

How Does the Number of Objective Function Evaluations Impact Our Understanding of Metaheuristics Behavior?

ANEZKA KAZIKOVA^{ID}, MICHAL PLUHACEK^{ID}, (Member, IEEE),
AND ROMAN SENKERIK^{ID}, (Member, IEEE)

Faculty of Applied Informatics, Tomas Bata University in Zlin, 76001 Zlin, Czech Republic

Corresponding authors: Anezka Kazikova (kazikova@utb.cz) and Roman Senkerik (senkerik@utb.cz)

This work was supported in part by the Internal Grant Agency of Tomas Bata University under Project IGA/CebiaTech/2021/001, and in part by the resources of AI Laboratory, Faculty of Applied Informatics, Tomas Bata University in Zlin.

ABSTRACT Comparing various metaheuristics based on an equal number of objective function evaluations has become standard practice. Many contemporary publications use a specific number of objective function evaluations by the benchmarking sets definitions. Furthermore, many publications deal with the recurrent theme of late stagnation, which may lead to the impression that continuing the optimization process could be a waste of computational capabilities. But is it? Recently, many challenges, issues, and questions have been raised regarding fair comparisons and recommendations towards good practices for benchmarking metaheuristic algorithms. The aim of this work is not to compare the performance of several well-known algorithms but to investigate the issues that can appear in benchmarking and comparisons of metaheuristics performance (no matter what the problem is). This article studies the impact of a higher evaluation number on a selection of metaheuristic algorithms. We examine the effect of a raised evaluation budget on overall performance, mean convergence, and population diversity of selected swarm algorithms and IEEE CEC competition winners. Even though the final impact varies based on current algorithm selection, it may significantly affect the final verdict of metaheuristics comparison. This work has picked an important benchmarking issue and made extensive analysis, resulting in conclusions and possible recommendations for users working with real engineering optimization problems or researching the metaheuristics algorithms. Especially nowadays, when metaheuristic algorithms are used for increasingly complex optimization problems, and meet machine learning in AutoML frameworks, we conclude that the objective function evaluation budget should be considered another vital optimization input variable.

INDEX TERMS Evolutionary computation, computational intelligence, performance evaluation, cost function, optimization, benchmark testing, optimization methods.

I. INTRODUCTION

Metaheuristic optimization has become a trendy topic over the last few decades. It concerns a wide range of applications from continuous single objective problems to discrete optimization tasks such as the Travelling Salesman Problem, engineering applications, circuits design, or scheduling [1]–[3].

There is a close connection between metaheuristics and benchmark testing. Metaheuristics typically prove their efficiency on a set of test problems, which should be diverse

The associate editor coordinating the review of this manuscript and approving it for publication was Yanbo Chen^{ID}.

and unbiased [4]. In this context, the official benchmarking testbeds (e.g., the IEEE CEC benchmarks) provide many benefits. They include problems with various characteristics. The problem selection is already predefined, not formed by the competence of the promoted algorithm. And finally, they offer performance measures and evaluation rules. As a result, benchmark testbeds' influence goes beyond the competitions, as they are becoming the etalon of current optimization practice.

One of the significant metaheuristic struggles is stagnation, in which the algorithms can no longer create a better solution to the solved problem. This phenomenon may appear anytime but usually happens at the end of the convergence curves,

depicting the mean solutions' quality based on the number of objective function evaluations (FEs) or corresponding iterations (see, e.g., [5]–[17]).

In theory, a moment should arise in which most metaheuristics will not improve the best-found solution. Therefore, all metaheuristics may be doomed to end their optimization process in stagnation, though hopefully in the desirable global optimum.

However, since many convergence curves end in such a stagnation manner (and given the often employed benchmark FEs recommendation practice), it may lead to the impression that further optimization with more evaluations may be pointless. But is it?

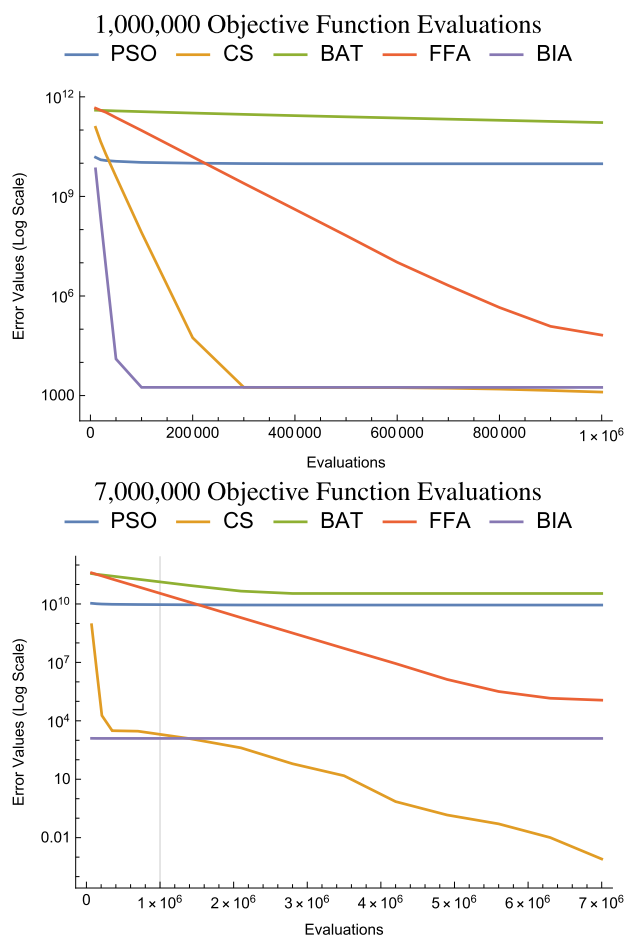


FIGURE 1. Convergence of the mean population error values ending in apparent stagnation in recommended objective function evaluation budget scenario (above) and increased budget (below) (CEC 2015 Function 2 - 100D).

A. MOTIVATION

During our research, we came upon several cases where some algorithms broke through apparent stagnation when the computation budget increased. Figure 1 presents an instance of such a case. The figure depicts the mean convergence of 51 runs on the standard number of FEs and the respective raised budget. Despite the stagnation that lasted for more than 700,000 FEs, some algorithms managed to invert the trend. Since this was not an isolated case, we decided to investigate

the influence of FEs budget on the performance and inner dynamics of selected metaheuristics.

Namely:

- the overall performance of selected optimization algorithms,
- the success (or failure) of particular optimization algorithms, especially when compared to other optimization techniques,
- their convergence curve,
- and the population diversity of the algorithms.

B. RESEARCH IMPORTANCE EXPLANATION

The goal of this article is *not* to compare several selected algorithms and it is *not* to present the comprehensive performance comparisons of well known algorithms, but to investigate the influence of FEs budget on the final interpretation of the results. The aim here is to examine the issues that can appear in benchmarking and comparisons of metaheuristics performance (no matter what the problem is).

Recently, many challenges, issues, questions have been raised regarding fair comparisons, good practices for benchmarking of metaheuristic algorithms [18], more in-depth insights into results, statistics, analysis of behavioral patterns, and resulting recommendations [19], [20]. Best practices in benchmarking represent a significant problem nowadays, when metaheuristic algorithms are used for increasingly complex optimization problems, and for the evolution of deep learning architectures in AutoML frameworks [21], [22]. Therefore professional organizations, like IEEE, and researchers have recently established benchmarking task-force¹ and networks.²

This article has picked an important benchmarking issue and made extensive analysis, resulting in conclusions and possible recommendations for users. Our hope and ambition for this work are to benefit and inspire both real-world applications and the future direction of benchmark profiling.

The paper is structured as follows: Section II sums up the current evaluation practice, investigates the most used number of FEs, and examines alternative performance measures. Section III briefly introduces the metaheuristic algorithms further examined in the experiments and describes the parameter tuning process. Section IV presents and partially discusses the results of the experiment. It compares the performance, convergence, and population diversity given two FEs budget scenarios first on a selection of swarm-based algorithms, and then analyses the impact on the effective optimizers, that won the examined IEEE CEC benchmark competitions. Section V discusses the results from a broader perspective. And finally, Section VI concludes the meaning of our findings for future research and practice.

II. IS THERE ANY STANDARD EVALUATIONS PRACTICE?

Many metaheuristic publications examine the proposals or modifications of existing algorithms, which leads to

¹<https://cmte.ieee.org/cis-benchmarking/>
²<https://sites.google.com/view/benchmarking-network/home>

an inevitable comparison between the algorithms' performance or their respective versions. The requirement of an equal number of objective function evaluations has become the standard practice. It offers several advantages:

- A predefined number of FEs defines a clear and straightforward termination condition
- The majority of publications use this approach
- It may provide a fairer condition than the alternative iteration-based comparison

The last point revolves around the changeable nature of metaheuristic designs. While some algorithms evaluate the population only once per iteration, others (for instance, hybrid algorithms that merge more metaheuristics into one) may do the same several times during a single iteration loop. In such a case, comparing these algorithms based on an equal set of iterations may be biased [23], [24]. Still, an iteration-based comparison is a standard practice of many publications (see, e.g., [5]–[9]).

The sole fulfillment of equal FEs alone does not ensure fair comparisons. There is a general assumption that the objective function holds for the most complicated computation step in the metaheuristic optimization process. That is why the number of FEs sets a base for algorithmic comparison and the computational complexity approximation [25]. However, no one can guarantee this assumption. Some optimization methods may include even more complicated operations than the currently solved problem defined by the objective function [24], [26].

Another important objection is that FEs equality is not the only necessary condition to a fair algorithmic comparison. A wide range of input variables impact the optimization process as well as the chance on an objective comparison: including the parameter settings, computational capabilities, used programming language, or the population size, as pointed out in [24], [26].

Although the use of evaluation budget is a common practice (e.g., in [10]–[15]), to the best of our knowledge, there is no rule limiting the "standard" maximum of FEs. The range of used evaluation limit reaches from 250 FEs [27], [28] over $10,000 \times D$ [29], to no limitation at all [30]. However, the benchmark test suites may serve as a nonnegligible inspiration source, and many publications adopt the recommended evaluation budgets from the benchmark definitions (see, e.g., [10]–[14], [31]).

Table 1 provides an overview of the IEEE CEC optimization benchmark testbeds and the corresponding limits of objective function evaluations. The IEEE CEC benchmarks range the FEs limit from $50 \times D$ to 10,000,000 evaluations [32], [33]. The difference lies in the problem domain specification. The high complexity of some real-world problems indirectly boosted the development of optimization methods designed to work well with a small number of FEs [27], [34]–[36]. Computationally expensive problems hence may be limited by a lower evaluation budget than easily solvable functions.

