relevant algorithms and techniques, presenting the trajectory they have followed over the last years and how some of these works have deeply influenced the top performing algorithms of today. The aim is to be both a useful reference for researchers new to this interesting research topic and a useful guide for current researchers in the field. We have observed that there are several algorithms, like the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), the Success-History based Adaptive Differential Evolution with Linear Population Size Reduction (L-SHADE), Mean-Variance Mapping Optimization (MVMO), and Multiple Offspring Sampling (MOS), which have obtained a strong influence over other algorithms. We have also suggested several techniques that are being widely adopted among the winning proposals, and that could be used for more competitive algorithms. Global optimization is a mature research field in continuous improvement, and the history of competitions provides useful information about the past that can help us to learn how to go forward in the future.

Keywords Continuous optimization · Global optimization · Large-scale global optimization · Multimodal optimization · Real-parameter competitions

Preliminaries

Global optimization, also referred to as real-parameter optimization, or continuous optimization, is a topic of great interest nowadays in the research community, due to the wide number of real-world applications in fields such as engineering that need to be optimized. Global optimization implies the minimization or maximization of a specific *objective* function (but we can consider only minimization without loss of generality). Mathematically, minimize objective function means to find x^* defined by Eq. 1.

$$x^* \in [a, b] \text{ where } f(x^*) \leq f(x), \forall x \in [a, b] \tag{1}$$

where $a, b \in \mathrm{R}^N$ and $f: \mathrm{R}^N \rightarrow \mathrm{R}$ is called the *objective* function or *fitness* function.

Due to both the real number nature and the extensive domain search, it is usually not possible to perform an exhaustive search in the entire domain, so some stochastic algorithms like meta-heuristics algorithms [1, 2] or evolutionary algorithms [3] are usually applied. They are able to obtain good results

✉ Daniel Molina
dmolina@decsai.ugr.es

Antonio LaTorre
atorre@fi.upm.es

Francisco Herrera
herrera@decsai.ugr.es

1 Department of Computer Science, University of Granada, Granada, Spain

2 Center for Computational Simulation, Universidad Politécnica de Madrid, Madrid, Spain

(the best one cannot be guaranteed) in a reasonable time (measured in processing time or limit of evaluated solutions), and without requiring any particular knowledge about the optimization problem [4]. Furthermore, these algorithms have been successfully applied to a large number of problems related to cognitive issues, feature selection [5], etc.

The recent and growing interest on this type of optimization has generated a huge number of optimization algorithms that tackle this type of optimization (and many more proposals arise each year), from more classical proposals such as genetic algorithms (GAs) [6, 7], simulated annealing (SA) [8], evolutionary strategy [9], estimation of distribution algorithms (EDAs) [10] to other evolutionary algorithms such as differential evolution (DE) [11, 12], nature-inspired algorithms [13], bio-inspired algorithms [14], hybrid algorithms such as co-evolutionary algorithms, that divide the problem in subproblems and then tackle each one using an algorithm [15, 16], and memetic algorithms (MA) [17], in which several different algorithms are used to solve the problem (usually one does the global exploration and another improves the achieved solutions).

These algorithms are very useful, and many of them have been successfully applied to many different optimization problems, including several difficult cognitive ones such as image recognition [18], identification of diseases [19], or detection of taxonomies [20].

Unfortunately, the great number of proposals makes following the evolution of this field difficult, and there is no clear criterion to select the most adequate algorithms. Even so, sometimes authors experiments on different functions and under different experimental conditions, disabling the direct comparison of results obtained by the different algorithms.

In order to give more visibility to the field, many special sessions have been proposed in international congresses such as the *IEEE Congress on Evolutionary Computation* (CEC) and in the *Genetic and Evolutionary Computation Conference* (GECCO). In these special sessions, real-parameter optimization competitions among the algorithms have been proposed. In these competitions, organizers have presented a specific benchmark with the implementation of the fitness functions to optimize and all the experimental conditions to allow fair comparisons among the proposals. Using the results obtained in these competitions, the evolution of this research topic can be observed over time, obtaining interesting conclusions.

In this paper, we are going to describe the different real-coding optimization competitions, noting the benchmark used for each one and briefly describing for each year the winners, obtaining several conclusions of the evolution of the competition. Then we are going to discuss several issues that we consider interesting about the research topic, the influence of winning algorithms over the years, and the evolution of real-parameter optimization.

This paper has the following structure: In "Global Optimization Competitions" section, we are going to describe benchmarks and winning algorithms in several competitions for real-parameter optimization (without constraints).

In "Constraint Real-Parameter Optimization" section, we present the evolution of a more specific optimization, real-parameter optimization with constraints. In "Multimodal Optimization" section, we study the competitions related with another particular optimization, multimodal optimization, in which the goal is to obtain all optima. In "Large-Scale Optimization" section, we are going to study the evolution and algorithms tackling problems with a higher dimensionality (with dimension 1000 or higher), which is called large-scale global optimization. In "Current Trends After a Decade of Competitions" section, we are going to discuss, considering all the competitions, the algorithms that have shown a stronger influence, and also the techniques that are starting to be widely adopted for algorithms in real-parameter optimization.

In "Conclusions" section, we are going to summarize the main conclusions obtained after this roadmap in global optimization. Finally, in the Appendix, we include a list of tables with the summary of the algorithms remarked in the different competitions.

Global Optimization Competitions

In this section, we are going to focus on global real-parameter optimization benchmarks with no particular constraints (only boundary constraints) in which the aim is to find one global optimum.

Historically, there have been different types of competitions, each one organized in the international congresses *IEEE Congress on Evolutionary Computation* (CEC) and *Genetic and Evolutionary Computation Congress* (GECCO). In each congress, the same group of organizers has proposed, in different years, several special competition sessions. Even though over time the proposed benchmark may differ, there is usually more similarity among the competitions within the same congress (CEC or GECCO) than among those in different congresses. Thus, we are going to describe and analyze each congress in different sections.

In the CEC special sessions, since 2005, several competitions have been traditionally proposed. Over the years, the proposed benchmark has evolved along with the algorithms that have participated in them. In the GECCO competition, a particular benchmark, called Black-Box Optimization Benchmark (BBOB), is used for the competitions. Recently, in 2015 and 2016, another type of competition, which is not related to any special session, has been proposed both in CEC and in GECCO, with the particularity of being completely black-box, since researchers have not knowledge about the problems to be solved.

First, in "IEEE Real-Coding Special Session Competitions" section, we are going to describe the different real-coding competitions in the CEC over the years. Then, in "BBOB Competition" section, we describe the different competitions in GECCO using the BBOB. In "Black-Box Optimization Competition", we describe the main results of the black-box competitions. Finally, in "Evolution of Real-Parameter Optimization: Lessons Learnt" section, we note several lessons learnt through the evolution of real-parameter competitions. Note that we will refer to each algorithm throughout this paper by the name given by the authors in the special session where it was proposed, without expanding the full acronym. The motivation for this is twofold: first, many of these methods are complex algorithms combining multiple strategies which leads to loosely defined names with a lot of components. Second, these names are already used to cite them in the literature (including the competitions comparisons carried out by the organizers). Nonetheless, each acronym is accompanied by the corresponding reference the first time it is introduced.

IEEE Real-Coding Special Session Competitions

In 1996, a benchmark was proposed, with both real-optimization and Traveling Salesman Problems, for a first competition [21]. Unfortunately, it was not widely adopted. It was not until 2005 when more consolidated benchmarks on real-parameter optimization were proposed, for a competition within the IEEE CEC [22, 23].

Over the following years, other special sessions with related competitions were proposed, with small differences in the benchmarks. However, they share the same following features:

- They are composed of a great number of functions to optimize with different domain searches.
- All functions have been shifted to avoid benefiting algorithms with bias to the center of the domain search. Also, several of them were rotated, especially in last benchmarks.
- The functions were divided in unimodal and multimodals, and new functions were composed of previous functions.
- Each function is evaluated 25 times, and the average for each function is used to compare results.
- The functions are evaluated with different dimension values: 10, 30, 50 (and 100 in recent competitions).
- The stopping criterion is the maximum evaluation number, maxEvals, which depends on the dimension value, using maxEvals = 10,000 × Dim where Dim is the dimensionality.

We can see an interesting evolution in the proposals over the past few years. In addition, most information regarding the competitions is available online,[1] including the definition of benchmarks, comparative slides, results data, and the source code of several of the competitions winners.

In this section, for each one of the considered real-parameter competitions, we are going to indicate the view of the competitors and the winning algorithms (taken from the analysis carried out by the competition organizers). Initially, we are going to consider the first three algorithms, but this value can vary considering the degree of differences among the algorithms.

CEC'2005 Real-Parameter Special Session

As previously mentioned, CEC'2005 was the first of a series of real-parameter competitions. Its benchmarks have been considered references, and it has been used in many papers.

In this first important competition, there were very different 11 algorithms[2]: three GAs, two DEs, two ES algorithms, one PSOs, and one co-evolutionary algorithm. The competition had such a high diversity and generated great interest perhaps because previously there had been not clear references regarding the performance of each model.

The clear winners were Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [24]-based algorithms, in particular, IPOP-CMA-ES [25], called G-CMA-ES in the competition, and L-CMA-ES [26]. CMA-ES was a previously published evolution strategy [24] with a very sophisticated adaptation of its parameter. L-CMA-ES [26] applied CMA-ES with a restarting criterion. IPOP-CMA-ES differ from the previous one in that, at each restart, the number of solutions generated in each step is increased, multiplied by a factor, obtaining better behavior in multimodal functions. Other algorithms that obtained good results were EDA [27] in unimodals and L-SaDE [28] and DMS-L-PSO [29] in multimodals. Both L-SaDE and DMS-L-PSO are MAs combining an exploratory algorithm with an improvement algorithm to improve obtained solutions, using in both cases the quasi-newton algorithm. In the case of L-SaDE, the algorithm is a self-adaptive DE, and in DMS-L-PSO, a multi-swarm PSO that randomly groups the individuals into sub-populations several times during the run.

This competition was important not only because it was the starting point for more popular competitions but also because the winner, IPOP-CMA-ES, has become an important reference in current research on real-parameter optimization.

[1] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm for the CEC'2005 competitions and http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/Forms/AllItems.aspx for the rest of CEC special sessions

[2] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm

CEC'2011 Real-World Numerical Optimization Special Session

In the CEC'2011, the benchmark was rather different of that proposed in previous competitions and in those to come. Instead of being composed by classical functions like other benchmarks, this benchmark was created with real-world numerical optimization problems [30]. Composed of 13 problems, the benchmark measures the error for different evaluation numbers, and obtains a mean of 25 runs.

Fourteen algorithms participate in the competition[3]: ten DEs, three GAs, and one other algorithm. Again, the majority of proposals in real-parameter competitions were DEs (70%), but in this case, there were a considerable number of GAs (20% of proposals). The majority of proposals were also MAs.

The winner was GA-MPC [31], a GA that uses a multiple parent crossover which sorts by fitness and generates three offspring, combined with an archive of best solutions to increase diversity.

The second one was DE−ΛCR [32], a DE that adapts the F and CR parameters using triangular distributions to adapt the F and CR values in function of the separability of the function, and it uses a sequential quadratic routine as the LS method. The winning algorithm that came last was SAMODE [33], proposed by the same authors as GA-MPC, SAMODE is one DE with four different mutation types and four different crossover operators, and it self-adapts the number of applications of each one according to the performance obtained in previous iterations.

Because there are many differences in the benchmark as compared to other years, the tendencies are different. First, it seems that in this benchmarks the different problems greatly differ in behavior, so the algorithms with the best results are those which adapt their parameters to the problem to be solved the fastest.

CEC'2013 Real-Parameter Special Session

In 2013, there was another real-parameter optimization competition in CEC that was accompanied by a new benchmark [34]. The main changes (as compared to the previous one, CEC'2005) are the following:

- The functions have changed, 28 instead of 25, 5 unimodals, 15 multimodal functions, and 8 composition functions.
- All functions are rotated.
- The optimum is different for each function.
- The same search ranges are used for all test functions, with all functions shifted to obtain a different optimum.

[3] http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC11-RWP.htm

The special session received 21 algorithms[4]: 8 DEs, 5 based on CMA-ES, 2 PSOs (one of them combining a PSO with an ABC), and other algorithms.

The winner was iCMAES-ILS [35], a hybrid algorithm that combines the IPOP-CMA-ES algorithm with an iterative local search. iCMAES-ILS applies both algorithms to the same solution (to which a certain ratio of evaluations are applied), and the best results are applied to the remaining evaluations. The second one was NBIPOPaCMA [36] that uses active CMA-ES, and this implies that the covariance matrix update has been done taking into account not only the best solutions but also the worst [37]. In addition, it has two restart regimes, one that reduces the step-size [38] and another with the default values. The third with best results, with worse results than previous ones, is DRMA-LSCh-CMA [39], a memetic algorithm that combines a GA with a CMA-ES as its local search using local search chaining [40], and dividing the domain search in hypercubes of equal size, with decreasing size during the search. Also, several algorithms appear which, although they do not get good results, they are the precursors of next winning algorithms, like MVMO [41] and SHADE [42].

In this year, all winning proposals have used or BI-POP-CMA-ES as an exploration method (with several improvements) or CMA-ES as their Local Search. Thus, 8 years after the CEC'2005 benchmark, the CMA-ES was still considered an important part of successful algorithms.

CEC'2014 Single Objective Real-Parameter Special Session

In this special session, two different benchmarks were proposed:

1. One benchmark for single objective real-parameter numerical optimization [43]. This benchmark was created from comments received from previous one [34]. It reduces the number of unimodal functions (to 3), 13 simple multimodal functions, 6 hybrid functions, and 7 composition functions. All functions have not only been shifted but also rotated.
2. One new benchmark for expensive optimization [44]. This benchmark was created to find algorithms that obtain good results with a very limited number of evaluations. There are eight functions with dimension values: 10, 20, and 30. Each algorithm is run 20 times for function, and the maximum number of evaluations is $50 \times Dim$ (in contrast to other benchmarks, in which it is about $10{,}000 \times D$). This radical reduction of evaluations implies a challenge for the algorithms that could use techniques like surrogate, or increase performance.

[4] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm

In the following, we are going to describe results of the two competitions.

First, in the single objective optimization, 17 algorithms were proposed[5]: with 9 DEs, showing as the more popular category of real-parameter algorithms.

The clear winner was L-SHADE [45], an improved version of SHADE [42] that reduces its population size during the search to incrementally reduce the diversity during the run. Moreover, it shares its mechanism to adapt CR and F with SHADE using function distributions whose means change based on a memory of previous improvement. This algorithm is going to be used to create new algorithms in future competitions. The second winner was UMOEA [46], a united multiple operator EA that randomly divides the population into three sub-populations. In the first half of the evaluations of the algorithm, each sub-population is improved by a different algorithm (DE, GA, and the CMA-ES), and then the identified best algorithm is applied to all sub-populations. In addition, periodically, the worst from each sub-population are improved using the best solutions. The third one was MVMO-SH [47], a hybrid version of MVMO with an additional improvement method. The proposed MVMO performs the search through a population of particles that are classified into good particles or bad particles: good particles evolve toward the current best, and the bad ones are crossed with a multi-parent crossover to create better particles.