The comparison of metaheuristics comes hand in hand with various performance measures. One of the essential foundations of the experiments is the definition of the termination condition for every algorithm. The primary termination conditions include the maximum number of iterations, evaluations, or predefined execution time. These conditions can also be further advanced to limit the maximum iterations without improvement or reaching the desired objective function value [54]. Apart from the budget limitation, some benchmark recommendations also suggest additional termination condition of an error rate lower than a predefined threshold, for example, set to E^{-8} [55].

The performance measures often include the mean statistical error of the best, worst, mean and median solutions, and their standard deviations. However, since the No Free Lunch theorem [55] states, there is no "universal" best performing algorithm to solve any possible problem, solely performance-oriented experiments usually cannot lead to general assumptions. An utter win of one algorithm on one set of problems does not mean that the algorithm would be usable on a different set. The comparison experiments need to deploy further insights of the algorithms' inner dynamics to support its future usability. Such performance measures investigate, e.g., the population diversity, the algorithms' robustness, or the convergence curves.

A vast number of convergence curves indicate later stagnation [5]–[17], which may lead to a (possibly false) impression that raising the FEs budget would lead to a redundant computational effort. Yet several studies imply that the maximum number of FEs affects the overall performance of the algorithms [19], [26], [56], [57]. In the Particle Swarm Optimization Evaluations study [26], Engelbrecht suggests that a large number of evaluations may not be beneficial to small populations due to the premature stagnation.

In the proposal of the Passing Vehicle Search algorithm [56], the authors (Savsani and Savsani, 2016) compare the proposed algorithm on 13 engineering applications. However, since the results of other algorithms were taken from broad literature, all the compared algorithms used different FEs limit. To provide a fair comparison, the authors executed the Passing Vehicle Search algorithm with the corresponding FEs budget to many of the respective scenarios. This experiment revealed a slight improvement of the mean and worst solutions with a higher evaluation budget.

In the Conceptual Comparison of Several Metaheuristics [57], the author (Ezugwu, 2020) shows the mean best solution error with a variable FEs limit. The results differ based both on the solved problems and the solving algorithm. Another noteworthy publication [19] (LaTorre, 2020) presents that the average ranks of compared algorithms differ significantly, given a various number of FEs.

This article investigates five swarm algorithms and two benchmark competition winners on two sets of IEEE CEC benchmark testbeds with two evaluation budgets. We aim to uncover the effect of the FEs limit on population diversity, convergence, and the algorithms' overall performance.

TABLE 1. IEEE CEC benchmark test suites and their respective function evaluation budget recommendations.

Benchmark	Problem Specification	Function Evaluation Budget
CEC 2005	[37] Real Parameter Single Objective	$10,000 \times D$
CEC 2006	[38] Constrained Real Parameter Single Objective	500,000
CEC 2007	[39] Real-Parameter Multi-Objective Optimization Algorithms	500,000
CEC 2008	[40] Large Scale Single Objective Global Optimization with Bound Constraints	$5,000 \times D$
CEC 2009	[41] Dynamic Optimization	$10,000 \times D$ ($Max_{FES}/change$)
CEC 2010	[42] Constrained / Bound Constrained Multi-Objective Optimization Algorithms	300,000
	[43] Large-scale Single Objective Global Optimization with Bound Constraints	3,000,000
	[44] Evolutionary Constrained Real Parameter Single Objective Optimization	$20,000 \times D$
CEC 2011	[45] Real-world Numerical Optimization Problems	150,000
CEC 2013	[46] Real Parameter Single Objective Optimization	$10,000 \times D$
CEC 2014	[47] Single Objective Real-Parameter Numerical Optimization	$10,000 \times D$
CEC 2015	[32] Computationally Expensive Single Objective Numerical Optimization	$50 \times D$
	[48] Bound Constrained Single-Objective Computationally Expensive Numerical Optimization	$50 \times D$
	[49] Learning-based Real-Parameter Single Objective Optimization	$10,000 \times D$
CEC 2016	[50] Single Objective Multi-Niche Optimization	$2000 \times D \times \sqrt{q}$
	[32] Computationally Expensive Single Objective Numerical Optimization	$50 \times D$
	[49] Learning-based Real-Parameter Single Objective Optimization	$10,000 \times D$
CEC 2017	[50] Single Objective Multi-Niche Optimization	$2000 \times D \times \sqrt{q}$
	[47] Single Objective Real-Parameter Numerical Optimization	$10,000 \times D$
	[29] Single Objective Real-Parameter Bound Constrained Numerical Optimization	$10,000 \times D$
CEC 2018	[51] Constrained Real-Parameter Optimization	$20,000 \times D$
	[48] Bound Constrained Single-Objective Computationally Expensive Numerical Optimization	$50 \times D$
	[29] Single Objective Real-Parameter Bound Constrained Numerical Optimization	$10,000 \times D$
CEC 2019	[51] Constrained Real-Parameter Optimization	$20,000 \times D$
	[48] Bound Constrained Single-Objective Computationally Expensive Numerical Optimization	$50 \times D$
	[30] 100 Digit Challenge Single Objective Numerical Optimization	No limit (termination with 10-digit accuracy, or when no improvement possible)
CEC 2020	[52] Multimodal Multiobjective Optimization	$10,000 \times n_{ops}$ 50,000 <i>if</i> $D = 5$
	[33] Single Objective Bound Constrained Numerical Optimization	1,000,000 <i>if</i> $D = 10$ 3,000,000 <i>if</i> $D = 15$ 10,000,000 <i>if</i> $D = 20$ 100,000 <i>if</i> $D \leq 10$ 200,000 <i>if</i> $10 < D \leq 30$ 400,000 <i>if</i> $30 < D \leq 50$ 800,000 <i>if</i> $50 < D \leq 15$ 1,000,000 <i>if</i> $150 < D$
	Where:	
	• D is the dimensionality of the problem,	
	• q is the goal optima number,	
	• and n_{ops} means the number of local and global Pareto optimal sets to be obtained.	

III. BRIEF DESCRIPTION OF COMPARED ALGORITHMS

To analyze the impact of a higher number of objective function evaluations, we compared the optimization performance of selected algorithms solving two benchmark sets of problems IEEE CEC 2015 and 2017. The examined swarm algorithms were: the Particle Swarm Optimization [58], the Cuckoo Search [59], the Bat Algorithm [60], the Firefly Algorithm [61], and the Bison Algorithm [62]. Further we analysed the impact of FEs budget on the winners of the CEC benchmark competitions: the L-SHADE with Eigenvector Crossover and Successful Parent-Selecting Framework [63], and the Effective Butterfly Optimizer with Covariance Matrix [64]. The following section briefly introduces the examined algorithms.

A. PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization algorithm (PSO) was proposed by Kennedy and Eberhart in 1995 [58]. It is by far the most popular swarm metaheuristic, with more than

55,000 publications in the Web of Science database.³ The algorithm found inspiration in the emerging behavior of bird flocks and fish swarms. There are two basic versions of the algorithm: local best and global best, distinguished by the solution's neighborhood topology [65]. This article studies PSO with the global best topology.

B. CUCKOO SEARCH OPTIMIZATION

The Cuckoo Search (CS) was developed by Yang and Deb in 2009 [59]. The algorithm simulates the egg-laying patterns of cuckoos. The algorithm offers the advantage of only two configurable parameters and has proved to be a mighty optimization tool.

C. BAT ALGORITHM

The Bat Algorithm (BAT) (Yang, 2010) simulates the echolocation ability of microbats [60]. The algorithm employs a

³In January 14th 2021, there were 55,132 publication records in the Web of Science database addressing the Particle Swarm Optimization topic.

frequency tuning mechanism, which, by extension, acts as a mutation factor – as it affects mainly local solutions [66].

D. FIREFLY ALGORITHM

The Firefly Algorithm (FFA) was designed by Yang in 2010 [61]. The algorithm simulates the courtship of fireflies. In the courting ritual, each firefly looks around, and when it finds a firefly shining brighter than itself, it moves closer. The most glowing firefly finally performs a movement in a random direction.

E. BISON ALGORITHM

The Bison Algorithm (BIA) (Kazikova, 2018) models bison herds' protecting and running behavior [67]. It divides the population into two groups. The first group exploits known solutions by moving them closer to the center of several fittest solutions. In contrast, the second group explores the search space to avoid the trap of local optima.

F. SPS L SHADE EIG

The L-SHADE with Eigenvector-Based Crossover and Successful Parent-Selecting Framework (SPS L SHADE EIG) was proposed by Guo *et al.* in 2015 [63]. The acronym stands for a Success-History based Adaptive Differential Evolution with a linear decrease of the population size (L-SHADE), Eigenvector-Based Crossover (EIG), and Successful Parent-Selecting Framework (SPS). It is a variant of a Differential Evolution, which implements the memories of previously successful parameters. The Successful Parent-Selecting Framework provides a way to overcome stagnation. The algorithm won the IEEE CEC 2015 competition [49].

G. EBO WITH CMAR

The Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase (EBO with CMAR) was proposed by Kumar *et al.* in 2017 [64]. It is a hybrid self-adaptive algorithm, which combines the features of global and local optimizers. The original Effective Butterfly Optimizer algorithm is enhanced with success-history based adaptation and linear population size reduction, while the Covariance Matrix Adapted Retreat Phase improves the local search. The algorithm won the IEEE CEC 2017 competition [29].

1) PARAMETER TUNING

To determine the appropriate parameters for the solved problems, we tested the algorithms in 10 and 30 dimensions on the IEEE CEC 2017 benchmark with a selection of recommended configurations from the following literature: [61], [68]–[76]. We investigated the statistical significance of the results ($p < 0.05$) with the Wilcoxon Rank-Sum test and the Friedman Rank test. Further experiments use the winning parameter configurations defined in Table 2. The Parameter Tuning Experiment was extended in [77].

The algorithms implementations were derived from the EvoloPy library [70] and [78]. All codes of the swarm

TABLE 2. Winning parameter configurations according to the Wilcoxon Rank-Sum test and the Friedman Rank test ($p < 0.05$).

Particle Swarm Optimization					
Source	NP	Wmax	Wmin	vmax	C1=C2
[70]	50	0.9	0.2	6	2
Bat Algorithm					
Source	NP	Loudness	Pulse rate	Qmin	Qmax
[70]	50	0.5	0.5	0	2
Firefly Algorithm					
Source	NP	Alpha	Gama	Beta	Lambda
[70]	50	0.5	1	0.2	1.0
Bison Algorithm					
Source	NP	Overstep	Elite group	Swarming group	
[79]	50	3.5	20	40	
Cuckoo Search					
Source	NP	Pa			
[70]	20	0.25			

optimizers are available and welcome to use at TBU A.I.Lab's GitHub repository.⁴ The codes of the competition winners: SPS L SHADE EIG and EBO with CMAR were adopted from the official CEC Benchmarking Github^{5,6} with the included tuned parameters.