It is remarkable that, DRMA-LSCh-CMA [48], the third in the previous year, was also proposed but it was not placed in the best positions, showing the continuous improvement of real-parameter optimization algorithms year by year.

In the expensive benchmarks, three algorithms were proposed: three of them using a surrogate technique, which allows algorithms to approximate the fitness of solutions without actually evaluating them. However, the winner was another version of MVMO [49], which differs from that proposed for non-expensive benchmark [47] is two aspects: first, there is only one memory with current solutions, not one memory for each particle (and its previous offspring), and the mutation method to create the offspring is completely different.

CEC'2015 Real-Parameter Single-Objective Special Session

In 2015, several competitions were proposed for real-parameter optimization:

- A learning-based benchmark [50]. This benchmark is very similar to previous ones, with 15 functions (unimodal, simple multimodal, hybrid functions, and composition functions) and dimensions 10 and 30. The main difference to previous competitions is the fact that for the first time the participants were allowed to optimize the parameter of their proposed algorithm for each problem, in the search of a highly tunable algorithm.
- A bound constraint single-objective computationally expensive benchmark [51]. In this benchmark, the number of evaluations to tackle the problem is lower than others, from 100 to 1500, in order to study the algorithm with better performance.
- A multimodal benchmark [52], with 15 scalable multimodal functions with four niches. All functions were rotated and shifted to create linkage among different dimensions and to place the optima at different locations.

In this paper, we are going to focus our attention on the learning-based benchmark and expensive benchmark, noting the winners of the competitions. For more details, you can consult the website of the competition[6] and the corresponding papers.

First, we are going to introduce the results obtained with the learning-based benchmarks. In this competition, the clear winner was SPS-L-SHADE-EIG [53] (first in dimensions 10D and 30D, and second in 50D). SPS-L-SHADE-EIG modifies the previous winner L-SHADE [45] replacing the original crossover operator by an eigen-based crossover operator, and uses the *successful-parent-selecting* framework [54] to select the parents differently when stagnation is detected, thus helping avoid the situation. The globally second best algorithm was DEsPA [55], an enhancement of JADE [56] that uses a success-based parameter adaptation with resizing population size. However, it was the best in dimension 50 and ranked average in dimensions 10 and 30. MVMO [57] and LSHADE-ND [58] tie in the third position, being much more robust than DEsPA, thus perhaps of more interest in lower dimensions. The proposed MVMO [57] differs from previous MVMO in that uses an only memory, and evolves a population of candidate solutions, each having their own memory. These solutions are classified into good and bad solutions, and the evolution of each solution depends on its category. LSHADE-ND [58] is an algorithm that iteratively chooses, in an adaptive way, to apply L-SHADE (with some minor changes) or a neuro-dynamic optimization method [59].

In the expensive competition, the results strongly depend on the functions. For example, the first functions are easier but their FEs budget is lower. In the first five functions, only the proposed MVMO [60] (with only one memory) obtained good results. In the following ten functions, there was no clear winner. The results obtained by MVMO were similar to other algorithms using CMA-ES [61, 62], and the proposed PSO algorithm [63]. However, MVMO was the clear winner in the functions with lower FEs (especially in simple functions),

[5] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014/CEC2014.htm

[6] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2015/CEC2015.htm

confirming the good behavior of MVMO in expensive optimization.

CEC'2016 Real-Parameter Special Session

In 2016, there was no new benchmark proposed for CEC'2016, and the authors were encouraged to use benchmarks from previous years.

There were nine proposals for the CEC'2014 benchmark, five for the CEC'2015 benchmark for learning-based algorithms, and six for the CEC'2015 benchmark for expensive techniques.[7]

These results show that CEC'2014 benchmark is of interest, even 2 years later. As expected, benchmarks and results obtained in competitions can be used as a reference to continuously improve an algorithm, so it is normal for more advanced algorithms to be designed using old benchmarks.

In the following sections, we are going to describe results obtained from each of the following benchmarks:

- For the CEC'2014 benchmark, nine algorithms were proposed, with more than half, five, based on SHADE or L-SHADE, the previous winner of that competition. This enforced the idea for winner of previous sessions to be used as references for new proposals. However, the winner was a very different algorithm, UMOEA-II [64], that differing from UMOEA [46] proposed in 2014 in that the GA is not used, a LS method and an improved adaptive selection method are introduced. The second, LSHADE-EpSin [65], uses a new ensemble sinusoidal approach to adapt automatically the values of the scaling factor of the DE. The third, i-LSHADE [66], improves L-SHADE with different default values for M_{CR}, and several fixed values in the L-SHADE parameter memory (H), among other changes. However, although UMOEA-II obtained the best average ranking, its good results are mainly in first dimension values, in the next dimension LSHADE-ESin obtained significant better results.
- Five algorithms were proposed from the learning-based category of the CEC'2015 benchmark. The best one was not actually published in the congress, but as technical report, MVMO-PHM [67], whose main feature is to combine a MVMO with a population model (each individual is a group of solutions) with an improvement method. The second winner was LSHADE44 [68] an implementation of L-SHADE with four strategies which compete among themselves, applying more time which has given the best results. The third was CCLSHADE, which uses the differential grouping decomposition [69] with L-SHADE to optimize each one.
- For the CEC'2015 benchmark in the expensive category, six algorithms were proposed. The winner was MVMO-PHM [67], as described in the previous paragraph. The good performance of the algorithm means it was winner not only in the learning-based benchmark but also in the expensive one. The second best was an ABC algorithm, AsBeC_tuned [70], which modifies the ABC model improving performance: considering distance between solutions to quickly explore different regions, changes in the onlooker solutions, and the addition of a local search, and a systematic global search method. The third was RYYPO [71], a yin-yang pair algorithm [72] with reduces point evolution, which are only evaluated when they follow certain conditions. In the results, we can see a great difference between the aforementioned algorithms and the others (with more similarity between MVMO-PHM and AsBeC_tuned).

To summarize, in CEC'2016, the previously successful algorithms like MVMO and LSHADE are used as a starting point for more sophisticated algorithms that improve results in the different competitions. The different benchmarks in these sessions have enough contributors, so researchers consider them complementary (CEC'2014 and CEC'2015 are different, mainly due to the allowed major flexibility in the CEC'2015 benchmark). In addition, expensive benchmark is more consolidated in the research field.

CEC'2017 Real-Parameter Special Session

In CEC'2017, a new benchmark was proposed for real-parameter optimization.[8] This special session also held a competition where nine papers were presented on and compared to this new benchmark. Additionally, there was one proposal for the CEC'2015 expensive optimization benchmark.

- Regarding the proposals for the real-parameter optimization benchmark, the organizers compared their performance concluding that the best three algorithms were, in this order, jSO [73], LSHADE-cnEpSin [74], and LSHADE_SPACMA [75]. The first algorithm, jSO [73], is an improved version of the i-LSHADE algorithm presented at the CEC'2016 Special Session, in which the main difference is that, within the mutation operator, the F control parameter is multiplied by a factor whose value increases through the execution of the algorithm. The second algorithm, LSHADE-cnEpSin [74], is also an improvement over a previously presented method, LSHADE-EpSin. There are two main differences compared to its predecessor. First, instead of just one

[7] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2016/CEC2016.htm

[8] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2017/CEC2017.htm

sinusoidal, it uses a pool that learns which of the two available sinusoidals to adapt and the F control parameter to use. Second, the crossover operator is applied to the target and trial vectors after translating them into a new coordinate system computed from the covariance matrix of the neighborhood of the best solution. The result of the crossover is then translated back into the original coordinate system. Finally, the third algorithm, LSHADE_SPACMA [75], combines a modified CMA-ES that implements a new crossover operator with a variant of the LSHADE algorithm where the parameter adaptation strategy is also changed.

- On the other hand, the only proposal dealing with the CEC'2015 benchmark on expensive optimization introduces a new social algorithm called Higher Order Cognitive Optimization (HOCO) [76], which uses several meta-cognitive components (personal and social) to attain the global optimum.

As can be seen, the two best algorithms in the 2017 competition are evolved versions of previously proposed methods that also obtained very competitive results in previous special sessions. Furthermore, in this case, the best algorithm is not an updated version of the best one from the previous year, which means that these competitions encourage researchers to continuously improve their proposals, especially when the differences in the performance of some of them are not very large.

BBOB Competition

Since the CEC'2005 real-parameter special sessions, there were no other real-parameter competitions (without constraints) until 2009. Since 2009, a real-parameter special session in the *Genetic and Evolutionary Computation Conference*, GECCO, was proposed. In this special session, the benchmark called Black-Box Optimization Benchmark, BBOB, was proposed, which has been evolving over the past few years.

The main differences between the BBOB benchmark and the IEEE CEC's benchmarks are the following: BBOB has two categories: noisy functions and noiseless functions, and there are functions with different dimension values. More information about the BBOB benchmark can be obtained in [77] and the software and other useful information are available at https://github.com/numbbo/coco.

In this section, we will describe the competitions that have been organized in the GECCO using the Black-Box benchmark, mentioning algorithms that we consider more interesting, and several conclusions, as in the previous session. However, differing to CEC special sessions, BBOB Competitions do not announce a ranking of winning algorithms, so we are going to mention the algorithms that we consider more interesting, considering both their good results and their influence. The number of total competitors into the BBOB competitions is high, about 150,[9] but several works include different variations of algorithms.

Black-Box Optimization Benchmarking (BBOB) 2009

The competition was well-received by the researchers, and it received thirty proposals[10](with several authors proposing several proposals each). From the published analysis [78] of the authors, the winner was BI-POP-CMAES [79], a CMA-ES that combines two different restart mechanisms, one with an increasing population size. The second, with very similar in results, was AMaLGaM IDEA [80], another EDA with normal distribution that uses maximum–likelihood estimates for the mean and an covariance matrix that scales up to prevent premature convergence on slopes.

Black-Box Optimization Benchmarking (BBOB) 2010

In the following year, 2010, there were 31 proposal,[11] although several of them had very similar algorithms.

Out of all the algorithms, IPOP-aCMAES [37] was the most scalable, obtaining very good results from dimension 10 to 20 (and also in more difficult functions with lower dimension). IPOP-aCMAES adds to the increasing population size of [25] the consideration not only the best solutions to guide the search but also the worsts. For dimension 40, however, CMA+DE+MOS [81], a MOS algorithm using CMA-ES, was able to obtain more accurate solutions in most functions (as expected considering the good results of MOS [82] in higher dimensionality)

Black-Box Optimization Benchmarking (BBOB) 2012

In the next competition in 2012, there were also many proposal, 29 proposals in total,[12] some of them very similar to each other.

For lower dimensionality, the most interesting (considering results and robustness) is IPOP-saACM [83, 84]. IPOP-SaACM is an algorithm based on IPOP-CMAES that approximates the fitness of solutions using a surrogate technique, $\hat{f}$: During $\hat{n}$ iterations, the CMA-ES is run using $\hat{f}$, and then the last λ are evaluated with the real *fitness* function. The parameter $\hat{n}$ and the calculation of $\hat{f}$ are self-adaptive to reduce the surrogate error (using an archive of solutions and their *fitness*).

NBIPOP-aCMAES [85] is another algorithm with similar results, which stands out in higher dimensions and noisy benchmarks. NBIPOP-aCMAES is an improved version of

[9] http://coco.gforge.inria.fr/doku.php?id=algorithms
[10] http://coco.gforge.inria.fr/doku.php?id=bbob-2009-algorithms
[11] http://coco.gforge.inria.fr/doku.php?id=bbob-2010
[12] http://coco.gforge.inria.fr/doku.php?id=bbob-2012

BIPOP-aCMAES [37] that differs in reducing the step size of the CMA-ES after each restart.

Black-Box Optimization Benchmarking (BBOB) 2013

In the next competition in 2013, there were 30 proposals,[13] with less similarity among the proposals than in previous competitions.

From the obtained results, HCMA [86] and HMLSL [87] obtained the best results and rather robust behavior. HCMA is a combination of two new algorithms, BIPOP-saMCA-ES-k, a version of BIPOP-saCMA from the previous year with two populations and a bigger population size during the surrogate, and BIPOP-aCMA-STEP (hybridization of BIPOP-aCMAES with a STEP algorithm [88]). Also, BIPOP-saMCA-ES-k in isolation obtains good results, especially in higher dimension values. HMLSL is a hybridization of a Multi-Level Single Linkage (MLSL) with a DE.

Black-Box Optimization Benchmarking (BBOB) 2015

In 2015, two competitions used BBOB, one in the GECCO, where traditionally they were organized, and another in the CEC.

In the BBOB competition, in 2015, there were 15 proposals,[14] from 6 different papers, and in the CEC competition 8 proposals from only 3 different papers.

From the BBOB’2015 in the CEC, there are interesting proposals like CMA-TPA and CMA-MSR (both defined in [89]), which change step size adaptation of CMA-ES, but achieve slightly better results than the original.

From the BBOB’2015 in the GECCO, the most interesting results (both in single and expensive competition) came from R-LSHADE [90], a restart version of state-of-art L-SHADE, obtaining a rather good robust behavior, but could not improve the results obtained by HCMA.

Black-Box Optimization Benchmarking (BBOB) 2016

In 2016, the BBOB workshop mainly focused on multi-objective optimization.[15] A new bi-objective testbed was proposed and most of the contributions evaluated their proposals in that benchmark. As this is out of the scope of this paper, we focus on the only contribution that considered the real-parameter testbeds. In particular, the authors [91, 92] proposed a new strategy to adapt the population size in CMA-ES and evaluated several configurations of this strategy on both the noiseless and the noisy testbeds, concluding that it has a similar performance to that of the best algorithm of BBOB 2009 on well-structured functions, but still inferior results in the remaining groups of functions.

Black-Box Optimization Benchmarking (BBOB) 2017

The most recent workshop of 2017 counted with participants presenting contributions in both benchmarks (real-parameter and multi-objective optimization).[16] Again, we focus on the first benchmark as the second one is out of the scope of this paper. Four of the five contributions related to the real-parameter testbed involved the CMA-ES algorithm to some extent (either modified CMA-ES versions [93–95] or combinations of CMA-ES with other algorithms [96]). This gives an idea of the preponderance of CMA-ES in this benchmark. However, as it was the case of the 2016 edition, none of the contributions was able to improve the results of the best algorithm of the 2009 session. Finally, the remaining contribution presented an architecture for the distributed asynchronous evaluation of a pool of solutions with different heterogeneous workers (GAs and PSO, in this case) [97].

Black-Box Optimization Competition

In this section, we are going to describe the results obtained in new black-box competitions available online, called BBComp.[17] These similar names should not be confused, and this new model of competition has several important differences as compared to the previous models:

- It is *actually* a black-box competition, not only can the algorithm not use any information about the benchmark but the researchers also must not have any information at all about the problems used in the competition: neither their mathematical expressions nor their properties. The only information given to the optimizer and participant is the dimension of the problem, bounds on all variables, and a pre-defined budget of black box evaluations.
- Researchers do not have access to the code, and all the evaluations are carried out in an external server belonging to the organizers, so there is a library which allows the algorithms to send their solutions to be evaluated through the Internet.
- There is a testing track available to make all algorithms tests, and then during each competition, the related track is available. The experiments on each algorithm can only be done once in the competition.
- They are sometimes problems with only one objective and problems with several objectives in the same competition/track (but not it is common). The ranking is done considering the error in problems with one objective and the

[13] http://coco.gforge.inria.fr/doku.php?id=bbob-2013
[14] http://coco.gforge.inria.fr/doku.php?id=bbob-2015
[15] https://numbbo.github.io/workshops/BBOB-2016/
[16] https://numbbo.github.io/workshops/BBOB-2017/
[17] http://bbcomp.ini.rub.de/index.html

hypercube in multi-objective problems and then each algorithm receives points according to its ranking for each function.

In the following, we are going to indicate the results of the different competitions using this new methodology. Because the competitions are not related to any special session, there are also commercial or unpublished algorithms, so sometimes the description of the algorithm is not possible.

GECCO'2015 Track

The GECCO 2015 track consisted of 1000 black box problems with different dimensions, from 2 to 64. The maximum evaluation number was between $10 \times dim - 100 \times dim$.

There were 28 participants,[18] from algorithms published to optimization software. The winner was KNITRO [98] version 10.1, a commercial software for non-lineal optimization from the company Artelys,[19] with a parallel multi-start method using several interior-point or active set algorithms. MVMO [57] was the second winner, coming in very close, and also came the third in the learning-based benchmark (see "CEC'2015 Real-Parameter Single-Objective Special Session" section).[20] The third algorithm was NSMO [99], which is based on a hierarchical partitioning of the domain search in nodes, and approximates the gradient in each dimension to estimate the potential improvement in *fitness* to decide when to expand a node, with source code freely available.[21]

CEC'2015 Track

The CEC 2015 track consisted of 1000 black box problems with different dimensions, from 2 to 64. The maximum evaluation number is $100 \times dim^2$. There were 24 participants,[22] with very different algorithms.

The winner was UMOEA [46], presented in CEC'2014 real-parameter numerical optimization, with source code available,[23] you can see "CEC'2014 Single Objective Real-Parameter Special Session" section for a description of the algorithm. The second winner was a two-stage algorithm [100], an algorithm that decides which two-stage variant to apply on which dimension: restarted local search and clustering methods, and the local search algorithms Nelder-Mead and CMA-ES. The source code is available.[24] Curiously, KNITRO [98], the winner in the GECCO'2015 Track, only ranked 11th.

[18] see http://bbcomp.ini.rub.de/results/BBComp2015GECCO/summary.html
[19] https://www.artelys.com/en/optimization-tools/knitro
[20] available at https://www.uni-due.de/mvmo/download
[21] at https://github.com/ash-aldujaili/NMSO
[22] see http://bbcomp.ini.rub.de/results/BBComp2015CEC/summary.html
[23] at http://bbcomp.ini.rub.de/results/BBComp2015CEC/mickey.zip
[24] at http://ls11-www.cs.uni-dortmund.de/staff/wessing/bbcomp

GECCO'2016 Tracks

In GECCO'2016, the organizers proposed five tracks, consisting of 1000 black-box problems with different dimensions, from 2 to 64. The differences among these tracks are the budget of fitness evaluations and the number of objectives of the problems.

- Single-objective track. In this track, problems consist of only one objective, and the number of fitness evaluations allowed was $100 \times dim^2$. There were 14 participants and the winning algorithm was a new version of the second winner at the CEC'2015 Track [100]. In this new version, among other changes, 10% of the evaluations are devoted to local optimization of current best solution, while the clustering method was also improved. The source code and description can be located at the author's website. There is no information about the second and third winners. The fourth best algorithm was a curved trajectories algorithm [101], and KNITRO [98] was the fifth best algorithm.
- Expensive single-objective track. In this track, the number of allowed fitness evaluations has been highly reduced, ranging from $10 \times dim - 100 \times dim$. There were 15 participants. The winner was the aforementioned KNITRO algorithm. The second winner obtained very similar results to the first one but, unfortunately, there is no information about that algorithm. The third one was a version of the single-objective winner, with a difference in the use of the local search, and that it includes in the archive all the sampled points.
- Several multi-objective tracks (two-objective, expensive two-objective, and three-objective). We are not going to describe them here, because we have not covered multi-objective benchmarks in this paper.

GECCO'2017 Tracks

In GECCO'2017, similar to the previous year, five different tracks were proposed, consisted of 1000 black box problems with different dimensions, from 2 to 64. The differences, as in the previous edition, are the budget of fitness evaluations and the number of objectives.

- Single-objective track, with only one objective, and the maximum number of fitness evaluations equal to $100 \times dim^2$. There were 13 participants. The considered winning algorithm did not obtained the best ratio of solved problems, especially in dimension 64, in which it was only able to obtain the optimum in less than 50% of the problems. Unfortunately, there is no information about the winning algorithm, neither a description nor a reference. The second winning algorithm was "Doubly trained surrogate CMA-ES", DTS-CMA-ES, with a self-adaptation of the population size per generation (and a fallback to the original IPOP-CMA-ES after 4 days of computation). The third winner was the previous mentioned two-stage algorithm.

- Expensive single-objective track. In this track, the number of allowed fitness evaluations is much smaller, ranging from $10{\cdot}dim - 100{\cdot}dim$. There were 13 participants. The winner was a combination of DTS-CMA-ES and BOBYQAP (it initially applies DTS-CMA-ES and, for the last evaluations, BOBYQAP). The second winner was a restarted model-based optimization with L-BFGS-B and a random search (see[25]). The third one was the KNITRO software.
- Several multi-objective tracks (two-objective, expensive two-objective, and three-objective). As in the previous section, we are not going to describe them here, because we have not covered multi-objective benchmarks in this paper.

EMO'2017 Track

In the 9th International Conference on Evolutionary Multi-Criterion Optimization, EMO'2017, a real-world problems track was proposed. In this benchmark, the ten participants were ranked based on the overall dominated hypervolume.

The winner was Model-based HV-maximization (see[25]) that combines a HV-maximization with the SMS-EMOEA [102] algorithm. The second winner was the Bayesian Multi-Objective Optimization [103] algorithm, which uses a Monte Carlo algorithm with a new extended domination rule. The third winner was PADDS-CHC but, unfortunately, there is no information or reference about it.

Evolution of Real-Parameter Optimization: Lessons Learnt

In this section, we are going to indicate several lessons that we have learnt after reviewing the competitions and their winners:

- First, although there are several competitions taking place (IEEE CEC and BBOB, mainly) there hardly influence each other. It is true that these competitions are organized in different conferences on similar dates, but it is curious that advanced versions of the previous winner in one congress usually continue participating in that same conference (IEEE or GECCO) for the next few years, but never in the other. Thus, the influence of one algorithm is unfortunately limited to the conference in which it was proposed.
- Since the beginning of the IEEE CEC competitions, a great majority of the proposals have been devoted to DE, due to their good results in these benchmarks (with the exception of the CEC'2011 competition). As consequence of this, DE-based methods have clearly evolved during the last few years to become more advanced algorithms such as L-SHADE [45], or derived methods such as L-SHADE-EpSin [65]. However, in recent competitions, non-DE algorithms have obtained very good results. For example, UMOEA-II [64] was the winner of the CEC and the Black-Box competition in 2016. Furthermore, other recent methods such as those of the MVMO family [47, 57] have obtained very good results in several competitions against multiple algorithms.
- In the BBOB competitions, most of the participating methods are versions of the CMA-ES [24] algorithm. This is partially due to the low dimension values used in the first versions of the benchmark. As a result of this, 22 variants of the CMA-ES algorithm have been proposed in the last few years, including the more robust version known as IPOP-aCMAES [85], which reports superior performance by also using the worst solutions to improve the search. Additionally, the incorporation of surrogate models has also even lead to improved results. The algorithm that could be considered the current best algorithm is HCMA [86], a combination of an algorithm using surrogate models with another more exploitation-oriented method (although both are based on the CMA-ES algorithm).
- In the BBComp competition, it is not possible to observe any evolution as the black-box model implies that for each competition the functions could be completely different. For this reason, we want only to note that the proposed algorithms, such as UMOEA [46], can improve more consolidate algorithms, such as KNITRO [98].

Constraint Real-Parameter Optimization

A general constraint minimization problem is a minimization problem in which the aim is to minimize $f(x)$ subject to the following:

$$g_i(x) = c_i \text{ for i} = 1, \ldots, \text{n}$$
$$h_j(x) \geq d_j \text{ for j} = 1, \ldots, \text{m}$$

Constraint optimization is very important because problems with constraints appear very frequently in real problems. Due to this popularity, several competitions have been proposed over the past few years.

[25] at http://ls11-www.cs.uni-dortmund.de/staff/wessing/bbcomp

The following sections are devoted to these constraint optimization special sessions, mainly held at the IEEE CEC conference, for which we highlight the algorithm with the best results, as in the previous section. There is extensive literature on these kinds of problems. However, as the present contribution is focused on the evolution of the field through competitions, we will not provide a complete review of the literature in this section. We refer the reader to [104] for a detailed survey on this topic.

CEC'2006 Constraint Optimization

In 2006, a benchmark specially designed for constraint optimization was proposed, and in the competition, 11 algorithms were proposed, 9 of them DEs.[26]

In the organizer analysis, the winners were *E*DE [105], a DE with a new *E* constrained method to sort the solutions based on their fitness and a measure of their feasibility. *E*DE uses a gradient-based method to find feasible solutions using the gradient of constraints in an infeasible solution. This *E*DE obtains very good results but it strongly depends on the *E* parameter. The second winner was DMS-PSO (similar to DMS-L-PSO of previous year) and jDE-2 [106] algorithm, a constraint implementation of jDE using a penalty to tackle the constraints and an improved adaptive mechanism for parameters F and CR.

CEC'2010 Constraint Optimization

In this special session, another constraint real-optimization benchmark was proposed [107], 18 functions with different unequal and equal constraints for function, and run 25 times each for two dimension values: 10 and 20, using a different MaxEvals for each dimension.

In this special session, 12 algorithms were proposed[27]: 7 DEs, 1 ABC, 1 GA, 1 PSO, a constraint version of MTS algorithm [108] described in "Large-Scale Optimization" section, and 2 other algorithms. Again, half of the proposals are DEs, mainly due to their good results (three of the four best algorithms were DEs).

From the analysis of organizers, the winner was εDEg [109], a modified version of the previous winner, εDE, with three important changes to improve results. It uses an archive to maintain more diversity; it assures that children are not worse than their parents, and it uses an adaptive mechanism to adapt its ε value (critical for the good behavior of εDE). The second winner was ECHT [110], a DE algorithm that uses four constraint methods in different populations, so the offspring obtained by one constraint method compete with the ones. These two algorithms are clearly the winners, while there are many algorithms in a possible third position.

Multimodal Optimization

Sometimes, in optimization problems, the aim is not to obtain only one global optimum, but all possible optima. This is called multimodal optimization. Finding all optima implies changes in the algorithms, usually incorporating *niching* techniques. Recently, several competitions have been proposed for multimodal optimization, obtaining interesting proposals.

In this section, we are going to show the different competitions and results for multimodal optimization, highlighting the algorithms with the best results. To get all the data, including the results of each competition, you can check the organizers' site.[28]

All these competitions use the same benchmark and the winners of one competition are used as reference algorithms in following competitions. This means that the winners of one competition could be considered better than all previous proposals under the compared benchmark.

Analogously to the constraint optimization section, it is not the aim of this paper to provide a full review of the literature, rather the evolution of the area through the different competitions. We recommend our readers to refer to [111] for an extensive survey on this issue.

CEC'2013 Niching Methods for Multimodal Optimization

In 2013, a new benchmark specially designed for multimodal modal optimization was presented [112]. This benchmark is composed of 12 functions with different numbers of optima and several dimension values (giving a total of 20 problems to optimize). For each problem, the ratio of the found optima is used for successful measure. These ratios are calculated for five threshold levels: 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}.

In the competition, 9 papers were presented with 12 proposals, 7 DE algorithms, and several niching versions of well-known algorithms, like CMA-ES, VMO, or NSGA-II.

From the organizers results, the winner was NEA2, a previously proposed algorithm [113]. NEA2 is an algorithm that iteratively applies the CMA-ES with a Nearest-Better Clustering Algorithm (NBC) [114]. NBC clusters the solutions obtained by previous CMA-ES, and then CMA-ES is applied to generate one population for each detected cluster. It is a remarkable algorithm, because it is still a very competitive algorithm. The second was dADE/nrand/1 [115], a DE that enforces diversity generating the solutions around the nearest neighborhood, and self-adapts its parameters as the

[26] http://www.ntu.edu.sg/home/epnsugan/index_files/CEC-06/CEC06.htm
[27] http://www3.ntu.edu.sg/home/EPNSugan/index_files/CEC10-Const/CEC10-Const.htm
[28] https://github.com/mikeagn/CEC2013/

JADE algorithm. Also, it uses an archive to store new inserted solutions and when the same individual is generated again, it is re-initialized. The third was the well-known CMA-ES with an archive (data given by the organizers) and the fourth (third proposal) was N-VMO [116], an algorithm that uses a mesh of solutions with different operators to center them around the local optima.

CEC'2015 Niching Methods for Multimodal Optimization

In 2015, another competition was proposed, also using the benchmark proposed in CEC'2013. From the information of the organizers, four algorithms were submitted and they were also compared to four algorithms from CEC'2013 (including the winners NEA2 and dADE/nrand/1).

The winner was the Niching Migratory Multi-Swarm Optimiser, NMMSO [117], a multi-swarm PSO with two important features: first, the number of swarms is dynamic, merging two swarms when they are too close, or when the mid-point between them is better than the best solution of each of them. Second, the particles in each swarm increase (with a limit, when it is achieved the worst are removed). The second winner was NEA2 [113], explained in previous section. The third (although it improves NEA2 in average ranking in the paper) was LSEDA [117], which applies a dynamic mechanism to define niches in a similar way to NMMSO and for each niche a localized search is carried out using a different EA (limited in its exploration to the immediate neighborhood), and with an additional search oriented to top solutions. NMMSO and LSEDA code are freely available.[29]

CEC'2016 Niching Methods for Multimodal Optimization

In 2016, another competition was proposed with the same benchmark, both in the CEC'2016 Congress as in the GECCO'2016 Conference. Four new proposals were submitted and they included in the comparisons five algorithms from previous competitions (including the previous winners NMMSO, NEA2 and dADE/nrand/1).[30]

The winner was RS-CMSA [118], an algorithm that explores evolving several populations using CMSA [119], a version of CMA-ES, including elitism and enforcing diversity using taboo points. It considers a taboo point to be the mean of better sub-populations and the found local optima, and it avoids solutions that are too close to these points (using an adaptive taboo distance).