IV. COMPARING ALGORITHMS WITH TWO OBJECTIVE FUNCTION EVALUATION BUDGETS

We compared the algorithms on two benchmark sets of IEEE CEC 2015 and 2017 [29], [49] in 10, 30, 50, and 100 dimensions in 51 runs. We examined two evaluation budgets: the **standard evals** scenario of $10,000 \times D$ FEs, as recommended in the Problem Definitions and Evaluation Criteria for both testbeds, and the **7 evals** scenario of $70,000 \times D$ FEs.

Our experiment first analyzed the impact on the swarm algorithms separately and then added the competition winning optimizers. Separating these two analyses allowed for an easier recognition of the FEs limit effect, as the influence of the higher budget is more visible on algorithms of comparable performance, and it is less complicated to detect a renewed convergence on a smaller range of error values.

We studied the influence of the FEs budget on the algorithms' inner dynamics. Therefore, we assumed that the choice of the algorithms should not be substantial, supposing, that what works on elementary optimizers may be considered even with more advanced optimization techniques. To confirm (or disprove) this hypothesis, we analyzed the competition testbeds' winning algorithms with a higher FEs budget subsequently.

All the swarm algorithms started with a randomly generated initial population. They were programmed in the same environment, using the same programming language, with parameters based on the Parameter Tuning Experiment.

⁴<https://github.com/TBU-AILab/Bison-Algorithm>

⁵<https://github.com/P-N-Suganthan/CEC2017-BoundConstrained>

⁶<https://github.com/P-N-Suganthan/CEC2015-Learning-Based>

The following sections first examined the swarm algorithms for the mean solution errors with statistical tests in Section IV-A, the convergence curves in Section IV-B, and the population diversities in Section IV-C. Finally, Section IV-D examined the impact on the competition winners. The experiments first compare the tested scenarios separately and then investigate the difference that the bigger evaluation budget caused.

A. OVERALL PERFORMANCE (MEAN SOLUTION ERROR)

This section investigates the algorithms' overall performance by comparing the solution errors with the Wilcoxon Rank-Sum tests ($p < 0.05$). The test determines on how many problems an algorithm significantly outperformed the others.

We compared the results across the five swarm algorithms in one of the scenarios – standard evaluations in Tables 3 and 4 and 7 evals in Tables 5 and 6. Then we investigated the impact of the different evaluation budgets. Table 7 and Table 8 examine the difference in the interpretation of the previous results. Figure 2 shows the ranks computed by the Friedman Rank test ($p < 0.05$), which compared all the swarm algorithms in both scenarios on the IEEE CEC 2015 testbed. The third column presents the difference in ranks for each algorithm.

TABLE 3. Winning algorithms on IEEE CEC 2015 benchmark (Wilcoxon, $p < 0.05$) in standard evals scenario.

	None	PSO	CS	BAT	FFA	BIA
10 dimensions	6	0	5	1	0	3
30 dimensions	4	0	3	1	2	5
50 dimensions	5	0	3	1	2	4
100 dimensions	4	0	0	1	5	5
Sum of wins	19	0	11	4	9	17

TABLE 4. Winning algorithms on IEEE CEC 2017 benchmark (Wilcoxon, $p < 0.05$) in standard evals scenario.

	None	PSO	CS	BAT	FFA	BIA
10 dimensions	8	0	13	0	2	7
30 dimensions	3	1	6	0	6	14
50 dimensions	7	1	5	0	7	10
100 dimensions	6	1	2	0	10	11
Sum of wins	24	3	26	0	25	42

TABLE 5. Winning algorithms on IEEE CEC 2015 benchmark (Wilcoxon, $p < 0.05$) in 7 evals scenario.

	None	PSO	CS	BAT	FFA	BIA
10 dimensions	5	1	8	0	0	1
30 dimensions	4	0	5	0	1	5
50 dimensions	3	0	5	0	2	5
100 dimensions	2	0	2	0	4	7
Sum of wins	14	1	20	0	7	19

Finally, we evaluated the individual impact of the bigger evaluation budget on every tested algorithm separately. The Wilcoxon pair-wise Rank-Sum tests ($p < 0.05$) comparing the 7 evals and standard evals scenarios of one particular algorithm are in Tables 9 and 10. The benefit is computed by Eq. 1, which subtracts the sum of 7 evals wins – standard

TABLE 6. Winning algorithms on IEEE CEC 2017 benchmark (Wilcoxon, $p < 0.05$) in 7 evals scenario.

	None	PSO	CS	BAT	FFA	BIA
10 dimensions	4	0	22	0	3	1
30 dimensions	5	0	9	0	4	12
50 dimensions	7	0	6	0	5	12
100 dimensions	6	0	4	0	6	14
Sum of wins	22	0	41	0	18	39

TABLE 7. Impact of higher evaluation budget on the IEEE CEC 2015 benchmark: compared the Wilcoxon Rank-Sum test results in Table 5-Table 3 [7 evals – standard evals] (Positive values mean a beneficial effect of higher evaluation number on the algorithm's final interpretation of results).

	None	PSO	CS	BAT	FFA	BIA
10 dimensions	-1	1	3	-1	0	-2
30 dimensions	0	0	2	-1	-1	0
50 dimensions	-2	0	2	-1	0	1
100 dimensions	-2	0	2	-1	-1	2
Sum of gains	-5	1	9	-4	-2	2

TABLE 8. Impact of higher evaluation budget on the IEEE CEC 2017 benchmark: comparing the Wilcoxon Rank-Sum test results in Table 6-Table 4 [7 evals – standard evals].

	None	PSO	CS	BAT	FFA	BIA
10 dimensions	-4	0	9	0	1	-6
30 dimensions	2	-1	3	0	-2	-2
50 dimensions	0	-1	1	0	-2	2
100 dimensions	0	-1	2	0	-4	3
Sum of gains	-2	-3	15	0	-7	-3

evals wins from the Wilcoxon Rank-Sum tests showed in Tables 16 and 17. The percentual benefit presented on the last line sums the percentage of problems positively affected by the higher evaluation budget (Eqs. 1, 2).

$$B_D = \frac{\sum_{i=1}^{F_{max}} W_i - \sum_{i=1}^{F_{max}} w_i}{F_{max}} \quad (1)$$

$$Benefit = \frac{B_{10D} + B_{30D} + B_{50D} + B_{100D}}{4 \times F_{max}} \times 100\% \quad (2)$$

where:

- B_D represents the benefit of the 7 evals scenario against the standard evaluations scenario,
- D presents the examined dimension,
- F_{max} is the number of problems in the benchmark test bed,
- W_i and w_i stand for the number of wins from the Wilcoxon Rank-Sum test from the 7 evals and standard evals scenarios respectively (from Tables 16 and 17),
- $Benefit$ represents the percentage of positively affected problems by higher evaluation budget,
- and B_{10D} presents the benefit computed by Eq. 1 on 10-dimensional benchmark problems.

Similarly, Figure 3 shows the pair-wise Friedman Rank test ($p < 0.05$), which compared the evaluation scenarios on every swarm algorithm individually. All the difference oriented experiments were computed by subtracting the 7 evals – standard evals results.

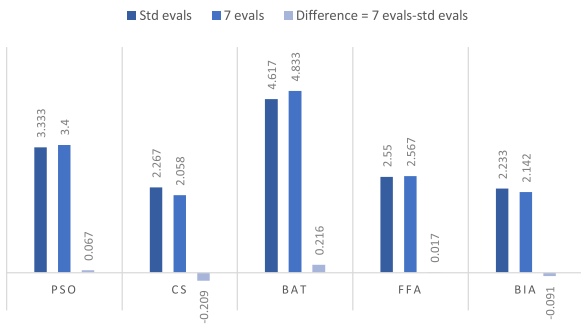


FIGURE 2. Friedman Rank test comparing all algorithms in two scenarios across all dimensions on CEC 2015.

OVERALL PERFORMANCE DISCUSSION

The comparison of the swarm algorithms revealed, that the higher evaluation budget strongly affected the interpretation of the Wilcoxon Rank-Sum tests (Tables 5, 6). In the standard evaluation scenario, this discussion would portray the success of the Bison Algorithm. Be that as it may, the higher evaluation budget flipped the verdict in favor of the Cuckoo Search algorithm, particularly in lower dimensions, and the final summary of Wilcoxon wins.

Tables 7 and 8 outline the extent of the general impact on the statistical test interpretation. The Cuckoo Search Algorithm enhanced its performance on most of the problems at the expense of other algorithms. The Bat Algorithm presents an exception since it was not affected in the IEEE CEC 2017 benchmark. The Friedman Rank test (Figure 2) confirmed a slight rank-wise improvement of the Cuckoo Search and the Bison Algorithm. The ranks of other algorithms dropped.

It is important to note that previous results compare only the final interpretations of the results with the statistical tests. The overall impact, concerning each algorithm on their own, was investigated in Tables 9 and 10. These tests compared the final results of both of the tested scenarios on a single algorithm, to estimate the benefit of higher evaluations for every individual algorithm separately. The results were then compared in Tables 9 and 10.

TABLE 9. Impact of higher evaluation budget on one algorithm based on the evaluation budget scenarios (pair-wise Wilcoxon, p < 0.05, on IEEE CEC 2015 benchmark) [7 evals – standard evals] (Positive values mean an overall beneficial effect of higher evaluation number solely on the algorithm’s performance).

	PSO	CS	BAT	FFA	BIA
10 dimensions	8	14	-9	14	10
30 dimensions	6	13	-2	14	11
50 dimensions	7	12	2	14	11
100 dimensions	7	13	5	14	10
Percentual benefit	47%	87%	-7%	93%	70%

Despite the previous findings, the effect of 7 evals budget was beneficial for most of the tested algorithms: mostly for the Firefly Algorithm (with 93% percentual benefit on CEC 2015) and the Cuckoo Search algorithm. This discovery

TABLE 10. Impact of higher evaluation budget on one algorithm based on the evaluation budget scenarios (pair-wise Wilcoxon, p < 0.05, on IEEE CEC 2017 benchmark) [7 evals – standard evals] (Positive values mean an overall beneficial effect of higher evaluation number solely on the algorithm’s performance).

	PSO	CS	BAT	FFA	BIA
10 dimensions	14	26	-15	22	12
30 dimensions	9	28	-5	20	13
50 dimensions	7	29	-1	18	14
100 dimensions	7	26	1	19	17
Percentual benefit	31%	91%	-17%	66%	47%

underlines the results in Tables 5 and 6 since almost all of the algorithms performed significantly better than in the standard evaluation scenario.

Unexpectedly, higher evaluations slightly harmed the Bat Algorithm, especially in the lower dimensions, whose percentual benefit ended up in the negative numbers. The Friedman Rank test in Figure 3 confirmed these findings. In the pair-wise comparison, every algorithm, with the BAT exception, significantly improved its rank in the higher evaluation scenario.