The second winner was NMMSO [120], from last year, and the third winner was an improved version of NEA2, NEA2+, from Mike Preuss. The next winning proposal was RLSIS: Restarted Local Search with Improved Selection of Starting Points, from Simon Wessing. The last two algorithms are still in press, so there is no reference available yet.

Large-Scale Optimization

As shown in the previous section, Evolutionary Algorithms are a very popular tool in the field of real coding optimization. This has encouraged researchers from many different disciplines to use these algorithms in very different contexts, both in the industrial and the scientific domain. However, some of these applications can be quite challenging: complex models need to simultaneously optimize hundreds, if not thousands, of real parameters. For this reason, some state-of-the-art algorithms with an outstanding performance in real coding optimization problems of moderate size cannot deal with their large-scale counterparts [25] (curse of dimensionality [121]). In this context, Large-Scale Global Optimization Special Sessions held during the last 8 years have played an important role in designing powerful scalable EAs that are able to face these new kinds of problems successfully.

To the best of our knowledge, the first special session of this kind was held back in 2008 at the IEEE Congress of Evolutionary Computation (CEC'2008) [122]. Since then, a session has been held on this topic every year at CEC, except in years 2009 and 2011. As well as this session, recently, there has also been another remarkable event: a special issue in the Soft Computing journal in 2011 [123]. In the following subsections, we review each of these events, paying special attention to how the results in previous editions influenced the development of new algorithms.

First Steps

In the initial competitions, there was no clear winner; the proposals in each competition improved previous winners.

CEC'2008 Special Session

In this special session, a benchmark of 7 scalable continuous optimization functions was proposed. Among these functions, two of them were unimodal whereas five of them were multimodal. On the other hand, three of them were separable (they could be easily solved optimizing each dimension individually) whereas the other four functions were non-separable.

Eight different entries were submitted to this special session and their results were compared in the proposed benchmark. Given that the functions are scalable, results were provided for 100, 500, and 1000 dimensions. However, we are mostly interested in large-scale problems, and therefore, we have focused on the 1000D results.

[29] https://github.com/fieldsend

[30] http://goanna.cs.rmit.edu.au/~xiaodong/cec16-niching/

The results reported in [124] reveal several interesting things:

- The first three positions of the ranking belong to algorithms of very different nature: a combination of local searches (MTS) [108], a variation of an Estimation of Distribution Algorithm LSEDA-gl [125] and a Self-Adaptive Differential Evolution algorithm with decreasing population size (jDEdynNP-F) [126].
- The algorithm that reports the best results, MTS, is a combination of three linear local searches. Good results for this algorithm can be expected in separable functions, which it obtains. However, it also gets very competitive results in non-separable functions (best results in F_3 and F_5 and second best results in F_2 and F_7).
- Algorithms based on Particle Swarm Optimization [127, 128] and Evolutionary Programming [129] do not get very good results, even if they are improved or combined with other algorithms.
- Dimension decomposition methods [130, 131] provide intermediate results.

In the light of these results, many researchers realized that, even in large-scale problems, local searches were able to significantly improve the results of EAs and decided to incorporate them into their algorithms, as we will see in the following subsections.

CEC'2010 Special Session

The benchmark proposed for the CEC'2010 Special Session [132] is an improved version of the one proposed for the CEC'2008. First, the number of functions has been increased by up to 20. Second, it incorporates functions with different degrees of separability by combining fully separable and non-separable functions.

In this special session, the algorithm with the best results (MA-SW-Chains) [133] combined a genetic algorithm with a local search. The following special sessions will confirm this trend: the combination of a population-based method with one (or more) local search attains the best overall results. On the other hand, as in the previous edition of the LSGO Special Session, the remaining top algorithms were also of a very different nature: a two-stage algorithm called EOEA [134] that combines one first exploratory step carried out with a modified EDA and a second intensification phase in which a cooperative co-evolution with three different sub-optimizers is applied, and an ACO-based algorithm (DASA) [135] that transforms a continuous problem into a graph-search problem.

The remaining contestants in the competition were three DE-based algorithms (one of them also including cooperative co-evolution) [136–138] and a PSO-based algorithm [139].

A comparative analysis of the three best algorithms draws the following conclusions:

- MA-SW-Chains gets systematically better results than the other two methods, except in the case of fully separable functions. This seems to be coherent with the characteristics of the three algorithms as it would probably take longer for the hybrid MA-SW-Chains algorithm to reach the global optimum for this kind of problem whereas problem decomposition in the case of EOEA can make it converge much faster.
- The relative difference in the performance of the three methods increases as the non-separability degree of the functions gets larger. However, the performance of the DASA algorithm gets an unexpected boost in the case of non-separable functions, which is hard to explain.

Regarding the other algorithms, there is no clear pattern that can help isolate the convenience of using one method or another for a particular type of problem: some algorithms obtain (relatively) good results in both fully separable and non-separable functions whereas their performance degrades for intermediate degrees of separability.

Differential Evolution Emerges

The Soft Computing special issue published in 2011 shows how differential evolution algorithms were consolidated as the most adequate algorithms for large-scale global optimization.

The Soft Computing Special Issue on *scalability of evolutionary algorithms and other metaheuristics for large-scale global optimization problems* [123] was a very successful event with a high participation ratio (up to 13 contributions were submitted to the special issue). This special issue was motivated by a previous special session held at the 2009 Intelligent Systems Design and Applications conference (ISDA'2009). As many of the proposals of the special issue are improved versions of the algorithms presented during the conference, we decided to consider only the latter in our review. It is interesting to note that more than half of these contributions proposed algorithms incorporating Differential Evolution to some degree [82, 140–145]. There were also two methods based on Particle Swarm Optimization [146, 147], a Path Relinking algorithm [148], a Line Search [149], a memetic algorithm based on a steady-state GA and a local search [150], a derivative-free unconstrained optimization method based on QR factorizations [151], and a Variable Mesh Optimization algorithm [152].

The benchmark proposed for this special issue was made up of 19 continuous scalable functions and participants were requested to provide results in 50, 100, 200, 500, and 1000 dimensions. As in the case of the CEC'2008 Special Session, we focus on high dimensional (i.e., 1000D) problems.

In the light of the results reported by each of the participants in the special issue, we can observe the following behavior:

- Differential Evolution seems to perform especially well in this benchmark. The three best algorithms (in this order): MOS [82], GaDE [144], and jDElscop [140] used Differential Evolution, although in different ways: MOS combined it with a strong local search (MTS-LS1), GaDE included self-adaptation of its parameter, and jDElscop used three different strategies and population size reduction.
- Small population sizes seem to be sufficient for these problems. In the case of MOS, a population of just 15 individuals was used. On the other hand, GaDE used a population size of 60 solutions, which is quite small for 1000D problems. Finally, jDElscop considered an initial population size of 100 individuals, which is reduced as the algorithm evolves.
- Both algorithms using Particle Swarm Optimization also obtained remarkable results, coming fourth [145] and fifth [147] in the ranking.
- Non-DE and non-PSO obtained the worst results in this benchmark. It seems that the benchmark might be somehow easy to solve when using these methods as opposed to other approaches. A more in depth analysis would be needed to confirm this hypothesis.

The Kingdom of MOS Algorithm

In the following competitions, MOS is still the best algorithm for large-scale global optimization.

CEC'2012 Special Session

The CEC'2012 Special Session continued using the CEC'2010 benchmark of functions. This allowed a broader comparison of algorithms by incorporating the results of CEC'2010 into those of the 2012 edition.

Five papers were finally accepted in the special session, among which a MOS-based algorithm combining two local searches [153] obtained the best overall results. The second place was obtained by a DE-based algorithm with multiple operators and a small varying population size (jDEsps) [154], whereas the third place belonged to a new version of the two-phase algorithm presented in the previous edition (CEC'2010), in which exploration and intensification phases are exchanged and the algorithms used to optimize the sub-components are also different (CCGS) [155]. Regarding the other two entries, one of them was based on DE [156] and the other one was a memetic Artificial Bee Colony algorithm [157].

It may be surprising that the best overall results were obtained by an algorithm with no global search component. This could be due to the inherent difficulty of the benchmark functions (only five of them could be solved to the maximum precision by any algorithm, and not all of them for the same one) that can be making it impossible for algorithms to escape from local optima. This also would explain the good behavior of the jDEsps algorithm, which uses a small population size, thus favoring intensification of the search.

An overall comparison including both algorithms from CEC'2010 and 2012 sessions reveals that the best three algorithms, in this order, were MOS, jDEsps, and MA-SW-Chains. As can be seen, the two first positions correspond to newly proposed algorithms, whereas the third one belongs to the best algorithm of the CEC'2010 special session. This seems to indicate an overall improvement in the performance of new algorithms that is confirmed if we compare the average performance of the algorithms in each session. This means that LSGO special sessions not only favor the creation of new algorithms but also of more powerful ones. This issue will be further discussed in "Evolution of Large-Scale Global Optimizers: Lessons Learnt" section.

CEC'2013 Special Session

For this special session, a new benchmark was proposed [158] that extends the previous one to include some missing challenging characteristics of actual large-scale problems such as non-uniformity in the size of subcomponents, imbalance in the contribution of subcomponents, overlapping of subcomponents, and transformation of the base functions to make them more complex (ill-conditioning, symmetry breaking, or irregularities). This new benchmark constitutes a new challenge for large-scale optimizers as it is specifically designed to be deceptive for most state-of-the-art techniques.

In this special session, only two entries with associated papers were received: a MOS-based hybrid algorithm combining a population-based algorithm (GA) with two local searches (MTS-LS1-Reduced and Solis-Wets) [159] and a cooperative co-evolution method with smoothing and auxiliary functions [160]. Nevertheless, other three approaches were included in the comparison: two other cooperative co-evolution methods (DECC-G [161] and CC-CMA-ES [162]) and a combination of DE and a Variable Mesh Optimization method (VMODE) [163].

Regarding the results of this competition, the following conclusions have been drawn:

- Three out of five algorithms used cooperative co-evolution methods. This is interesting because, as the size of the problems grows, efficient decomposition mechanisms will be needed to be able to tackle these new scenarios.

- The other two methods were hybrid algorithms combining two or more search techniques trying to exploit the benefits of different search strategies.
- The best results were obtained by one of these hybrid algorithms, the MOS-based one, whereas the other hybrid method finished second to last. This means that, to successfully exploit the benefits of multiple algorithms, a direct combination of them is not enough: this hybridization has to be able to detect when one method or another must be used more intensely.
- Two of the cooperative co-evolution methods obtained competitive results, although not as good as MOS. However, to some extent one of the local searches used in MOS (MTS-LS1-Reduced) was conducting a decomposition of the problem as it automatically identified and exploited the variables that maximized the reward in the fitness function.

CEC'2014 Special Session

In 2014, there were no competitions held on LSGO at CEC. Instead, several of the authors who submitted papers to this special session used existing benchmarks from previous competitions to evaluate their algorithms and compare their results. In particular, two of them [164, 165] used the CEC'2010 benchmark, whereas another one [166] used the CEC'2013 benchmark. However, none of these algorithms was able to outperform the existing best results from previous competitions. In the case of the algorithms evaluated in the CEC'2010 benchmark, both methods explored the idea of co-evolution. The first one [164] proposed a new decomposition method specific for LSGO that tries to identify the interactions between variables to define subcomponents. The other one [165] proposes a hybrid approach to alternate between different algorithms in the optimization of the subcomponents. However, as stated before, none of these methods was able to deliver competitive results when compared with the best algorithms for this benchmark [133, 154, 159].

Results were much better for the algorithm that was evaluated in the CEC'2013 benchmark [166] than those from the other aforementioned papers, being competitive to some extent with those of the best algorithm for this benchmark [159]. This proposal considers a well-known Differential Evolution-based algorithm, SaNSDE, which is applied to solutions where separable and non-separable sets of variables have been previously identified.

From these results, we can conclude that:

- Cooperative co-evaluation and problem decomposition seem to be very popular approaches to LSGO problems.
- However, we can observe some degree of stagnation in the results for the algorithms. This trend seems to be confirmed, as will be shown in the following sections.

CEC'2015 Special Session

The LSGO competition at the IEEE CEC was repeated in the 2015 special session. The benchmark used this year was that same as in 2013 and there were two new contributions, as well as the five entries that took part in the previous competition, reviewed in "CEC'2013 Special Session" section. These two new methods include a memetic algorithm that combines DE with a local search in an iterated way [167] and a hybridization of DE and PSO [168].

The results of this competition confirm the trend identified in the previous year: among the four best algorithms, there is only one new method: IHDELS [167] in the second place. The best algorithm for this benchmark continues to be the MOS-based hybrid [159] with CC-CMA-ES and DECC-G in third and fourth places, respectively. Moreover, the two new algorithms also confirm that research in LSGO is basically focusing on two directions: hybrid approaches and cooperative co-evolution algorithms (with decomposition methods).

CEC'2016 Special Session

This special session did not hold a companion competition. With this in mind, some authors [169, 170] proposed new methods that were evaluated on the benchmarks proposed in previous editions of the special session. Some of these methods did not consider the entire benchmarks but focused instead on a subset of functions [171, 172] sharing a common characteristic that they wanted to study independently. Finally, other authors started to explore methods to solve extremely large problems (up to 100 million variables) [173]. We will briefly review these contributions in the following paragraphs.

In [169], the authors propose a new improved function decomposition method to detect interactions among variables in the context of a cooperative co-evolution algorithm. The results obtained with this new approach were evaluated in the CEC'2010 benchmark reporting better results than other state-of-the-art cooperative co-evolutionary algorithms such as DECC-G [161], but far from the results reported by the best algorithms for this benchmark, as reported in "CEC'2010 Special Session" and "CEC'2012 Special Session" sections. On the other hand, Salcedo-Sanz et al. [170] proposed a new version of the Coral Reefs Optimization (CRO) algorithm that extends the original method to deal with LSGO problems by incorporating several substrate layers (search operators) which establish a competitive co-evolution process. It also incorporates a LS for better performance. It was evaluated in the CEC'2013 benchmark, and although it might be competitive with some other approaches such as DECC-G, it is also far from the best results reported by [159].

Omidvar et al. [172] proposed an improvement to an existing Contribution-Based Cooperative Co-evolution algorithm (CBCC) to better manage the exploration/exploitation ratio and evaluated it in a subset of the CEC'2013 benchmark. Mahdavi et al. [171] on the other hand proposed several initialization methods around the center of the search space trying to improve the performance of a CC method on non-separable functions (and thus, its evaluation was only conducted in the corresponding subset of the CEC'2013 benchmark).

Finally, in [173], the authors envision the future of LSGO by modifying a well-known LSGO algorithm (MA-SW-Chains) to be run on GPUs and thus were able to solve problems of up to 100 million variables. The results for this extremely large dimensionality are unprecedented in the literature, and for this reason, they compared their results to those of a random search to analyze to which extent the algorithm was actually progressing in the search, concluding that the results were significantly better in most of the cases according to the statistical tests conducted.

The results of this special session confirmed both observations made in the previous edition: cooperative co-evolution and hybridization seem to be the two most important current research lines and new results do not seem to improve those of reference algorithms.

CEC'2015 Big Optimization Special Session

In 2015, a Big Data Competition[31] was carried out and a new benchmark for large-scale global optimization was proposed [174].

This benchmark has a multi-objective version and a single objective version (combining linearly the results of two fitness functions). In both of them, the aim is to solve a big electroencephalography data optimization problem with 1024, 3072, and 4864 variables. This is a very interesting proposal because it is a real-world problem, with noisy and noiseless versions, and with an increasing number of variables.

Unfortunately, although several methods were proposed [174–176], there were not enough to run a real competition. Among these few proposals, a multi-agent genetic algorithm (MAGA) [175] obtained the best results.

Evolution of Large-Scale Global Optimizers: Lessons Learnt

After almost 10 years of special sessions and issues, the field of continuous LSGO seems to be mature enough to identify some trends and well-defined research lines. In this section, we summarize our findings after a careful review of the results reported over these years:

- In our opinion, there seem to be two different (and maybe complementary) research lines. The first one is the hybridization of multiple methods. The algorithms obtaining the best results in all the special sessions for all the benchmarks combined different search strategies within the same algorithm. Furthermore, all these methods included some kind of control mechanism to orchestrate the combination of the composing methods to get the most out of each of them. The second one is cooperative co-evolution. Many of the approaches recently proposed implement these kinds of methods. Though their performance has not been, in general, comparable to that of the best hybrid methods, they have the advantage of being potentially more scalable than other approaches. In this sense, a lot of effort has been put in developing new function decomposition methods that are able to identify groups of interacting variables that should be optimized altogether. Good decomposition methods help to simplify the problem as subcomponents can be optimized independently.
- Differential Evolution seems to play an important role in many of the algorithms with the best performance. However, in most cases, it must be combined with other algorithms (normally, a local search) to perform as well.
- The inclusion of strong local searches also seems to be the trend in many of the best-performing methods. If the combination of a population-based method and a local search is properly managed, a good balance between exploration and exploitation can be obtained.
- Another interesting emergent strategy is to include mechanisms to manage how Fitness Evaluations (FEs) are allocated, trying to expend more FEs on dimensions (or components) with more influence on the overall fitness value.
- Some researchers are focusing their efforts on particular characteristics of the functions. For example, some recent studies have only considered non-separable functions. Although benchmarks always try to include functions with different characteristics, real-world problems do not always include all of them. For this reason, in these scenarios, using specialized methods for some particular characteristics can be very useful.
- A new type of very large-scale problems seems to be gaining popularity. The complexity of scientific and industrial problems is continuously increasing and we will need to able to deal with problems of several thousands or even millions of variables. In this sense, there are incipient studies that explore the possibility of using GPUs to make state-of-the-art algorithms in continuous LSGO successful also when the problem size is pushed to the limit.
- Finally, although it is not a single algorithm rather a family of them constructed with the same framework, Multiple

[31] http://www.husseinabbass.net/BigOpt.html

Offspring Sampling (MOS) seems to be a clear reference in the field of continuous LSGO, as the results in the special sessions of 2012, 2013, and 2015 and the special issue 2011 indicate [177].

These findings are summarized in Table 1.

Current Trends after a Decade of Competitions

After a decade of competitions in real-optimization and many proposed algorithms, this historical information can be used to try to answer several important questions:

- Are there any algorithms that hold a strong influence on the following algorithms in the field?
- What algorithms' components are more successful?
- Are there any techniques that use some successful algorithms that other algorithms also use to improve themselves?

In this section, we are going to try to answer these questions.

Influential Algorithms

In any research field, a good indication of its evolution is to what extent some algorithms have influenced others over the years, reusing ideas from previous successful proposals. For this reason, the first big question that we would like to answer is: does this happen within the real-parameter competitions? We discuss the influence of three important algorithms: CMA-ES, L-SHADE, and MVMO.

Table 1 Summary of the identified trends in continuous LSGO

Main trends	
Hybridization	Good results, scalability?
Cooperative Co-Evolution	Average results, better scalability
Relevant algorithms	
Differential Evolution	Base of many hybrid methods
Local Searches	Strong LS for better exploitation
Other characteristics	
FEs allocation methods	More effort in more rewarding variables
GPus	Necessary for very large-scale problems
Specialized optimizers	Focus on particular characteristics of problems
Prominent results	
Multiple Offspring Sampling (MOS)	Not a single algorithm but a framework

The first algorithm with a strong influence in the field was CMA-ES [24]. The algorithms based on this approach, such as IPOP-CMA-ES [25], BIPOP-CMA-ES [79], or Ni-BIPOP-aCMA [36], were the winners of the CEC'2005 and the BBOB'2009 competitions, and the runner-up in the CEC'2013 competition, respectively. Furthermore, there are many algorithms inspired by CMA-ES that have won several editions of the BBOB competition. It has been used as local search method, as in DRMA-LSCh-CMA [39] or as a component in a hybridization such as iCMAES-ILS [35], the winner of the CEC'2013 competition. However, its influence is not only constrained to global optimization. There are some algorithms using it in large-scale global optimization, such as CC-CMA-ES [162]. In Fig. 1, the influence of the CMA-ES algorithm can be observed.

A second algorithm with great influence is L-SHADE [45], which evolves from the previous algorithm SHADE [42] (an improvement of the well-known JADE [56] method) with a small change, the ability to reduce the population size during the search. With this small change, L-SHADE won in 2014, when SHADE only ranked fourth in 2013. Its good results, and maybe the availability of its source code,[32] have favored an interesting list of winners using it in CEC'2015 (L-SHADE-ND [58], SPS-L-SHADE-EIG [53]) and CEC'2016 (LSHADE-EpSin [65], iLSHADE [66], LSHADE44 [68], and others with worse results). Analogously, Fig. 2 depicts the influence of L-SHADE in the last few years, although all these algorithms share a common origin, which is the JADE algorithm.

A third algorithm with good results was the MVMO scheme algorithm, which proves that being a scheme is not only good for global optimization [47, 67] but also for when the number of evaluations is very scarce (as shown by its good behavior in expensive benchmark [47]). In this case, the different proposals came from the same authors, maybe because that scheme is not as popular as the DE scheme. The relationship among those versions is graphically shown in Fig. 3.

Obviously, the algorithms with the best results in the competitions are the most influential. However, sometimes, such as with the SHADE or VMO algorithms, their first versions were not among the winners, but they were evolved and, over the next few years, other methods based on them performed better achieving winning positions.

Bio-inspired Algorithms: Increasing Scenario

In recent years, a great variety of bio-inspired algorithms have been published in the literature [2, 14]. These algorithms simulate some biological processes such as natural evolution, where solutions are individuals that mutate and reproduce to generate new candidate ones. When they

[32] at https://sites.google.com/site/tanaberyoji/home

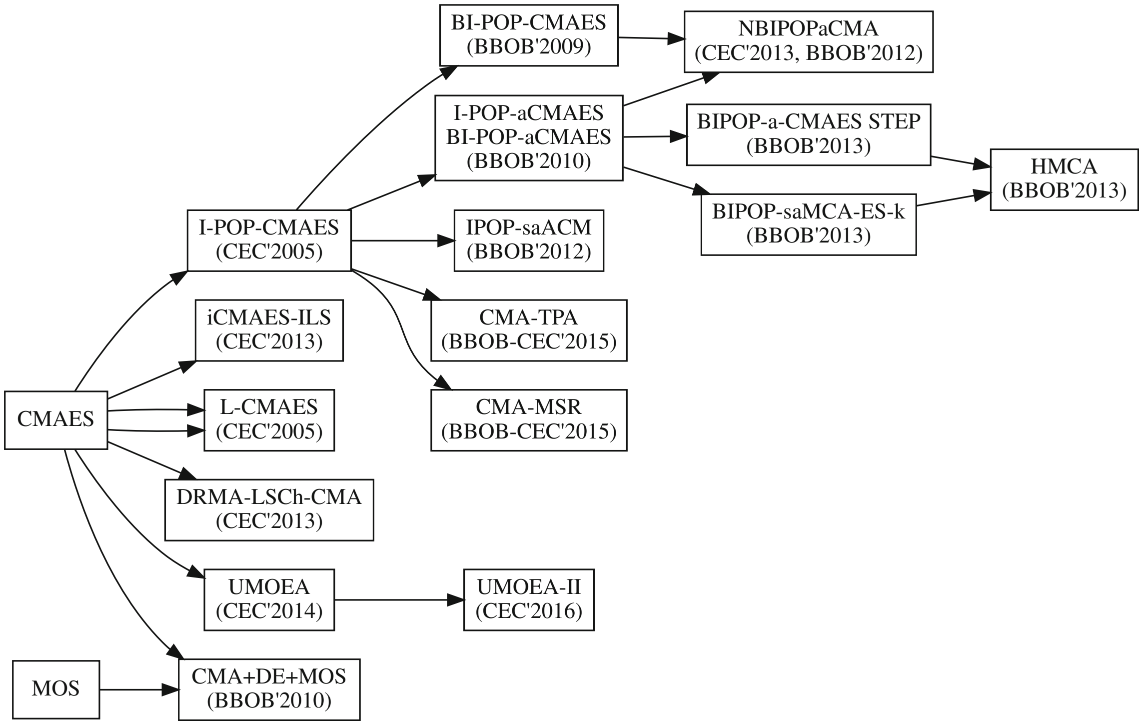

Fig. 1 Influence of CMA-ES algorithm in competitions

mimic a collective behavior they are called Swarm Intelligence [178]. These are inspired by different biological behaviors: movements of birds [179], bats [180], or small insects such as fireflies [181, 182], grasshoppers [183], or even mussels [184]; mechanisms to locate food exhibited by colony animals such as ants in Artificial Ant Colony (ACO) [185, 186], or bees in Artificial Bee Colony (ABC) algorithms [187]; hunting mechanisms used by different animals, from small ones such as dragonflies [188], to wild wolfs [189] or marine animals such as dolphins [190] or whales [191]; even the reproduction of corals [192], the behavior of very small animals such as krill [193] or the immune system in Artificial Immune System (AIS) optimization [194], to name a few.

These algorithms have recently proven to be especially good for a large number of cognitive problems. For instance, grasshopper optimization for identifying relevant features for the classification of diseases [19], ABC for solving cognitive wireless sensor networks [195], evolutionary algorithms for cognitive multitasking [196], bio-inspired algorithms for denoising biomedical images [197], PSO for estimating unknown parameters [198], for planning [199] and for creating cognitive taxonomies [20], nature-inspired chemical reaction optimization [200], and the dolphin algorithm for learning a neural network (used for image recognition) [201] or cuckoo search for cognitive image registration [18]. Furthermore, the use of swarm intelligence has been evaluated as a synthetic collective intelligence capable of exploring decision making [202].

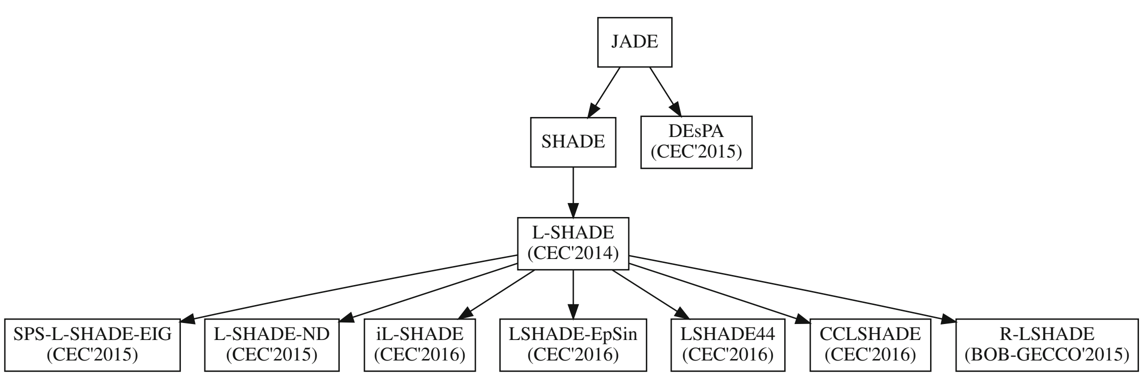

Fig. 2 Influence of L-SHADE algorithm in competitions

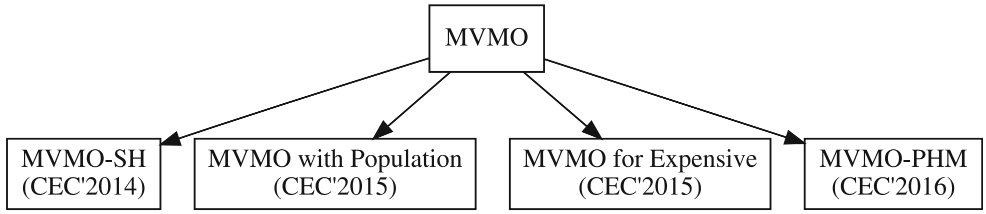

Fig. 3 Influence of MVMO algorithm in competitions

Finally, we would also like to note that the metaheuristics with the best results in these competitions (CMA-ES, DE, and MVMO) are far from being bio-inspired algorithms, although some of them retain their nature-inspired roots. In this sense, these results remind us that the *novelty* of new metaheuristics is very important but also subordinated to their performance in solving optimization problems.

There is a continuous debate in the nature-inspired community between novelty and the need to get competitive results (see the discussion in [203–205]).

Highlighted Techniques/Components

One of the most interesting issues when studying successful algorithms is the identification of the different components that each algorithm uses, as they can be further used by other methods to boost their performance.

In the following paragraphs, we are going to highlight several popular techniques/components that are currently being used by many different algorithms.

- A frequent problem with many algorithms is the selection of appropriate values for their parameters, as the performance of one algorithm can be completely different when these values differ. As a result, they should be carefully chosen or an automatic parameter tuning tool [205] should be used. Furthermore, the different nature of the functions in these benchmarks requires a robust algorithm, a behavior difficult to obtain with fixed values. Thus, a lot of algorithms use self-adaptive criteria to adapt their parameters (a good example are parameters F and CR of DE: most of the popular methods use techniques to adapt these two parameters). The general idea is to generate diverse values for these parameters by learning a function distribution and maintaining a memory. This memory is used to adapt the mean of the distribution taking into account which values have produced the best solutions (and sometimes also the worst ones) to enforce even better ones.