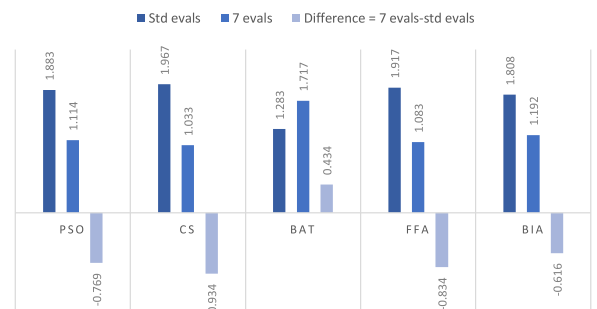


FIGURE 3. Friedman Rank test comparing one algorithm in two scenarios across all dimensions on CEC 2015.

B. CONVERGENCE

The convergence experiment investigated the development of the mean solution errors based on a varying number of evaluations. Figure 4 depicts the convergence curve of 15 100-dimensional problems of the IEEE CEC 2015 testbed with a higher evaluation budget. It also shows the standard evaluation threshold, which would be the last stop of the optimization, if it followed the benchmark recommendations [29], [49].

Figure 5 shows a selection of interesting cases in which the convergence of the standard evaluation budget ends in stagnation. Surprisingly, further evaluations provided an unprecedented improvement in the mean error.

It should be noted, that the convergence curves (Figs. 4-6) were approximated from 14 error values according to the IEEE CEC 2017 benchmark set recommendation [29]. The error values were recorded after (0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0) × MaxFES for each run.

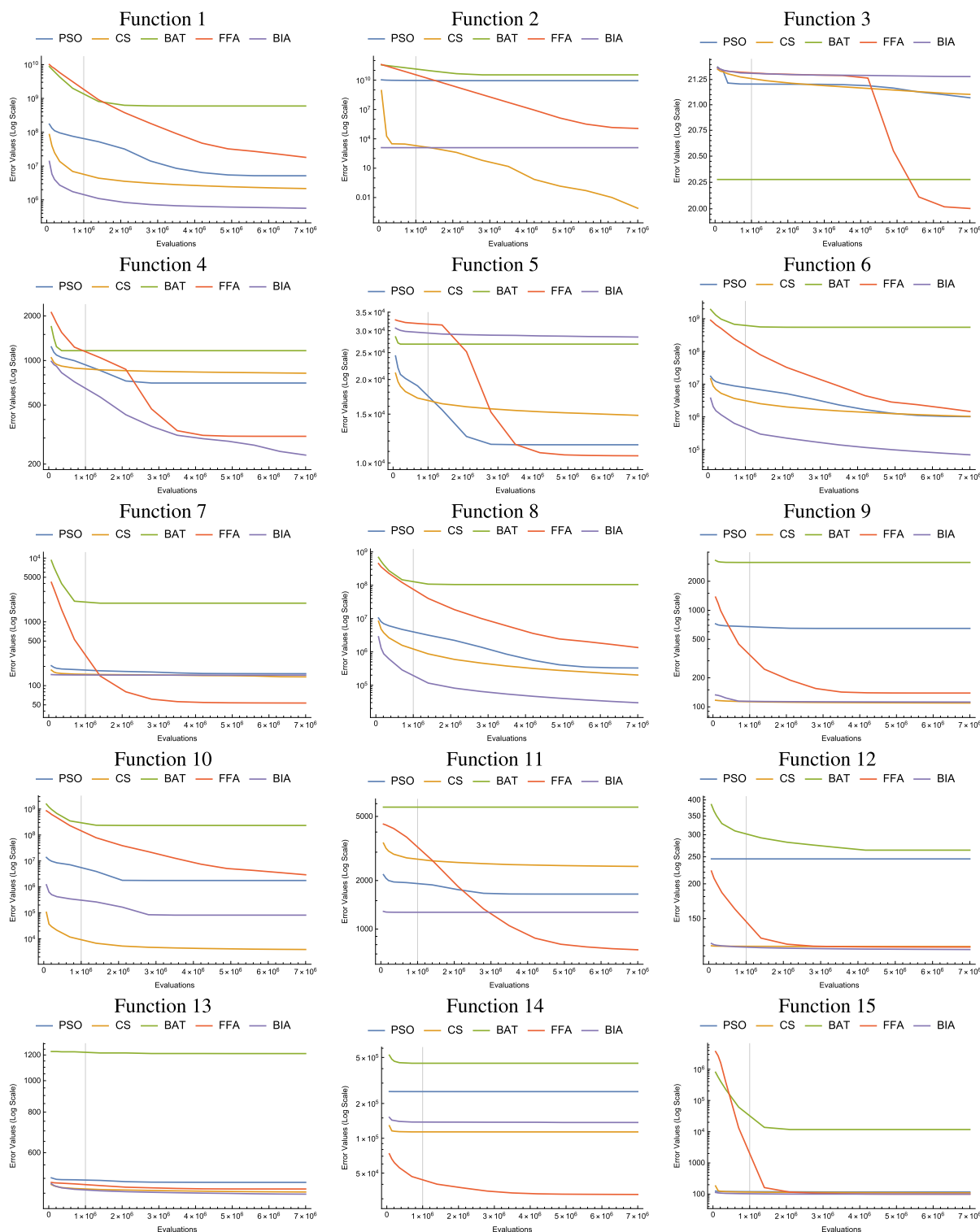


FIGURE 4. Convergence curves of 15 problems from the IEEE CEC 2015 benchmark in 100D with the 7 evals budget. The vertical line indicates where the standard evaluation budget would stop the optimization process.

1) CONVERGENCE DISCUSSION

Figure 4 shows a substantial drop of mean error value even after the standard evaluation threshold in most of the displayed cases. The only exception is Function 11, which stagnated for more than $6.5E^6$ FEs.

Figure 5 illustrates the potential benefit of higher evaluations, with some remarkable convergence twists. This figure emphasizes that even apparent stagnation does not necessarily mean that further optimization is pointless. This notably concerns the Cuckoo Search optimization but other

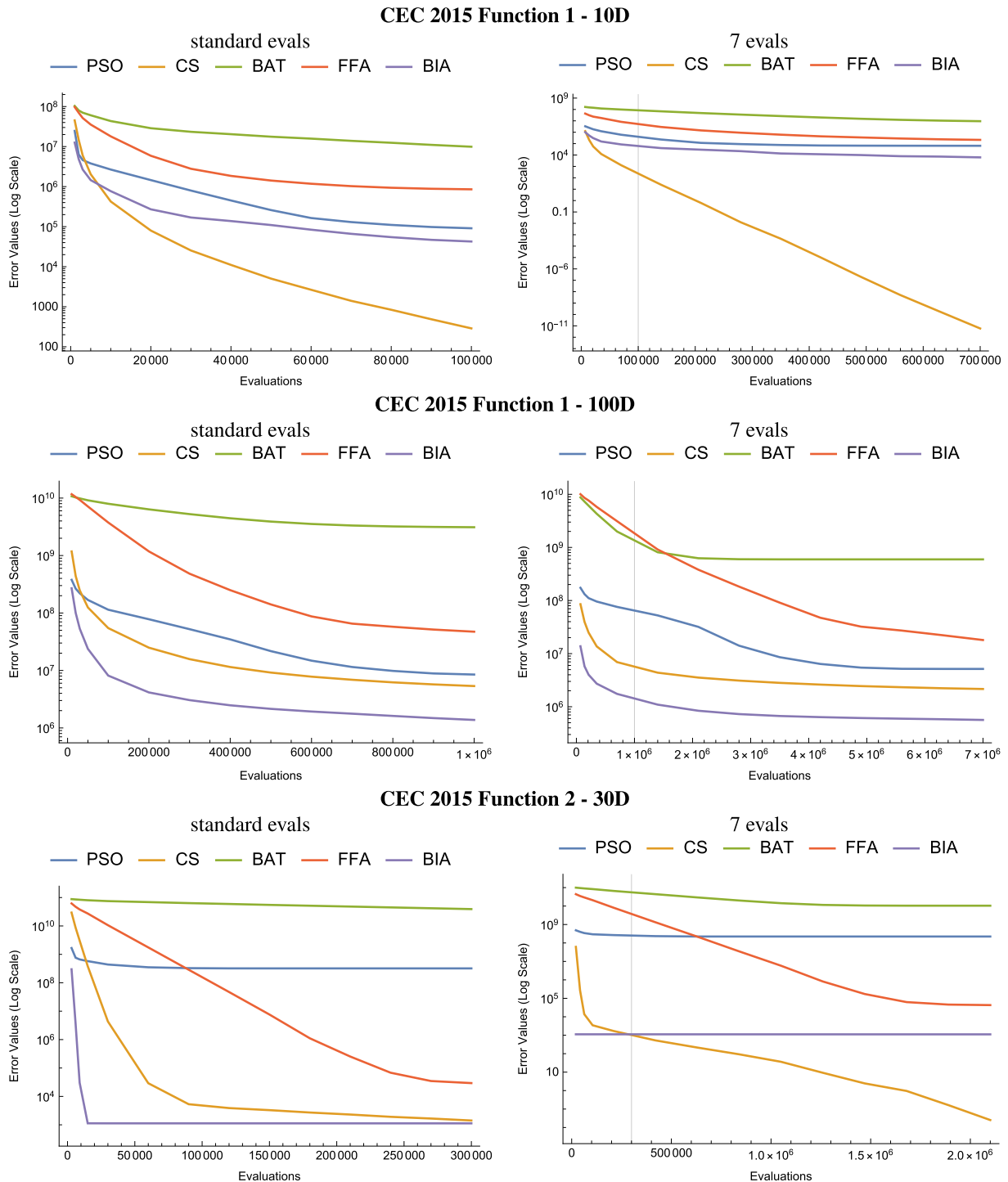


FIGURE 5. Selected cases of mean error values convergences showing surprising development of the convergence curve after a period of stagnation.

algorithms such as the Firefly Algorithm, the Bison Algorithm, or the Particle Swarm Optimization as well.