  Curiously, due to this trend in designing robust algorithms, most of the algorithms implement this self-adaptive behavior, even when a specific benchmark, such as the one at CEC'2015 for learning-based optimization, allows researchers to use different values for each function.
- Other algorithms not only self-adapt their parameters but also their components, having several components that provide the same functionality (like the crossover operator, constraint technique, etc.) and then selecting one of them according to their performance when they were previously used. This approach is usually applied in two different ways. The first option is to select a component with a certain probability and, after a number of fitness

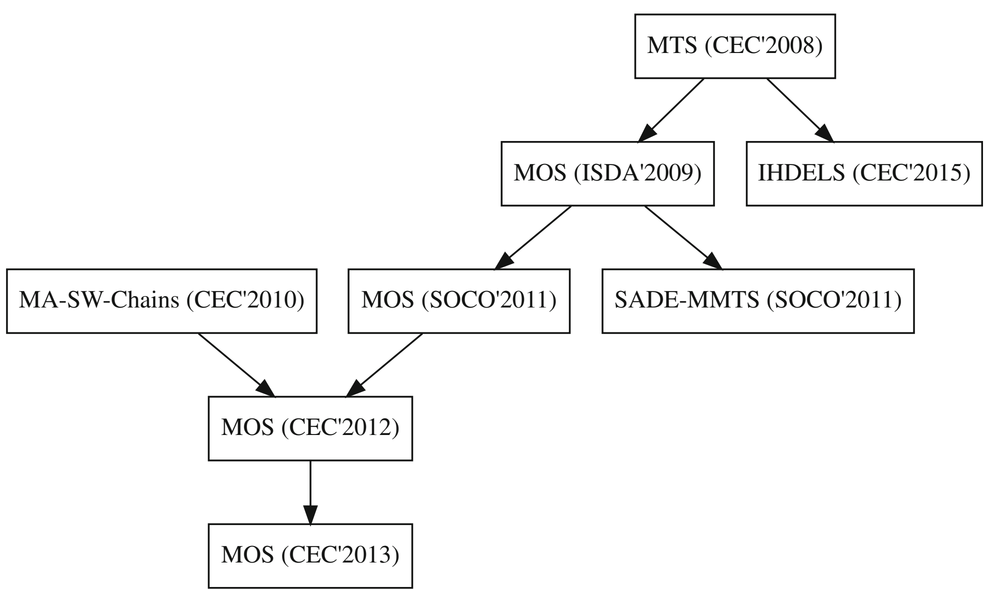

Fig. 4 Evolution of Memetic Large-Scale Global Optimizers

evaluations, to increase the probability of the component that generates the most successful solutions (considering which one was inserted into the population or by its fitness). The second option, widely followed by several winner algorithms, is to apply (to the same solutions or to a different sub-population) different algorithms during a certain number of evaluations. Then, the algorithm that obtained the best results is applied to all the new solutions until the maximum number of evaluations is reached.

- When the self-adaptive component is not a part of an optimization algorithm but of a complete algorithm, the proposal can be considered as a framework of algorithms. One proposal can be designed to have a particular combination, such as in the case of iCMAES-ILS [35], or allow a more open selection of optimization methods, as in MOS [82].
- Most of the proposals are memetic algorithms, because they incorporate an improvement method to obtain accurate solutions, while global exploration component is incorporated at the same time. The local search method used depends on many different types of methods, from more general ones as quasi-newton to more advanced ones such as CMA-ES, or more specific approaches, such as the ones used in MTS and other LSGO algorithms. Figure 4 graphically depicts this scenario, showing how a majority of the most successful algorithms in LSGO incorporate a strong local search as the intensification component of the method.
- In order to increase the selective pressure in the populations, one approach popularized by L-SHADE and adopted by other algorithms is to decrease the population size during the run. This traditionally reduced the diversity too fast and was therefore not acceptable. However, nowadays many algorithms use a memory of solutions to maintain diversity in the search, and so the reduction of the population size does not imply bad diversity and can thus improve the search.
- Traditionally, in most of the best-performing methods, only the best solutions were considered to guide the search. This means that a lot of information was being wasted in each generation. In more recent algorithms, such as CMA-aCMA [37], bad solutions are also used to guide the search. Similarly, in the MVMO family of algorithms [47, 49, 60], not only is the best solution considered but also the average of a group of good solutions.

Conclusions

The use of Bio-inspired and Evolutionary Algorithms for real-parameter optimization is of great interest today, and thus many approaches based on this type of optimization are proposed each year. This large number of proposals makes it difficult for researchers to follow the evolution of the field. In this paper, we have reviewed the different competitions for each type of real-parameter optimization problems, highlighting the winners. We have observed that there are several algorithms, like CMA-ES, L-SHADE, MVMO, and MOS, which have obtained a strong influence over other algorithms. We have also suggested several techniques that are being widely adopted among the winning proposals and which could be used for more competitive algorithms.

The objective of this review and analysis of the evolution of the competitions is to offer a useful reference to new researchers in this research topic and help them to continue improving the field.

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

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

Compliance with Ethical Standards

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Appendix: List of algorithms in competitions

In order to summarize the most relevant algorithms in the different competitions covered by the paper, Tables 2, 3, 4, 5, and 6 show this information, grouped by benchmark and conference and ordered according to their ranking in the corresponding competitions. In particular, Tables 2 and 3 show the more relevant algorithms in global optimization for the different CEC and BBOB benchmarks, respectively. On the other hand, Table 4 shows the algorithms in the Black-Box Competition for the different conferences, whereas Table 5 shows the algorithms for other competitions such as constraint and multimodal optimization. Finally, Table 6 summarizes the information of the algorithms for large-scale global optimization.

Table 2 List of algorithms in CEC global competitions (sorted by ranking)

Benchmark	Competition'Year	Algorithm	Reference
CEC'2005 Global Optimization	CEC'2005	IPOP-CMAES	[25]
		L-SaDE	[28]
		DMS-L-PSO	[29]
	CEC'2011	GA-MPC	[31]

Table 2 (continued)

Benchmark	Competition'Year	Algorithm	Reference
CEC'2011 Global Optimization		DE-ΛCR SAMODE	[32] [33]
CEC'2013 Global Optimization	CEC'2013	IPOP-CMA-ES NBIPOPaCMA DRMA-LSCh-CMA	[35] [36] [39]
CEC'2014 Global Optimization	CEC'2014	L-SHADE UMOEA MVMO-SH	[45] [46] [47]
CEC'2015 Learning-Based Optimization	CEC'2015	SPS-L-SHADE-EIG DEsPA MVMO LSHADE-ND	[53] [55] [57] [58]
CEC'2015 Expensive Optimization	CEC'2015	MVMOexp TunnedCMAES DRPSO	[60] [62] [63]
CEC'2014 Global Optimization	CEC'2016	UMOEA-II LSHADE-EpSin DRPSO	[64] [65] [63]
CEC'2015 Learning-Based Optimization	CEC'2016	MVMO-PHM LSHADE44 CCLSHADE	[67] [68] [69]
CEC'2015 Expensive Optimization	CEC'2016	MVMO-PHM AsBeC_tunned RYYO	[67] [70] [71]
CEC'2017 Global Optimization	CEC'2017	jSO LSHADE-cnEpSin LSHADE_SPACMA	[73] [74] [75]
CEC'2015 Expensive Optimization	CEC'2017	HOCO	[76]

Table 4 List of the most relevant algorithms in the Black-Box Benchmark competitions (BBComp)

Competition	Algorithm	Reference
GECCO'2015 Track	KNITRO MVMO'2015 NSMO	[98] [57] [99]
CEC'2015 Track	UMOEA Two-stage algorithm	[46] [100]
GECCO'2016 Track	Two-stage algorithm	[100]
Expensive GECCO'2016 Track	KNITRO Two-stage algorithm	[98] [100]
GECCO'2017 Track	DTS-CMA-ES Two-stage algorithm	Unpublished [100]
Expensive GECCO'2017 Track	DTS-CMA-ES-BOBYQAP Restarted model-based optimization with L-BFGS-B	Unpublished Unpublished
EMO'2017 Track	Model-based HV-maximization Bayesian Multi-Objective Optimization PADDS-CHC	Unpublished [103] Unpublished

Table 3 List of the most relevant algorithms in the Black-Box Optimization Benchmark competitions

Congress	Competition'Year	Algorithm	Reference
GECCO'2009	BBOB'2009	BI-POP-CMAES AMaLGaM IDEAL	[79] [80]
GECCO'2010	BBOB'2010	IPOP-aCMAES	[37]
GECCO'2012	BBOB'2012	IPOP-saACM NBIPOP-aCMAES	[83] [85]
GECCO'2013	BBOB'2013	HCMA HMLSL	[86] [87]
GECCO'2015	BBOB'2015	R-LSHADE	[90]
CEC'2015	BBOB'2015	CMA-TPA CMA-MSR	[89] [89]
GECCO'2017	BBOB'2017	Modified CMA-ES Algorithms Distributed Pool	[93, 94 95] [97]

Table 5 List of the most relevant algorithms in other competitions

Benchmark competition	Algorithm	Reference
CEC'2006 Constraint Optimization	εDE DMS-PSO jDE-2	[105] [29] [106]
CEC'2010 Constraint Optimization	εDEg ECHT	[109] [110]
CEC'2013 Multimodal Optimization	NEA2 dADE/nrand/1 CMAES with archive	[113] [115] Unpublished
CEC'2015 Multimodal Optimization	NMMSO NEA2 LSEDA	[120] [113] [117]
CEC'2016 Multimodal Optimization	RS-CMSA NMMSO RLSIS	[118] [120] Unpublished

Table 6 List of the most relevant algorithms in Large-Scale Global Optimization competitions

Benchmark	Competition	Algorithm	Reference
CEC'2008 LSGO	CEC'2008 LSGO	MTS	[108]
		LSEDA-gl	[125]
		jDEdynNP	[126]
CEC'2010 Benchmark	CEC'2010 LSGO	MA-SW-Chains	[133]
		EOEA	[134]
		DASA	[135]
Soft Computing Benchmark	Soft Computing Special Issue	MOS	[82]
		GaDE	[144]
		jDElscop	[140]
CEC'2010 Benchmark	CEC'2012 LSGO	Improved MOS	[153]
		jDEsps	[154]
		Two-phase algorithm with CCGS	[155]
CEC'2013 Benchmark	CEC'2013 LSGO	MOS with MTS-LS1-Reduced	[159]
		Smoothing and auxiliary CC	[160]
CEC'2010 Benchmark	CEC'2014 LSGO	Center-Based Initialization CC	[164]
		Hybrid approach	[165]
CEC'2013 Benchmark	CEC'2014 LSGO	Variable grouping DE	[166]
CEC'2013 Benchmark	CEC'2015 LSGO	IHDELS	[167]
		DEEPSO	[168]
CEC'2015 BigOpt	CEC'2015 Big Optimization	MAGA	[175]
CEC'2013 Benchmark	CEC'2016 LSGO	CC-DIG	[169]
		CRO	[170]
		CBCC	[172]

References

1. Glover F, Kochenberger GA. Handbook of metaheuristics. Springer; 2003.
2. Xiong N, Molina D, Ortiz ML, Herrera F. A walk into metaheuristics for engineering optimization: principles, methods and recent trends. Int J Comput Intell Syst, 2015. 2015 June;8(4): 606–36.
3. Bäck T, Fogel DB, Michalewicz Z, editors. Handbook of evolutionary computation. Bristol: IOP Publishing Ltd.; 1997.
4. Beyer HG, Deb K. On self-adaptive features in real-parameter evolutionary algorithms. IEEE Trans Evol Comput. 2001;5(3): 250–70.
5. Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. IEEE Trans Evol Comput. 2016 Aug;20(4):606–26.
6. Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley; 1989.
7. Herrera F, Lozano M, Verdegay JL. Tackling real-coded genetic algorithms: operators and tools for the behavioral analysis. Artificial Intelligence Reviews. 1998;12(4):265–319.
8. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983;220:671–80.
9. Beyer HG, Evolution Strategies SHP. Nat Comput. 2002;1:3–52.
10. Lozano JA, Larrañaga P, Inza I, Bengotxea E. Towards a new evolutionary computation: advances on estimation of distribution algorithms: Springer Verlag; 2006.
11. Storn R, Price K. Differential Evolution - A Simple and Efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997;11:341–359.
12. Price KV, Rainer RM, Lampinen JA. Differential evolution: a practical approach to global optimization: Springer-Verlag; 2005.
13. Siddique N, Adeli H. Nature inspired computing: an overview and some future directions. Cogn Comput. 2015;7(6):706–14.
14. Kumar A. Bio inspired computing—a review of algorithms and scope of applications. Expert Syst Appl. 2016;59:20–32.
15. Paredis J. Coevolutionary computation. Artificial Life. 1995;2(4): 355–75.
16. Smith JE. Coevolving memetic algorithms: a review and progress report. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2007;37(1):6–17.
17. Moscato P, Cotta C. A gentle introduction to memetic algorithms. In: Handbook of metaheuristics. Kluwer Academic Publishers, Boston, MA; 2003. p. 105–144.
18. Chakraborty S, Dey N, Samanta S, Ashour AS, Barna C, Balas MM. Optimization of non-rigid demons registration using cuckoo search algorithm. Cogn Comput. 2017;9(6):817–26.
19. Aljarah I, Al-Zoubi AM, Faris H, Hassonah MA, Mirjalili S, Saadeh H. Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cognitive Computation. 2018;p. In press.
20. Yahya AA, Osman A, El-Bashir MS. Rocchio algorithm-based particle initialization mechanism for effective PSO classification of high dimensional data. Swarm and Evolutionary Computation. 2017;34:18–32.