C. POPULATION DIVERSITY

Finally, we investigated the impact of a higher evaluation budget on the population diversity of the swarm algorithms. The population diversity measure is described in Eqs. 3, 4 [80], and the data are presented in a relative percentage to

a theoretical maximum of the diversity value.

$$Diversity = \frac{1}{NP} \sum_{i=1}^{NP} \sum_{j=1}^D (x_{i,j} - \bar{x}_j)^2 \tag{3}$$

$$\bar{x}_j = \frac{1}{NP} \sum_{i=1}^{NP} x_{i,j} \tag{4}$$

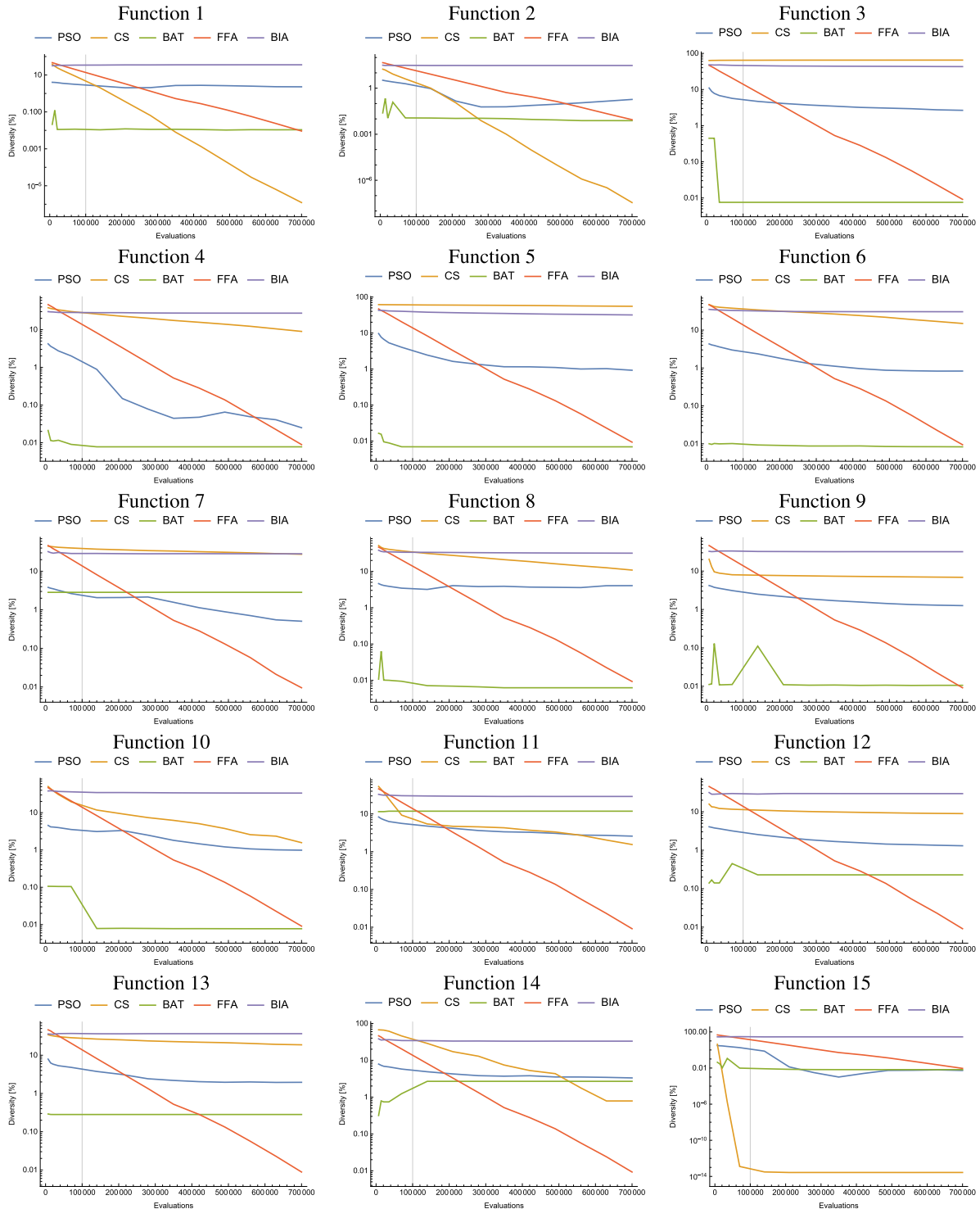


FIGURE 6. Convergence of the mean population diversity on the 15 problems of IEEE CEC 2015 in 10 D percentual to the maximum possible diversity, 7 evals scenario.

where:

- NP is the population size,
- D presents the dimensionality of the problem,
- i and j are the population and dimension iterators, respectively,
- $x_{i,j}$ represents the vector value of the solution at the given dimension,

- and \bar{x}_j presents the corresponding mean of the solutions.

Figure 6 presents the population diversity development in the 7 evals scenario on 15 problems of the IEEE CEC 2015 benchmark in 10 dimensions. The vertical line specifies the standard evaluation threshold. We investigated the population diversity statistics given the standard evals scenario

TABLE 11. Mean and median values of mean diversities percentual to the theoretical maximal diversity value computed from all functions in IEEE CEC 2015 benchmark with the standard evaluations budget.

	PSO		CS		BAT		FFA		BIA	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
10 D	2.03%	1.69%	24.96%	27.64%	0.32%	0.01%	0.01%	0.01%	33.72%	33.22%
30 D	1.66%	1.21%	25.83%	23.45%	0.37%	0.01%	0.01%	0.01%	34.83%	32.97%
50 D	1.56%	0.46%	24.75%	22.58%	1.87%	0.01%	0.01%	0.01%	35.23%	34.16%
100 D	1.92%	0.70%	22.45%	10.85%	2.42%	0.01%	0.01%	0.01%	36.10%	35.93%

TABLE 12. Mean and median values of mean diversities percentual to the theoretical maximal diversity value computed from all functions in IEEE CEC 2015 benchmark with the 7 evals budget.

	PSO		CS		BAT		FFA		BIA	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
10 D	1.53%	1.26%	14.76%	8.95%	1.20%	0.01%	0.01%	0.01%	32.06%	31.76%
30 D	1.57%	0.76%	20.34%	19.35%	1.06%	0.01%	0.01%	0.01%	32.55%	31.82%
50 D	2.27%	0.29%	21.27%	17.11%	1.07%	0.01%	0.01%	0.01%	33.86%	32.50%
100 D	2.17%	0.47%	21.46%	7.00%	0.77%	0.01%	0.01%	0.01%	34.34%	34.65%

TABLE 13. The impact of higher evaluation budget: Difference between mean diversities and median mean diversities given different numbers of evaluations on IEEE CEC 2015 benchmark [7 evals-std evals] (Positive values would mean beneficial effect of higher evaluation budget on population diversity).

	PSO		CS		BAT		FFA		BIA	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
10 D	-0.50%	-0.43%	-10.20%	-18.69%	0.88%	0.00%	0.00%	0.00%	-1.66%	-1.46%
30 D	-0.09%	-0.45%	-5.49%	-4.10%	0.69%	0.00%	0.00%	0.00%	-2.28%	-1.15%
50 D	0.71%	-0.17%	-3.48%	-5.47%	-0.80%	0.00%	0.00%	0.00%	-1.37%	-1.66%
100 D	0.25%	-0.23%	-0.99%	-3.85%	-1.65%	0.00%	0.00%	0.00%	-1.76%	-1.28%

TABLE 14. The impact of higher evaluation budget: Difference between mean diversities and median mean diversities given different numbers of evaluations on IEEE CEC 2017 benchmark [7 evals-std evals] (Positive values would mean beneficial effect of higher evaluation budget on population diversity).

	PSO		CS		BAT		FFA		BIA	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
10 D	-0.14%	0.00%	0.01%	0.00%	-0.10%	0.00%	0.00%	0.00%	1.06%	0.00%
30 D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
50 D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
100 D	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

(Tables 11, 18) and 7 evals scenario (Tables 12, 19). Since the population diversities are very similar for both of the tested benchmarks, the partial diversity results for IEEE CEC 2017 benchmark.

The final difference between the examined diversities (Table 12 - Table 11 and Table 19 - Table 18) on both tested benchmark testbeds is calculated in Table 13 and Table 14. Again, the positive value in these last two tables would mean a higher diversity of the 7 evals budget.

1) POPULATION DIVERSITY DISCUSSION

The higher evaluation budget effect on the overall population diversity is both algorithm and problem dependable. While the population diversity mostly lowers when solving the IEEE CEC 2015 benchmark (Table 13), the effect on IEEE CEC 2017 population diversity was negligible (Table 14).

Table 13 revealed that for most of the algorithms, the higher evaluation budget lowered the population diversity on the IEEE CEC 2015 benchmark testbed. The uncomplimentary effect concerned foremost the Cuckoo Search optimization,

especially in the lower dimensions, but even so, the diversity rate still stayed relatively high (see Table 12). The effect on other algorithms' diversities was either low (BIA, PSO, BAT) or none (FFA).

The extremely low population diversities of some algorithms (as can be seen at the end of some algorithms in Fig. 6) hint that the population merged merely into one location. This affects the exploration ability of the algorithms and points to a possible local optimum containment. On the other hand, the general progress of the population distribution (Figure 6, Tables 11 and 12) proved a steady diversity rate of the Bison Algorithm on the whole range of tested problems.

D. IMPACT OF RAISED FES BUDGET ON THE CEC COMPETITIONS WINNERS

So far, we have focused on the selected five swarm algorithms available from the TBU AILab's Github. However, we were also interested whether the raised budget would impact even advanced optimizers such as the winners of the examined CEC competitions. We used the public codes

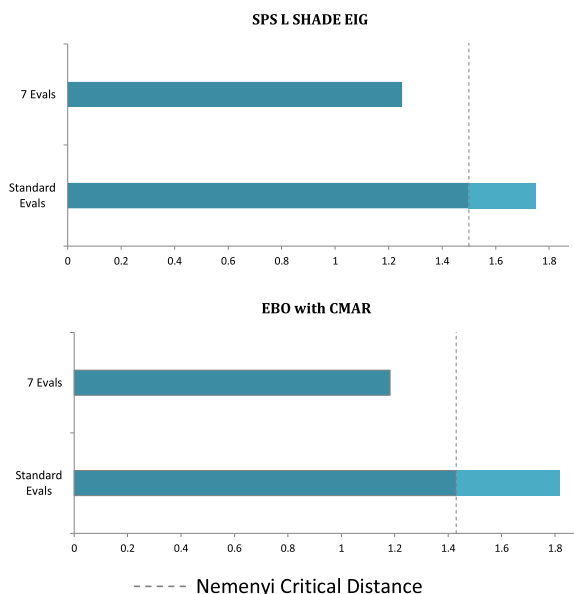


FIGURE 7. Pair-wise Friedman Rank test comparing the std evals and 7 evals budget scenarios.

of the competing algorithms available at the official CEC Benchmarking Github^{7,8} with raised FEs budget of the SPS L SHADE EIG on the CEC 2015 test set and EBO with CMAR on the CEC 2017 learning-based test set. We followed the benchmark directions during this experiment, with 51 optimization runs per problem and max FEs of $10,000 \times D$, and $70,000 \times D$ in the std evals and 7 evals scenarios, respectively. The parameters of the optimizers were already tuned in the original source codes provided in the official CEC Benchmarking Github with the included tuned parameters.

The following section discusses the results comparing the algorithm in the two evaluation scenarios. Table 15 presents the impact of the higher evaluation budget on the CEC 2015 winner and the CEC 2017 winner on their respective winning pools. The table shows the number of problems in which the raised budget produced significant improvement against the standard evaluation budget according to the Wilcoxon Rank-Sum test ($\alpha = 0.05$) and the corresponding percentual benefit (see Eqs. 1, 2). Similarly, Figure 7 presents the Friedman Rank test comparing the two evaluation scenarios on the examined algorithm on its respective winning test pool.