21. Bersini H, Dorigo M, Langerman S, Seront G, Gambardella L. Results of the first international contest on evolutionary optimisation (1st ICEO). In: IEEE congress on evolutionary computation (CEC'96); 1996. p. 611–4.
22. Garcia S, Molina D, Lozano M, Herrera F. 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. 2009;15(6):617–44.
23. Liao T, Molina D, de Oca M, Sützle T. A note on bound constraints handling for the IEEE CEC'05 benchmark function suite. Evol Comput. 2014;22(2):351–9.
24. Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 2003;1(11):1–18, 1.
25. Auger A, Hansen NA. Restart CMA evolution strategy with increasing population size. 2005 I.E. Congress on Evolutionary Computation. 2005:1769–76.
26. Auger A, Hansen N. Performance evaluation of an advanced local search evolutionary algorithm. 2005 I.E. Congress on Evolutionary Computation. vol. 2. 2005:1777–84.
27. Yuan B, Gallagher M. Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA. 2005 I.E. Congress on Evolutionary Computation vol 2. 2005:1792–9. **2.**
28. Qin AK, Suganthan PN. Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 I.E. congress on evolutionary computation. Vol. 2; 2005. p. 1785–91. **2.**
29. Liang JJ, Suganthan PN. Dynamic multi-swarm particle swarm optimizer with local search. In: 2005 I.E. congress on evolutionary computation. Vol. 1; 2005. p. 522–8. **1.**
30. S Das PNS. Problem definitions and evaluation criteria for CEC 2011 Competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, India and Nanyang; 2010.
31. Elsayed SM, Sarker RA, Essam DL. GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems. IEEE Congress of Evolutionary Computation (CEC); 2011. p. 2011:1034–40.
32. Reynoso-Meza G, Sanchis J, Blasco X, Herrero JM. Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems. IEEE Congress of Evolutionary Computation (CEC); 2011. p. 2011:1551–6.
33. Elsayed SM, Sarker RA, Essam DL. Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. IEEE Congress of Evolutionary Computation (CEC); 2011 p. 2011:1041–8.
34. Liang JJ, Qu BY, Suganthan PN, Hernández-Díaz AG. Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore; 2013.
35. Liao T, Stützle T. Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. In: 2013 I.E. Congress on Evolutionary Computation. 2013:1938–44.
36. Loshchilov I. CMA-ES with restarts for solving CEC 2013 benchmark problems. In: 2013 I.E. congress on evolutionary computation; 2013. p. 369–376.
37. Hansen N, Ros R. Benchmarking a weighted negative covariance matrix update on the BBOB-2010 noiseless testbed. In: Proceedings of the 12th annual conference companion on genetic and evolutionary computation. GECCO '10. New York: ACM; 2010. p. 1673–80.
38. Loshchilov I, Schoenauer M, Sebag M. Alternative restart strategies for CMA-ES. In: Parallel problem solving from nature - PPSN XII: 12th international conference, Taormina, Italy, September 1–5, 2012, proceedings, part I. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. p. 296–305.
39. Lacroix B, Molina D, Herrera F. Dynamically updated region based memetic algorithm for the 2013 CEC special session and competition on real parameter single objective optimization. In: 2013 I.E. congress on evolutionary computation; 2013. p. 1945–1951.
40. Molina D, Lozano M, García-Martínez C, Herrera F. Memetic algorithms for continuous optimisation based on local search chains. Evol Comput. 2010;18(1):27–63.
41. Rueda JL, Erlich I. Hybrid mean-variance mapping optimization for solving the IEEE-CEC 2013 competition problems. In: 2013 I.E. congress on evolutionary computation; 2013. p. 1664–71.
42. Tanabe R, Fukunaga A. Evaluating the performance of SHADE on CEC 2013 benchmark problems. In: 2013 I.E. Congress on Evolutionary Computation. 2013:1952–9.
43. Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore; 2013.
44. Liu B, Chen Q, Zhang Q, Liang JJ, Suganthan PN, Qu BY. Problem definitions and evaluation criteria for computationally expensive single objective numerical optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore; 2013.
45. Tanabe R, Fukunaga AS. Improving the search performance of SHADE using linear population size reduction. In: 2014 I.E. Congress on Evolutionary Computation (CEC); 2014. p. 1658–1665.
46. Elsayed SM, Sarker RA, Essam DL, Hamza NM. Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization. IEEE Congress on Evolutionary Computation (CEC); 2014 p. 2014:1650–7.
47. Erlich I, Rueda JL, Wildenhues S, Shewarega F. Evaluating the mean-variance mapping optimization on the IEEE-CEC 2014 test suite. In: 2014 I.E. Congress on Evolutionary Computation (CEC); 2014. p. 1625–1632.
48. Molina D, Lacroix B, Herrera F. Influence of regions on the memetic algorithm for the CEC'2014 special session on real-parameter single objective optimisation. 2014 I.E. Congress on Evolutionary Computation (CEC); 2014 p:1633–40.
49. Erlich I, Rueda JL, Wildenhues S, Shewarega F. Solving the IEEE-CEC 2014 expensive optimization test problems by using single-particle MVMO. In: 2014 I.E. Congress on Evolutionary Computation (CEC); 2014. p. 1084–1091.
50. Liang JJ, Chen Q, Qu BY, Liu B, Suganthan PN, Chen Q. Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University; 2014.
51. Chen Q, Liu B, Zhang Q, Liang JJ, Suganthan PN, Qu BY. Problem definition and evaluation criteria for CEC 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization. Computational Intelligence Laboratory: Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University; 2014.
52. Qu BY, Liang JJ, Wang ZY, Chen Q, Suganthan PN. Novel benchmark functions for continuous multimodal optimization with

comparative results. Swarm and Evolutionary Computation. 2016;26:23–34.
53. Guo SM, Tsai JSH, Yang CC, Hsu PH. A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 1003–1010.
54. Guo SM, Yang CC, Hsu PH, Tsai JSH. Improving differential evolution with successful-parent-selecting framework. IEEE Transaction on Evolutionary Computation. 2015;19(5):717–30.
55. Awad N, Ali MZ, Reynolds RG. A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 1098–1105.
56. Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. Evolutionary Computation, IEEE Transactions on. 2009;13(5):945–58.
57. Rueda JL, Erlich I. Testing MVMO on learning-based real-parameter single objective benchmark optimization problems. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 1025–1032.
58. Sallam KM, Sarker RA, Essam DL, Elsayed SM. Neurodynamic differential evolution algorithm and solving CEC2015 competition problems. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 1033–1040.
59. Xia Y, Wang J. A general projection neural network for solving monotone variational inequalities and related optimization problems. IEEE Trans Neural Netw. 2004;15(2):318–28.
60. Rueda JL, Erlich I. MVMO for bound constrained single-objective computationally expensive numerical optimization. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 1011–1017.
61. Andersson M, Bandaru S, Ng AHC, Syberfeldt A. Parameter tuned CMA-ES on the CEC'15 expensive problems. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 1950–1957.
62. Berthier V. Experiments on the CEC 2015 expensive optimization testbed. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015 p 1059–1066.
63. Tanweer MR, Suresh S, Sundararajan N. Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems. IEEE Congress on Evolutionary Computation (CEC); 2015. p. 2015:1943–9.
64. Elsayed S, Hamza N, Sarker R. Testing united multi-operator evolutionary algorithms-II on single objective optimization problems. In: 2016 I.E. Congress on Evolutionary Computation (CEC); 2016. p. 2966–2973.
65. Awad NH, Ali MZ, Suganthan PN, Reynolds RG. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In: 2016 I.E. Congress on Evolutionary Computation (CEC); 2016. p. 2958–2965.
66. Brest J, M Sepesy Maˇcec BB.iL-SHADE: improved L-SHADE algorithm for single objective real-parameter optimization testing united multi-operator evolutionary algorithms-II on single objective. In: 2016 I.E. congress on evolutionary computation (CEC); 2016. p. 1188–1195.
67. Rueda JL, Erlich I. Solving the CEC2016 real-parameter single objective optimization problems through MVMO-PHM: Delft University of Technology and University Duisburg-Essen; 2016.
68. Poláková R, Tvrdík J, Bujok P. L-SHADE with competing strategies applied to CEC2015 learning-based test suite. In: 2016 I.E. Congress on Evolutionary Computation (CEC); 2016. p. 4790–4796.
69. Omidvar MN, Li X, Mei Y, Yao X. Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans Evol Comput. 2014;18(3):378–93.
70. E Ampellio LV. A hybrid ABC for expensive optimizations: CEC 2016 competition benchmark. In: 2016 I.E. Congress on Evolutionary Computation (CEC); 2016. p. 1157–1164.
71. Punnathanam V, Kotecha P. Reduced yin-Yang-pair optimization and its performance on the CEC 2016 expensive case. In: 2016 I.E. Congress on Evolutionary Computation (CEC); 2016 p 2996–3002.
72. Punnathanam V, Kotecha P. Yin-Yang-pair optimization: a novel lightweight optimization algorithm. Eng Appl Artif Intell. 2016;54:62–79.
73. Brest J, Maucec MS, Boskovic B. Single objective real-parameter optimization: algorithm jSO. In: 2017 I.E. Congress on Evolutionary Computation (CEC); 2017. p. 1311–1318.
74. Awad N, Ali M, Suganthan PN. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 I.E. Congress on Evolutionary Computation (CEC); 2017. p. 372–379.
75. Mohamed A, Hadi A, Fattouh A, Jambi K. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In: 2017 I.E. Congress on Evolutionary Computation (CEC); 2017 p 145–152.
76. Tanweer MR, Sundaram S, Sundararajan N. Development of a higher order cognitive optimization algorithm. In: 2017 I.E. Congress on Evolutionary Computation (CEC); 2017. p. 2752–2758.
77. Hansen N, Auger A, Mersmann O, Tusar T, Brockhoff D. COCO: a platform for comparing continuous optimizers in a black-box setting. ArXiv e-prints. 2016.
78. Hansen N, Auger A, Ros R, Finck S, Pošík P. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Proceedings of the 12th annual conference companion on genetic and evolutionary computation. GECCO '10. New York: ACM; 2010. p. 1689–96.
79. Hansen N. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In: Proceedings of the 11th annual conference companion on genetic and evolutionary computation conference: late breaking papers. GECCO '09. New York: ACM; 2009. p. 2389–96.
80. Bosman PAN, Grahl J, Thierens D, AMaLGaM IDEA. In noiseless black-box optimization benchmarking. In: Proceedings of the 11th annual conference companion on genetic and evolutionary computation conference: late breaking papers. GECCO '09. New York: ACM; 2009. p. 2247–54.
81. LaTorre A, Muelas S, Peña JM. Benchmarking a MOS-based algorithm on the BBOB-2010 noiseless function testbed. In: Proceedings of the 12th annual conference companion on genetic and evolutionary computation. GECCO '10. New York: ACM; 2010. p. 1649–56.
82. LaTorre A, Muelas S, Peña JM. A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test. Soft Computing - A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2187–99.
83. Loshchilov I, Schoenauer M, Sebag M. Black-box Optimization Benchmarking of IPOP-saACM-ES on the BBOB-2012 Noisy Testbed. In: Proceedings of the 14th annual conference companion on genetic and evolutionary computation. In GECCO '12 Proceeding; 2012. p. 261–268.
84. Loshchilov I, Saclay TI, Schoenauer M, Sebag M. Self-Adaptive Surrogate-Assisted Covariance Matrix Adaptation Evolution Strategy. In: Proceedings of the 14th annual conference companion on genetic and evolutionary computation. In GECCO'2012 Proceedings; 2012. p. 321–328.
85. Loshchilov I, Schoenauer M, Sebag M. Black-box optimization benchmarking of NIPOP-aCMA-ES and NBIPOP-aCMA-ES on the BBOB-2012 noiseless testbed. In: Proceedings of the 14th

annual conference companion on genetic and evolutionary computation. GECCO ‘12. New York: ACM; 2012. p. 269–76.
86. Loshchilov I, Schoenauer M, Sebag M. BI-population CMA-ES algorithms with surrogate models and line searches. In: BBOB workshop of genetic and evolutionary computation conference (GECCO 2013); 2013. p. 1177–84.
87. Pál L. Benchmarking a hybrid multi level single linkage algorithm on the Bbob noiseless testbed. In: Proceedings of the 15th annual conference companion on genetic and evolutionary computation. GECCO ‘13 companion. New York: ACM; 2013. p. 1145–52.
88. Ulmer H, Streichert F, Zell A. Evolution strategies assisted by Gaussian processes with improved pre-selection criterion. In: in IEEE Congress on Evolutionary Computation,CEC 2003; 2003. p. 692–699.
89. Atamna A. Benchmarking IPOP-CMA-ES-TPA and IPOP-CMA-ES-MSR on the BBOB noiseless testbed. In: Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation. GECCO ‘15 companion. ACM; 2015. p. 1135–42.
90. Tanabe R, Fukunaga A. Tuning differential evolution for cheap, medium, and expensive computational budgets. In: 2015 I.E. Congress on Evolutionary Computation (CEC); 2015. p. 2018–2025.
91. Nishida K, Akimoto Y. Evaluating the population size adaptation mechanism for CMA-ES on the BBOB noiseless testbed. In: Proceedings of the 2016 on genetic and evolutionary computation conference companion. GECCO ‘16 companion; 2016. p. 1185–92.
92. Nishida K, Akimoto Y. Evaluating the population size adaptation mechanism for CMA-ES on the BBOB noisy testbed. In: Proceedings of the 2016 on genetic and evolutionary computation conference companion. GECCO ‘16 companion; 2016. p. 1193–200.
93. Nguyen DM, Hansen N. Benchmarking CMAES-APOP on the BBOB noiseless testbed. In: Proceedings of the genetic and evolutionary computation conference companion. GECCO ‘17; 2017. p. 1756–63.
94. Pitra Z, Bajer L, Repický J, Holeňa M. Comparison of ordinal and metric Gaussian process regression as surrogate models for CMA evolution strategy. In: Proceedings of the genetic and evolutionary computation conference companion. GECCO ‘17. New York, NY, USA; 2017. p. 1764–71.
95. Yamaguchi T, Akimoto Y. Benchmarking the novel CMA-ES restart strategy using the search history on the BBOB noiseless testbed. In: Proceedings of the genetic and evolutionary computation conference companion. GECCO ‘17. New York, NY, USA; 2017. p. 1780–7.
96. Aydin D, Yavuz G. Self-adaptive search equation-based artificial bee colony algorithm with CMA-ES on the noiseless BBOB testbed. In: Proceedings of the genetic and evolutionary computation conference companion. GECCO ‘17. New York, NY, USA; 2017. p. 1742–9.
97. García-Valdez M, Merelo JJ. Benchmarking a pool-based execution with GA and PSO workers on the BBOB noiseless testbed. In: Proceedings of the genetic and evolutionary computation conference companion. GECCO ‘17. New York, NY, USA; 2017. p. 1750–5.
98. Byrd RH, Nocedal J, Waltz RA. Knitro: an integrated package for nonlinear optimization. Boston, MA: Springer US; 2006.
99. Al-Dujaili A, Suresh S. A naive multi-scale search algorithm for global optimization problems. Inf Sci. 2016;372:294–312.
100. Wessing S. Two-stage methods for multimodal optimization [PhD dissertation]. In: techniche universität dortmund; 2015.
101. Jiménez AJ. Sparse Hessian factorization in curved trajectories for unconstrained minimization. Optimization Methods and Software. 2014;29(1):1–9.
102. Beume N, Naujoks B, Emmerich M. SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur J Oper Res. 2007;181(3):1653–69.
103. Feliot P, Bect J, Vazquez E. A Bayesian approach to constrained single- and multi-objective optimization. J Glob Optim. 2017 Jan;67(1):97–133.
104. Mezura-Montes E, Coello Coello CA. Constraint-handling in nature-inspired numerical optimization: Past, present and future. Swarm and Evolutionary Computation. 2011;1(4):173–194. Cited By 216.
105. Takahama T, Sakai S. Constrained optimization by the constrained differential evolution with gradient-based mutation and feasible elites. In: 2006 I.E. international conference on. Evol Comput. 2006:1–8.
106. Huang VL, Qin AK, Suganthan PN. Self-adaptive differential evolution algorithm for constrained real-parameter optimization. In: 2006 I.E. international conference on evolutionary computation; 2006. p. 17–24.
107. Mallipeddi R, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. Singapore: Nanyang Technological University; 2010.
108. Tseng LY, Chen C. Multiple trajectory search for large scale global optimization. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 3052–9.
109. Takahama T, Sakai S. Constrained optimization by the constrained differential evolution with an archive and gradient-based mutation. IEEE Congress on Evolutionary Computation. 2010:1–9.
110. Mallipeddi R, Suganthan PN. Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems. IEEE Congress on Evolutionary Computation. 2010:1–8.
111. Dasa S, Maity S, Qu BY, Suganthan PN. Real-parameter evolutionary multimodal optimization-A survey of the state-of-the-art. Swarm and Evolutionary Computation. 2011;1(2):71–78. Cited By 110.
112. Li X, Engelbrecht A, Epitropakis MG. Benchmark functions for CEC’2013 special session and competition on niching methods for multimodal function optimization’. Melbourne: Evolutionary Computation and Machine Learning Group, RMIT University; 2013.
113. Preuss M. Niching the CMA-ES via nearest-better clustering. In: Proceedings of the 12th annual conference companion on genetic and evolutionary computation. GECCO ‘10. New York: ACM; 2010. p. 1711–8.
114. Preuss M, Schönemann L, Emmerich M. Counteracting genetic drift and disruptive recombination in (μ Pluskomma λ)-EA on multimodal fitness landscapes. In: Proceedings of the 7th annual conference on genetic and evolutionary computation. GECCO ‘05. New York: ACM; 2005. p. 865–72.
115. Epitropakis MG, Li X, Burke EKA. Dynamic archive niching differential evolution algorithm for multimodal optimization. In: 2013 I.E. congress on evolutionary computation; 2013. p. 79–86.
116. Molina D, Puris A, Bello R, Herrera F. Variable mesh optimization for the 2013 CEC special session niching methods for multimodal optimization. In: 2013 I.E. congress on evolutionary computation; 2013. p. 87–94.
117. Fieldsend JE. Using an adaptive collection of local evolutionary algorithms for multi-modal problems. Soft Comput. 2015;19(6):1445–60.
118. Ahrari A, Deb K, Preuss M. Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations. Evol Comput. 2016 Apr:1–33.
119. Beyer HG, Sendhoff B. Covariance matrix adaptation revisited—the CMSA evolution strategy. Lect Notes Comput Sci. 2008:123–32.