These results imply that the raised budget helps even successful versions of SHADE and EBO. It might appear that the higher budget provided a lower percentual benefit for the SPS L SHADE EIG algorithm when compared to other algorithms (see Table 9). However, it is important to note that the CEC 2015 winner sometimes found the exact optimum already within the standard evaluation budget. In these cases,

⁷<https://github.com/P-N-Suganthan/CEC2017-BoundConstrained>

⁸<https://github.com/P-N-Suganthan/CEC2015-Learning-Based>

TABLE 15. Impact of higher evaluation budget on one algorithm based on the evaluation budget scenarios (pair-wise Wilcoxon, $p < 0.05$, on IEEE CEC 2015 benchmark for the SPS L SHADE EIG and IEEE CEC 2017 for the EBO with CMAR algorithm) [7 evals– standard evals] (Positive values mean an overall beneficial effect of the higher FEs budget).

	SPS L SHADE EIG on CEC 2015	EBO with CMAR on CEC 2017
10 D	0/15	8/30
30 D	9/15	20/30
50 D	5/15	21/30
100 D	7/15	22/30
Percentual benefit	35%	59%

raising the budget does not contribute to the algorithm’s performance, as can be seen in the 10 dimensions in Table 15. On the other hand, in 30 dimensions, the higher evaluation budget significantly improved 9 out of 15 problems. These conclusions were confirmed by the pair-wise Friedman rank test in Figure 7, in which the 7 evals scenario outperformed the standard budget scenario significantly.

The improvements appeared even in the convergence analysis of the winners. Figure 8 shows the mean convergences of selected cases, in which the algorithms overcame an apparent stagnation when allowed for the raised FEs budget.

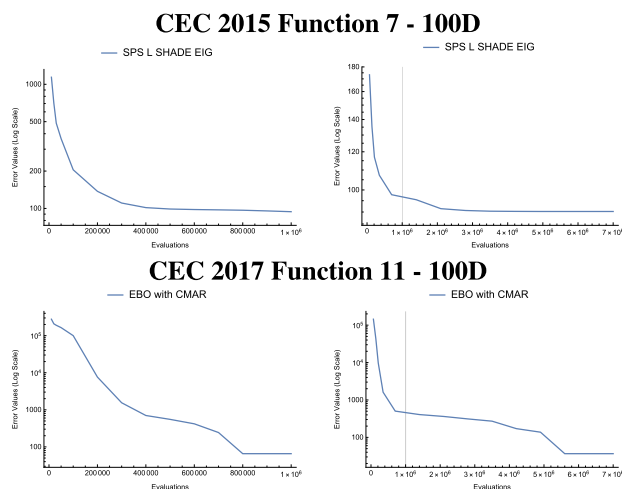


FIGURE 8. Mean convergence of IEEE CEC competition winners - special cases.

The previous section discussed the impact of a higher FEs budget on the comparison experiment’s final interpretation. However, when we added the benchmark winners to the optimizers, the statistical tests favored the winning algorithm regardless of the budget scenario. In several cases, the higher budget neutralized the lead; where the winner had significantly better results in the standard FEs budget, it lost the lead’s significance with a higher budget. Figures 9 and 10 show the Friedman Rank test comparing the algorithms on their respective winning test pools.

V. DISCUSSION

The heuristic optimization may resemble any other race: even this year’s Formula 1 champion is not a guaranteed winner

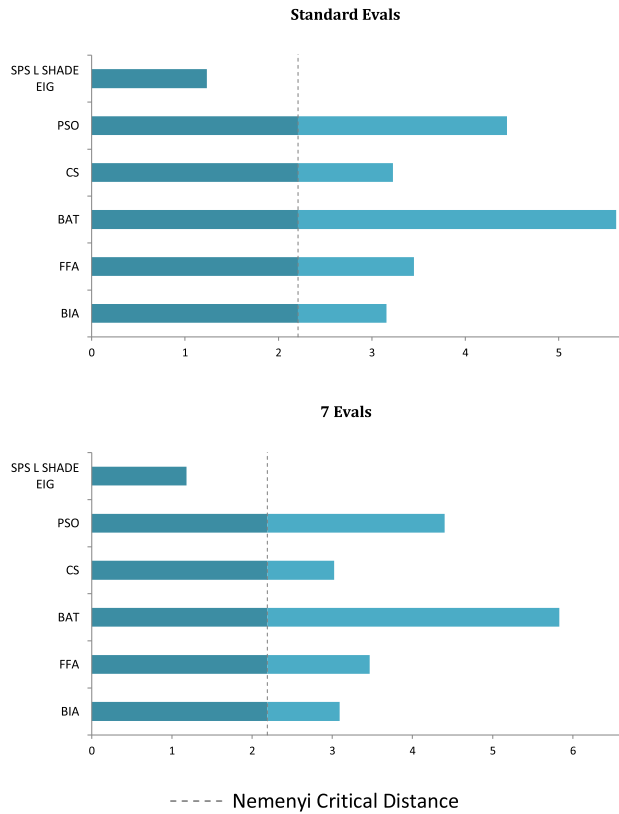


FIGURE 9. Friedman Rank test comparing all algorithms in two scenarios across all dimensions on CEC 2015.

TABLE 16. Wilcoxon Rank-Sum test ($p < 0.05$) comparing the pair standard evaluations vs 7 evals of each algorithm separately on CEC 2015.

10 dimensions	SPS L SHADE EIG	PSO	CS	BAT	FFA	BIA
None	5	7	1	2	1	5
Std evals	5	0	0	11	0	0
7 Evals	5	8	14	2	14	10
30 dimensions	SPS L SHADE EIG	PSO	CS	BAT	FFA	BIA
None	4	9	2	5	1	4
Std evals	1	0	0	6	0	0
7 Evals	10	6	13	4	14	11
50 dimensions	SPS L SHADE EIG	PSO	CS	BAT	FFA	BIA
None	4	8	3	3	1	4
Std evals	3	0	0	5	0	0
7 Evals	8	7	12	7	14	11
100 dimensions	SPS L SHADE EIG	PSO	CS	BAT	FFA	BIA
None	4	8	2	2	1	5
Std evals	2	0	0	4	0	0
7 Evals	9	7	13	9	14	10

on a different track, riding a backup vehicle, having one liter of fuel less, or racing other competitors. Similarly, in metaheuristics comparison, investigation of inner principles is preferable than to rank them purely by performance measures. In this article, we focused not only on the performance measures but also on the algorithms' convergence patterns and population diversities.

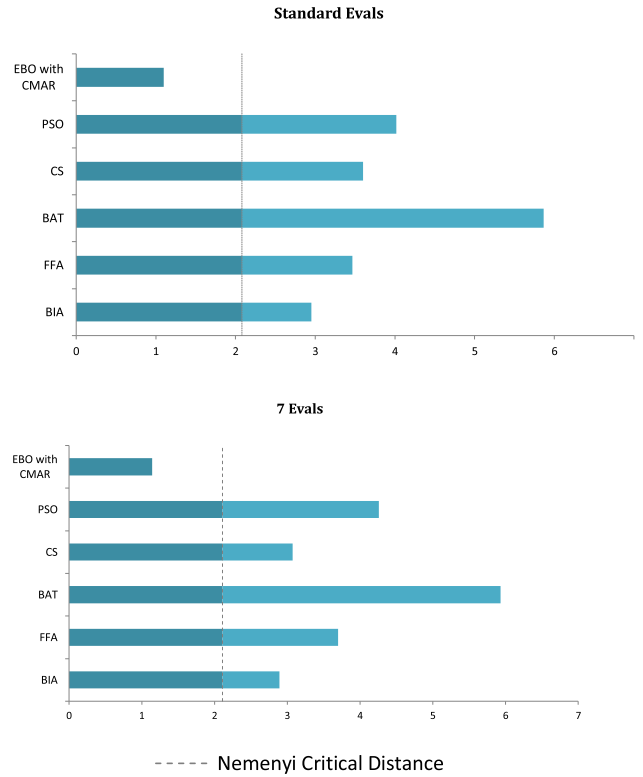


FIGURE 10. Friedman Rank test comparing all algorithms in two scenarios across all dimensions on CEC 2017.

TABLE 17. Wilcoxon Rank-Sum test ($p < 0.05$) comparing the pair standard evaluations vs 7 evals of each algorithm separately on CEC 2017.

10 dimensions	EBO with CMAR	PSO	CS	BAT	FFA	BIA
None	20	16	4	3	6	16
Std evals	1	0	0	21	1	1
7 Evals	9	14	26	6	23	13
30 dimensions	EBO with CMAR	PSO	CS	BAT	FFA	BIA
None	8	19	2	5	8	17
Std evals	1	1	0	15	1	0
7 Evals	21	10	28	10	21	13
50 dimensions	EBO with CMAR	PSO	CS	BAT	FFA	BIA
None	9	23	1	7	8	16
Std evals	0	0	0	12	2	0
7 Evals	21	7	29	11	20	14
100 dimensions	EBO with CMAR	PSO	CS	BAT	FFA	BIA
None	6	23	4	7	7	13
Std evals	1	0	0	11	2	0
7 Evals	23	7	26	12	21	17

We investigated the mean error convergences and found that some algorithms improved significantly even after a long stagnation. We discovered that the effect of the evaluation budget on population diversity is problem-dependent. While more evaluations diminished the population diversity on IEEE CEC 2015 benchmark's problems, its impact on IEEE CEC 2017 was negligible. Finally, we found that the

TABLE 18. Mean and median values of mean diversities percentual to the theoretical maximal diversity value computed from all functions in IEEE CEC 2017 benchmark with the standard evaluations budget.

	PSO		CS		BAT		FFA		BIA	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
10 D	1.52%	1.15%	21.98%	19.57%	3.29%	0.01%	0.01%	0.01%	32.44%	32.87%
30 D	0.75%	0.41%	26.11%	30.20%	0.02%	0.01%	0.01%	0.01%	33.14%	32.62%
50 D	0.96%	0.47%	25.97%	27.83%	0.01%	0.01%	0.01%	0.01%	33.97%	33.26%
100 D	1.31%	0.50%	24.65%	25.36%	0.01%	0.01%	0.01%	0.01%	34.43%	34.29%

TABLE 19. Mean and median values of mean diversities percentual to the theoretical maximal diversity value computed from all functions in IEEE CEC 2017 benchmark with the 7 evals budget.

	PSO		CS		BAT		FFA		BIA	
	Median	Mean	Mean	Median	Mean	Median	Mean	Median	Mean	Median
10 D	1.38%	1.15%	21.99%	19.57%	3.19%	0.01%	0.01%	0.01%	33.50%	32.87%
30 D	0.75%	0.41%	26.11%	30.20%	0.02%	0.01%	0.01%	0.01%	33.14%	32.62%
50 D	0.96%	0.47%	25.97%	27.83%	0.01%	0.01%	0.01%	0.01%	33.97%	33.26%
100 D	1.31%	0.50%	24.65%	25.36%	0.01%	0.01%	0.01%	0.01%	34.43%	34.29%

evaluation budget significantly affected six of the tested algorithms and considerably improved their performance.

The most critical effect concerned the interpretation of the statistical results when comparing the swarm algorithms separately. While in the standard evaluation scenarios (the last rows of Tables 3, 4), the sum of Wilcoxon Rank-Sum test wins was in favor of the Bison Algorithm, more evaluations (Tables 5, 6) promoted the Cuckoo Search optimization, and therefore flipped the final winner of this particular race in the swarm optimizers comparison. The effect was damped, when we added the competition winners to the list of optimizers, as the Friedman Rank test in Figures 9 and 10 ranked the SPS L SHADE EIG and EBO with CMAR the first in both of the tested scenarios.

We also examined the impact of a higher evaluation budget on successful optimizers that won the IEEE CEC competitions. The raised budget allowed for a significantly better performance of these algorithms.

Interestingly, a higher evaluation budget's overall percentual benefit was considerably lower for the CEC 2015 winner (compare Tables 9 and 15). The reason may be that the SPS L SHADE EIG sometimes found the exact optimum already within the standard evaluation budget – hence the 0% benefit for 10-dimensional problems. However, in higher dimensions, the percentual benefit was renewed (up to 60% in 30 dimensions). Therefore, we can recommend raising the evaluation budget even for effective optimizers when solving problems without the known optimum.

In the end, we would like to highlight that many applications, like the computationally expensive problems, do not allow for numerous evaluations. Furthermore, this article did not mention the algorithms' complexity, which relatively does not change with FEs, but can greatly affect the options to set the evaluations budget due to the limited resources or time restrictions. The time consumed by the presented experiments took minutes to days based on the solving algorithm.

VI. CONCLUSION

The function evaluation limit impacts both the performance and behavior of metaheuristics. Therefore, when one solves

a nontrivial problem without a pressing deadline, it may be reasonable to use a higher evaluation budget to obtain better results.

It might seem that our paper states the obvious – that a longer optimization process sometimes produces better results. However, there is more to the story. Our primary goal was to point out the importance of focusing on good practice in benchmarking and transferring this knowledge into engineering optimization practice and metaheuristic development. The most important findings can be summarised as follows:

We discovered that some metaheuristics conceal a hidden asset that only a raised number of objective function evaluations may reveal: the ability to renew the convergence after apparent stagnation. This feature may be crucial for specific real-world applications. Also, not every algorithm has this ability – some algorithms are more affected by the evaluation budget than others. In this context, the Cuckoo Search optimization with more evaluations carried out impressive results, as it repeatedly recovered convergence even after periods of stagnation.

The impact of the function evaluation budget on population diversity was no less interesting. While one set of problems was intact by evaluation change, there was a significant drop in the population diversity on the other one. However, lower population diversity in higher iterations raises the chance of capturing the entire population to a local optimum.

The world of metaheuristic optimization is currently ruled by the No Free Lunch theorem, which states that there is no universal best-to-solve-it-all algorithm. According to this theory, various algorithms are more (or less) suitable to solve different kinds of problems. At the same time, the success of a metaheuristic stands on a whole list of conditions: including dimensionality, solving algorithm selection, parameter configuration, adopted border strategies, problem dynamics, and the objective function. To the account of conditions affecting optimizations, we cordially recommend adding another input variable: the function evaluation limit.

So far, the benchmarking testbeds focused mostly on solving problems fast, which corresponds with most of the

objective function evaluation budget limitations. However, as our paper revealed, some algorithms can overcome the potential stagnation with an increased FEs budget. Hence, it might be useful if the future direction of benchmark testing would consider increasing the FEs budget besides the currently adopted limits. The aim of uncovering this feature and the investigation of why some algorithms can renew the convergence even after apparent stagnation might form the next generation of future benchmark profiling.

In this article, we have discovered that some algorithms are able to renew the convergence within the extended FEs budget. It is possible that with an even larger FEs budget, other algorithms would be able to renew the convergence as well, despite failing to do so within our experiments. However, for practical purposes, the number of function evaluations must be limited at a finite value. Therefore, we believe that instead of still increasing the FEs limit, an in-deep study of the algorithms is necessary to uncover the key features for convergence renewal ability, and our work should encourage the meaningfulness of such a study.

REFERENCES

- [1] M. Pluhacek, R. Senkerik, A. Viktorin, T. Kadavy, and I. Zelinka, "A review of real-world applications of particle swarm optimization algorithm," in *AETA 2017—Recent Advances in Electrical Engineering and Related Sciences: Theory and Application* (Lecture Notes in Electrical Engineering), V. H. Duy, T. T. Dao, I. Zelinka, S. B. Kim, and T. T. Phuong, Eds. Cham, Switzerland: Springer, 2018, pp. 115–122.
- [2] D. Oliva, S. Hinojosa, and M. V. Demeshko, "Engineering applications of metaheuristics: An introduction," *J. Phys., Conf. Ser.*, vol. 803, Jan. 2017, Art. no. 012111.
- [3] I. H. Osman and J. P. Kelly, "Meta-heuristics theory and applications," *J. Oper. Res. Soc.*, vol. 48, no. 6, p. 657, 1997.
- [4] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimization problems," 2013, *arXiv:1308.4008*. [Online]. Available: <http://arxiv.org/abs/1308.4008>
- [5] E. H. Houssein, M. R. Saad, F. A. Hashim, H. Shaban, and M. Hassaballah, "Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103731.
- [6] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541.
- [7] S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 20–34, Apr. 2019.
- [8] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
- [9] P. Reche-López, N. Ruiz-Reyes, S. G. Galán, and F. Jurado, "Comparison of metaheuristic techniques to determine optimal placement of biomass power plants," *Energy Convers. Manage.*, vol. 50, no. 8, pp. 2020–2028, Aug. 2009.
- [10] Ī. B. Aydılek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft Comput.*, vol. 66, pp. 232–249, May 2018.
- [11] A. Mortazavi, V. Togan, A. Daloglu, and A. Nuhoglu, "Comparison of two metaheuristic algorithms on sizing and topology optimization of trusses and mathematical functions," *Gazi Univ. J. Sci.*, vol. 31, no. 2, pp. 416–435, 2018.
- [12] A. Bolufé-Röhler and S. Chen, "A multi-population exploration-only exploitation-only hybrid on CEC-2020 single objective bound constrained problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.
- [13] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, "Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm," *Swarm Evol. Comput.*, vol. 26, pp. 8–22, Feb. 2016.
- [14] J. Pierzezan and L. Dos Santos Coelho, "Coyote optimization algorithm: A new metaheuristic for global optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.
- [15] G.-G. Wang, "Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Comput.*, vol. 10, no. 2, pp. 151–164, Jun. 2018.
- [16] F. J. Rodriguez, C. Garcia-Martinez, and M. Lozano, "Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: Taxonomy, comparison, and synergy test," *IEEE Trans. Evol. Comput.*, vol. 16, no. 6, pp. 787–800, Dec. 2012.
- [17] T. Wortmann, "Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization," *J. Comput. Des. Eng.*, vol. 6, no. 3, pp. 414–428, Jul. 2019.
- [18] T. Bartz-Beielstein, C. Doerr, D. van den Berg, J. Bossek, S. Chandrasekaran, T. Eftimov, A. Fischbach, P. Kerschke, W. L. Cava, M. Lopez-Ibanez, K. M. Malan, J. H. Moore, B. Naujoks, P. Orzechowski, V. Volz, M. Wagner, and T. Weise, "Benchmarking in optimization: Best practice and open issues," 2020, *arXiv:2007.03488*. [Online]. Available: <https://arxiv.org/abs/2007.03488>
- [19] A. LaTorre, D. Molina, E. Osaba, J. D. Ser, and F. Herrera, "Fairness in bio-inspired optimization research: A prescription of methodological guidelines for comparing meta-heuristics," Apr. 2020, *arXiv:2004.09969*. [Online]. Available: <http://arxiv.org/abs/2004.09969>
- [20] D. Molina, J. Poyatos, J. D. Ser, S. García, A. Hussain, and F. Herrera, "Comprehensive taxonomies of nature- and bio-inspired optimization: Inspiration versus algorithmic behavior, critical analysis and recommendations," 2020, *arXiv:2002.08136*. [Online]. Available: <https://arxiv.org/abs/2002.08136>
- [21] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Mach. Intell.*, vol. 1, no. 1, pp. 24–35, Jan. 2019.
- [22] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 293–312.
- [23] T. Ting, X.-S. Yang, S. Cheng, and K. Huang, "Hybrid metaheuristic algorithms: Past, present, and future," in *Recent Advances in Swarm Intelligence and Evolutionary Computation*. Cham, Switzerland: Springer, 2015, pp. 71–83.
- [24] J. Silberholz and B. Golden, "Comparison of metaheuristics," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), M. Gendreau and J.-Y. Potvin, Eds. Boston, MA, USA: Springer, 2010, pp. 625–640.
- [25] U. Balande and D. Shrimankar, "SRIFA: Stochastic ranking with improved-firefly-algorithm for constrained optimization engineering design problems," *Mathematics*, vol. 7, no. 3, p. 250, Mar. 2019.
- [26] A. P. Engelbrecht, "Fitness function evaluations: A fair stopping condition?" in *Proc. IEEE Symp. Swarm Intell.*, Dec. 2014, pp. 1–8.
- [27] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2006.
- [28] J. Knowles and E. J. Hughes, "Multiobjective optimization on a budget of 250 evaluations," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), C. A. C. Coello, A. H. Aguirre, and E. Zitzler, Eds. Berlin, Germany: Springer, 2005, pp. 176–190.
- [29] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, and B. Qu, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 2016, p. 34.
- [30] K. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Problem definitions and evaluation criteria for the 100-digital challenge special session and competition on single objective numerical optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 2018.