120. Fieldsend JE. Running up those hills: multi-modal search with the niching migratory multi-swarm optimiser. In: 2014 I.E. Congress on Evolutionary Computation (CEC); 2014. p. 2593–2600.
121. Chen S, Montgomery J, Röhler AB. Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. Appl Intell. 2015;42(3):514–26.
122. Tang K, Yao X, Suganthan PN, MacNish C, Chen YP, Chen CM, et al. Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. In: China; 2007.
123. Lozano M, Molina D, Herrera F. Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15:2085–7.
124. Tang K. Summary of results on CEC'08 competition on large scale global. Optimization. 2008;
125. Wang Y, Li BA. Restart univariate estimation of distribution algorithm: sampling under mixed Gaussian and Lévy probability distributions. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 3917–24.
126. Brest J, Zamuda A, Boskovic B, Maucec MS, Zumer V. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 2032–9.
127. Hsieh ST, Sun TY, Liu CC, Tsai SJ. Solving large scale global optimization using improved particle swarm optimizer. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 1777–84.
128. Zhao SZ, Liang JJ, Suganthan PN, Tasgetiren MF. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 3845–52.
129. MacNish C, Yao X. Direction matters in high-dimensional optimisation. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 2372–9.
130. Zamuda A, Brest J, Bošković B, Zumer V. Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 3718–25.
131. Yang Z, Tang K, Yao X. Multilevel cooperative coevolution for large scale optimization. In: 2008 I.E. congress on evolutionary computation (CEC 2008); 2008. p. 1663–70.
132. Tang K, Li X, Suganthan P, Yang Z, Weise T. Benchmark functions for the CEC'2010 special session and competition on large scale global. Optimization. 2010;
133. Molina D, Lozano M, Herrera F. MA-SW-chains: memetic algorithm based on local search chains for large scale continuous global optimization. 2010 I.E. Congress on Evolutionary Computation (CEC 2010). 2010:3153–60.
134. Wang Y, Li B. Two-stage based ensemble optimization for large-scale global optimization. In: 2010 I.E. congress on evolutionary computation (CEC 2010). Barcelona; 2010. p. 4488–4495.
135. Korosec P, Tashkova K, Silc J. The differential ant-Stigmergy algorithm for large-scale global optimization. In: 2010 I.E. congress on evolutionary computation (CEC 2010); 2010. p. 4288–95.
136. Brest J, Zamuda A, Fister I, Maucec MS. Large scale global optimization using self-adaptive differential evolution algorithm. In: 2010 I.E. congress on evolutionary computation (CEC 2010); 2010. p. 3097–104.
137. Omidvar MN, Li X, Yao X. Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In: 2010 I.E. congress on evolutionary computation (CEC 2010); 2010. p. 1762–9.
138. Wang H, Wu Z, Rahnamayan S, Jiang D. Sequential DE enhanced by neighborhood search for large scale global optimization. In: 2010 I.E. congress on evolutionary computation (CEC 2010); 2010. p. 4056–62.
139. Zhao SZ, Suganthan PN, Das S. Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search. In: 2010 I.E. Congress on Evolutionary Computation (CEC 2010). 2010:1983–90.
140. Brest J, Maučec MS. Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2157–74.
141. García-Martínez C, Rodríguez FJ, Lozano M. Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11): 2109–26.
142. Wang H, Wu Z, Rahnamayan S. Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2127–40.
143. Weber M, Neri F, Tirronen V. Shuffle or update parallel differential evolution for large-scale optimization. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2089–107.
144. Yang Z, Tang K, Yao X. Scalability of generalized adaptive differential evolution for large-scale continuous optimization. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15:2141–55.
145. Zhao SZ, Suganthan PN, Das S. Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2175–85.
146. García-Nieto J, Alba E. Restart particle swarm optimization with velocity modulation: a scalability test. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11): 2221–32.
147. Montes de Oca MA, Aydın D, Stützle T. An incremental particle swarm for large-scale continuous optimization problems: an example of tuning-in-the-loop (re) design of optimization algorithms. Soft computing—a fusion of foundations. Methodologies and Applications. 2011;15:2233–55.
148. Duarte A, Martí R, Gortazar F. Path relinking for large-scale global optimization. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2257–73.
149. Gardeux V, Chelouah R, Siarry P, Glover F. EM323: a line search based algorithm for solving high-dimensional continuous non-linear optimization problems. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11): 2275–85.
150. Molina D, Lozano M, Sánchez AM, Herrera F. Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2201–2220.
151. Neumaier A, Fendl H, Schilly H, Leitner T. VXQR: derivative-free unconstrained optimization based on QR factorizations. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;15(11):2287–98.
152. Puris A, Bello R, Molina D, Herrera F. Variable mesh optimization for continuous optimization problems. Soft Computing—A Fusion of Foundations, Methodologies and Applications. 2011;16(3):511–25.
153. LaTorre A, Muelas S, Peña JM. Multiple offspring sampling in large scale global optimization. In: 2012 I.E. congress on evolutionary computation (CEC 2012). Brisbane, Australia; 2012. p. 964–71.

154. Brest J, Bošković B, Zamuda A, Fister I, Maučec MS. Self-adaptive differential evolution algorithm with a small and varying population size. In: 2012 I.E. congress on evolutionary computation (CEC 2012). Brisbane; 2012. p. 1–8.
155. Zhang K, Li B. Cooperative coevolution with global search for large scale global optimization. In: 2012 I.E. congress on evolutionary computation (CEC 2012). Brisbane; 2012. p. 1–7.
156. Takahama T, Sakai S. Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In: 2012 I.E. congress on evolutionary computation (CEC 2012); 2012. p. 1–8.
157. Fister I, Fister Jr I, Brest J, Zumer JB. Memetic artificial bee colony algorithm for large-scale global optimization. In: 2012 I.E. congress on evolutionary computation (CEC 2012); 2012. p. 1–8.
158. Li X, Tang K, Omidvar MN, Yang Z, Quin K. Benchmark functions for the CEC'2013 special session and competition on large scale global optimization: RMIT University; 2013.
159. LaTorre A, Muelas S, Peña JM. Large scale global optimization: experimental results with MOS-based hybrid algorithms. In: 2013 I.E. congress on evolutionary computation (CEC 2013). Cancún, Mexico; 2013. p. 2742–9.
160. Wei F, Wang Y, Huo Y. Smoothing and auxiliary functions based cooperative coevolution for global optimization. In: 2013 I.E. congress on evolutionary computation (CEC 2013); 2013. p. 2736–41.
161. Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. Inf Sci. 2008;178(15):2985–99.
162. Liu J, Tang K. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. Birmingham. 2013;
163. López ED, Puris A, Bello RR. VMODE: a hybrid metaheuristic for the solution of large scale optimization problems. Revista Investigación Operacional. 2015;36(3):232–9.
164. Mahdavi S, Shiri ME, Rahnamayan S. Cooperative co-evolution with a new decomposition method for large-scale optimization. In: 2014 I.E. congress on evolutionary computation (CEC 2014); 2014. p. 1285–92.
165. Ye S, Dai G, Peng L, Wang M. A hybrid adaptive coevolutionary differential evolution algorithm for large-scale optimization. In: 2014 I.E. congress on evolutionary computation (CEC 2014); 2014. p. 1277–84.
166. Wei F, Wang Y, Zong T. Variable grouping based differential evolution using an auxiliary function for large scale global optimization. In: 2014 I.E. congress on evolutionary computation (CEC 2014); 2014. p. 1293–8.
167. Molina D, Herrera F. Iterative hybridization of DE with local search for the CEC'2015 special session on large scale global optimization. In: 2015 I.E. Congress on Evolutionary Computation (CEC 2015). 2015:1974–8.
168. Miranda V, Alves R. Differential Evolutionary Particle Swarm Optimization (DEEPSO): a successful hybrid. In: 2013 BRICS Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC 2013). 2013:368–74.
169. Dai G, Chen X, Chen L, Wang M, Peng L. Cooperative coevolution with dependency identification grouping for large scale global optimization. In: 2016 I.E. congress on evolutionary computation (CEC 2016); 2016. p. 5201–8.
170. Salcedo-Sanz S, Camacho-Gómez C, Molina D, Herrera F. A Coral reefs optimization algorithm with substrate layers and local search for large scale global optimization. In: 2016 I.E. congress on evolutionary computation (CEC 2016); 2016. p. 3574–81.
171. Mahdavi S, Rahnamayan S, Deb K. Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization. In: 2016 I.E. congress on evolutionary computation (CEC 2016); 2016. p. 3357–565.
172. Omidvar MN, Kazimipour B, Li X, Yao X. CBCC3—a contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance. In: 2016 I.E. congress on evolutionary computation (CEC 2016); 2016. p. 3541–8.
173. Cano A, García-Martínez C. 100 million dimensions large-scale global optimization using distributed GPU computing. In: 2016 I.E. congress on evolutionary computation (CEC 2016); 2016. p. 3566–73.
174. Goh SK, Tan KC, Mamun AA, Abbass HA. Evolutionary big optimization (BigOpt) of signals. In: IEEE congress on evolutionary computation, CEC 2015, Sendai, Japan. May 25–28, 2015. 2015:3332–9.
175. Zhang Y, Zhou M, Jiang Z, Liu J. A multi-agent genetic algorithm for big optimization problems. IEEE Congress on Evolutionary Computation (CEC); 2015. p. 2015:703–7.
176. Omidvar MN, Li X, Mei Y, Yao X. Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans Evol Comput. 2014 June;18(3):378–93.
177. LaTorre A, Muelas S, Peña JMA. Comprehensive comparison of large scale global optimizers. Inf Sci. 2014;316:517–49.
178. Tang Q, Shen Y, Hu C, Zeng J, Gong W. Swarm intelligence: based cooperation optimization of multi-modal functions. Cogn Comput. 2013;5(1):48–55.
179. Kennedy J, Eberhart RC. Particle swarm optimization. IEE International Conference on Neural Networks. 1995:1942–8.
180. Yang XS. Bat algorithm: literature review and applications. International Journal of Bio-Inspired Computation 2013 08. 2013;5:141–9.
181. Apostolopoulos T, Vlachos A. Application of the firefly algorithm for solving the economic emissions load dispatch problem. Hindawi publishing corporation international. J Comb. 2011;23:1–23.
182. Amiri B, Hossain L, Crawford JW, Wigand RT. Community detection in complex networks: multi–objective enhanced firefly algorithm. Knowl-Based Syst. 2013;46:1–11.
183. Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. Adv Eng Softw. 2017;105:30–47.
184. An J, Kang Q, Wang L, Wu Q. Mussels wandering optimization: an ecologically inspired algorithm for global optimization. Cogn Comput. 2013;5:188–99.
185. Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 1996 Feb;26(1):29–41.
186. Dorigo M, Stützle T. Ant colony optimization. MIT Press; 2004.
187. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim. 2007;39(3):459–71.
188. Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput & Applic. 2016 May;27(4):1053–73.
189. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61.
190. Kaveh A, Farhoudi N. A new optimization method: dolphin echolocation. Adv Eng Softw. 2013;59:53–70.
191. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95:51–67.
192. Salcedo-Sanz S, Casanova-Mateo C, Pastor-Sánchez A, Sánchez-Girón M. Daily global solar radiation prediction based on a hybrid coral reefs optimization—extreme learning machine approach. Sol Energy. 2014;105:91–8.
193. Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul. 2012;17(12):4831–45.

194. Timmis J, Andrews P, Hart E. On artificial immune systems and swarm intelligence. Swarm Intelligence. 2010;4:247–73.
195. Kim SS, McLoone S, Byeon JH, Lee S, Liu H. Cognitively inspired artificial bee colony clustering for cognitive wireless sensor networks. Cogn Comput. 2017 Apr;9(2):207–24.
196. Ong YS, Gupta A. Evolutionary multitasking: a computer science view of cognitive multitasking. Cogn Comput. 2016;8(2):125–42.
197. Javed SG, Majid A, Ali S, Kausar N. A bio-inspired parallel-framework based multi-gene genetic programming approach to Denoise biomedical images. Cogn Comput. 2016;8(4):776–93.
198. Zeng N, Wang Z, Zhang H, Alsaadi FE. A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay. Cogn Comput. 2016;8(2):143–52.
199. Song B, Wang Z, Zou L. On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. Cogn Comput. 2017;9(1):5–17.
200. Siddique N, Adeli H. Nature-inspired chemical reaction optimisation algorithms. Cogn Comput. 2017;9(4):411–22.
201. Wu T, Yao M, Yang J. Dolphin swarm extreme learning machine. Cogn Comput. 2017;9(2):275–84.
202. Solé R, Amor DR, Duran-Nebreda S, Conde-Pueyo N, Carbonell-Ballestero M, Montañez R. Synthetic collective intelligence. Biosystems. 2016;148:47–61. What Synthetic Biology can offer to Artificial Intelligence
203. Sörensen K. Metaheuristics—the metaphor exposed. Int Trans Oper Res. 2015;22:3–18.
204. García-Martínez C, Gutiérrez PD, Molina D, Lozano M, Herrera F. Since CEC 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis's weakness. Soft Comput. 2017:1–11.
205. Liao T, Molina D, Stützle T. Performance evaluation of automatically tuned continuous optimizers on different benchmark sets. Appl Soft Comput. 2015;27:490–503.