- [31] A. Kazikova, M. Pluhacek, and R. Senkerik, "Performance of the bison algorithm on benchmark IEEE CEC 2017," in *Artificial Intelligence and Algorithms in Intelligent Systems* (Advances in Intelligent Systems and Computing), R. Silhavy, Ed. Cham, Switzerland: Springer, 2019, pp. 445–454.
- [32] B. Liu, J. J. Liang, P. N. Suganthan, and B. Qu, "Problem definitions and evaluation criteria for computationally expensive single objective numerical optimization," *Comput. Intell. Lab., Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2013.
- [33] A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, "A test-suite of non-convex constrained optimization problems from the real-world and some baseline results," *Swarm Evol. Comput.*, vol. 56, Aug. 2020, Art. no. 100693.
- [34] C. A. C. Coello and G. T. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), E. Zitzler, L. Thiele, K. Deb, C. A. C. Coello, and D. Corne, Eds. Berlin, Germany: Springer, 2001, pp. 126–140.
- [35] L. V. Santana-Quintero, N. Ramírez-Santiago, C. A. C. Coello, J. M. Luque, and A. G. Hernández-Díaz, "A new proposal for multiobjective optimization using particle swarm optimization and rough sets theory," in *Parallel Problem Solving From Nature* (Lecture Notes in Computer Science), T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds. Berlin, Germany: Springer, 2006, pp. 483–492.
- [36] H. Eskandari, C. D. Geiger, and G. B. Lamont, "FastPGA: A dynamic population sizing approach for solving expensive multiobjective optimization problems," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Germany: Springer, 2007, pp. 141–155.
- [37] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Nanyang Technol. Univ., Singapore, KanGAL Rep.*, 2005.
- [38] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *J. Appl. Mech.*, vol. 41, no. 8, pp. 8–31, 2006.
- [39] V. L. Huang, A. K. Qin, K. Deb, E. Zitzler, P. N. Suganthan, J. J. Liang, M. Preuss, and S. Huband, "Problem definitions for performance assessment of multi-objective optimization algorithms," *Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2007.
- [40] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, and Z. Yang, "Benchmark functions for the CEC'2008 special session and competition on large scale global optimization," *Nature Inspired Comput. Appl. Lab., USTC, Hefei, China, Tech. Rep.*, 2007, pp. 1–18, vol. 24.
- [41] C. Li, S. Yang, T. T. Nguyen, E. L. Yu, X. Yao, Y. Jin, H. G. Beyer, and P. N. Suganthan, "Benchmark generator for CEC 2009 competition on dynamic optimization," *Dept. Comput. Sci., Inst. Artif. Intell., Univ. Leicester, Leicester, U.K., Tech. Rep.*, 2008. [Online]. Available: <https://www.dora.dmu.ac.uk/handle/2086/11893>
- [42] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *Univ. Essex, Nanyang Technol. Univ., Colchester, U.K., Tech. Rep.*, 2008.
- [43] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC 2010 special session and competition on large-scale global optimization (2010)," *Nature Inspired Comput. Appl. Lab., Nanyang Technol. Univ., USTC, Hefei, China, Tech. Rep.*, 2009.
- [44] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," *Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2010, vol. 24.
- [45] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," *Jadavpur Univ., Nanyang Technol. Univ., Kolkata, India, Tech. Rep.*, 2010, pp. 341–359.
- [46] J. J. Liang, B. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2013, pp. 281–295, vol. 201212, no. 34.
- [47] J. Liang, B. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2014.
- [48] Q. Chen, B. Liu, Q. Zhang, J. Liang, P. N. Suganthan, and B. Qu, "Problem definitions and evaluation criteria for CEC 2015 special session on bound constrained single-objective computationally expensive numerical optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2014.
- [49] J. J. Liang, B. Qu, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 201411A, 2014, pp. 625–640, vol. 29.
- [50] B. Qu, J. J. Liang, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on single objective multi-niche optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep.*, 2014.
- [51] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," *Nat. Univ. Defense Technol., Changsha, China, Kyungpook Nat. Univ., Daegu, South Korea, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2017.
- [52] J. J. Liang, P. N. Suganthan, B. Qu, D. Gong, and Y. Caitong, "Problem definitions and evaluation criteria for the CEC 2020 special session on multimodal multiobjective optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2019.
- [53] C. Yue, K. Price, P. N. Suganthan, J. Liang, M. Ali, B. Qu, N. Awad, and P. Biswas, "Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep.* 201911, 2019.
- [54] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez, "Metaheuristic optimization frameworks: A survey and benchmarking," *Soft Comput.*, vol. 16, no. 3, pp. 527–561, Mar. 2012.
- [55] M. Hellwig and H.-G. Beyer, "Benchmarking evolutionary algorithms for single objective real-valued constrained optimization—A critical review," *Swarm Evol. Comput.*, vol. 44, pp. 927–944, Feb. 2019.
- [56] P. Savsani and V. Savsani, "Passing vehicle search (PVS): A novel metaheuristic algorithm," *Appl. Math. Model.*, vol. 40, nos. 5–6, pp. 3951–3978, Mar. 2016.
- [57] A. E. Ezugwu, O. J. Adeleke, A. A. Akinyelu, and S. Viriri, "A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 6207–6251, May 2020.
- [58] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov. 1995, pp. 1942–1948.
- [59] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.
- [60] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Apr. 2010, arXiv:1004.4170*. [Online]. Available: <http://arxiv.org/abs/1004.4170>
- [61] X.-S. Yang, "Firefly algorithm, Lévy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*, M. Bramer, R. Ellis, and M. Petridis, Eds. London, U.K.: Springer, 2010, pp. 209–218.
- [62] A. Kazikova, M. Pluhacek, T. Kadavy, and R. Senkerik, "Introducing the run support strategy for the bison algorithm," in *AETA 2018—Recent Advances in Electrical Engineering and Related Sciences: Theory and Application* (Lecture Notes in Electrical Engineering), I. Zelinka, P. Brandstetter, T. T. Dao, V. H. Duy, and S. B. Kim, Eds. Cham, Switzerland: Springer, 2020, pp. 272–282.
- [63] S.-M. Guo, J. S.-H. Tsai, C.-C. Yang, and P.-H. Hsu, "A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 1003–1010.
- [64] A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1835–1842.
- [65] A. P. Engelbrecht, "Particle swarm optimization: Global best or local best?" in *Proc. BRICS Congr. Comput. Intell., 11th Brazilian Congr. Comput. Intell.*, Sep. 2013, pp. 124–135.
- [66] I. Fister, I. Fister, X.-S. Yang, S. Fong, and Y. Zhuang, "Bat algorithm: Recent advances," in *Proc. IEEE 15th Int. Symp. Comput. Intell. Informat. (CINTI)*, Nov. 2014, pp. 163–167.

- [67] A. Kazikova, M. Pluhacek, R. Senkerik, and A. Viktorin, "Proposal of a new swarm optimization method inspired in bison behavior," in *Recent Advances in Soft Computing (Advances in Intelligent Systems and Computing)*, R. Matoušek, Ed. Cham, Switzerland: Springer, 2019, pp. 146–156.
- [68] F. Vandenbergh and A. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Inf. Sci.*, vol. 176, no. 8, pp. 937–971, Apr. 2006.
- [69] Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. Congr. Evol. Comput.*, vol. 1, May 2001, pp. 81–86.
- [70] H. Faris, I. Aljarah, S. Mirjalili, P. A. Castillo, and J. J. Merelo, "EvolvoPy: An open-source nature-inspired optimization framework in Python," in *Proc. 8th Int. Joint Conf. Comput. Intell.*, 2016, pp. 171–177.
- [71] K. R. Harrison, B. M. Ombuki-Berman, and A. P. Engelbrecht, "Optimal parameter regions for particle swarm optimization algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 349–356.
- [72] P. Maca and P. Pech, "The inertia weight updating strategies in particle swarm optimisation based on the beta distribution," *Math. Problems Eng.*, vol. 2015, Mar. 2015, Art. no. 790465.
- [73] Y.-B. Mo, Y.-Z. Ma, and Q.-Y. Zheng, "Optimal choice of parameters for firefly algorithm," in *Proc. 4th Int. Conf. Digit. Manuf. Automat.*, Jun. 2013, pp. 887–892.
- [74] F. Xue, Y. Cai, Y. Cao, Z. Cui, and F. Li, "Optimal parameter settings for bat algorithm," *Int. J. Bio-Inspired Comput.*, vol. 7, no. 2, pp. 125–128, May 2015.
- [75] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2008.
- [76] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," Dec. 2010, *arXiv:1005.2908*. [Online]. Available: <http://arxiv.org/abs/1005.2908>
- [77] A. Kazikova, M. Pluhacek, and R. Senkerik, "Why tuning the control parameters of metaheuristic algorithms is so important for fair comparison?" *Mendel*, vol. 26, no. 2, pp. 9–16, Dec. 2020.
- [78] I. Fister, Jr., X.-S. Yang, I. Fister, and J. Brest, "Memetic firefly algorithm for combinatorial optimization," May 2012, *arXiv:1204.5165*. [Online]. Available: <http://arxiv.org/abs/1204.5165>
- [79] A. Kazikova, M. Pluhacek, and R. Senkerik, "Tuning of the bison algorithm control parameters," in *Proc. Eur. Council Modelling Simulation*, 2018, pp. 156–162.
- [80] R. Polakova, J. Tvrđik, P. Bujok, and R. Matousek, "Population-size adaptation through diversity-control mechanism for differential evolution," in *Proc. 22th Int. Conf. Soft Comput.*, 2016, pp. 49–56.



Award in 2013 and the Dean's Award in 2015 for her bachelor's and master's theses, respectively.

ANEZKA KAZIKOVA received the degree in engineering informatics (Hons.) from Tomas Bata University in Zlin, Czech Republic, in 2015, where she is currently pursuing the Ph.D. degree with the Artificial Intelligence Department. She studies the secrets and challenges of novel metaheuristic development. In 2017, she developed the Bison Algorithm optimization with non-negligible help from the TBU's AI Laboratory and they keep on running ever since. She received the EdHouse



MICHAL PLUHACEK (Member, IEEE) received the Ph.D. degree in information technologies from Tomas Bata University in Zlin, Czech Republic, in 2016, with the dissertation topic: Modern method of development and modifications of evolutionary computational techniques. He currently works as a Junior Researcher with the Regional Research Centre CEBIA-Tech, Tomas Bata University in Zlin. In 2019, he finished six months long research stay at the New Jersey Institute of Technology, USA, focusing on swarm robotics. Since 2020, he has been the Head of the AI Laboratory, Swarm Robotics Research Group, Tomas Bata University in Zlin. His research interests include evolutionary computing and swarm intelligence (theory and applications) and artificial intelligence in general.



ROMAN SENKERIK (Member, IEEE) is currently an Associate Professor and, since 2017, the Head of the AI Laboratory, Department of Informatics and Artificial Intelligence, Tomas Bata University in Zlin. He is the author of more than 40 journal articles, 250 conference papers, several book chapters, and editorial notes. His research interests include the development of evolutionary algorithms, modifications and benchmarking of EA, soft computing methods, and interdisciplinary applications in optimization and cyber-security, machine learning, neuro-evolution, data science, the theory of chaos, and complex systems. He is a recognized Reviewer of many leading journals in computer science/computational intelligence. He was a part of the organizing teams for special sessions/symposiums at IEEE WCCI, CEC, or SSCI events. He was a Guest Editor of several special issues in journals and an Editor of proceedings for several conferences.

...