A Comparison of the State-of-the-Art Evolutionary Algorithms with Different Stopping Conditions

Jana Herzog[a], Janez Brest[b] and Borko Bošković[c]
Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, Maribor, Slovenia

Keywords: Evolutionary Algorithms, Fixed-Budget Approach, Stopping Conditions.

Abstract: This paper focuses on the comparison of the state-of-the-art algorithms and the influence of a stopping condition, the maximum number of function evaluations, on the optimization process. The main aim is to compare the chosen state-of-the-art algorithms with different predetermined stopping conditions and observe if they are comparable in reaching a certain quality of solution on a given set of benchmark functions. For this analysis, the four most recent state-of-the-art evolutionary algorithms were chosen for comparison on the latest set of benchmark functions. We utilized a fixed-budget approach with different values of stopping conditions. Small differences in the algorithms' performances are observed and the obtained results are also statistically analyzed. Different values of the stopping conditions show different rankings of evolutionary algorithms without the significant difference. The possible reason for this is that their performances are very close.

1 INTRODUCTION

It is extremely demanding to improve an evolutionary algorithm's performance for a specific set of benchmark functions. Various mechanisms can be helpful in such tasks, for example mechanisms such as population size reduction (Tanabe and Fukunaga, 2014), (Piotrowski et al., 2020), perturbation techniques (Van Cuong et al., 2022), using multiple mutation strategies (Zhu et al., 2023), multiple crossover techniques (Bujok and Kolenovsky, 2022), self-adaptive parameters (Brest et al., 2014) and re-initialization mechanism (Meng et al., 2021). The aim of these mechanisms is the same; to enhance and refine the algorithm's performance to reach a better quality of solutions, to increase the speed of convergence, or to lower the runtime while considering a fixed-budget (Jansen, 2020) or a fixed-target approach (Hansen et al., 2021).

With the fixed target approach, we set a specific value as a target value (quality of solution) and observe if it will be reached by a given algorithm. Here we also observe the number of function evaluations or runtime spent to reach the optimal or suboptimal target value. While, with the fixed-budget approach, we set a specific budget (number of function evaluations or runtime) and observe the quality of the solution that will be reached.

However, both approaches have some shortcomings, for example with the fixed-budget approach, the algorithm is restricted to a certain number of function evaluations. This can hinder the algorithm's ability to explore the search space fully and it can result in solutions of suboptimal quality. Whereas for the fixed-target approach, a choice of targets may present a challenge, since not every solver is able to reach the wanted quality of solutions.

The main intention of every experiment is to reach the optimal solution with the smallest possible number of function evaluations as possible, while also observing the best quality of solutions. The maximum number of function evaluations or the budget (in the fixed-budget approach) is a common stopping condition used in competitions, such as IEEE Congress on Evolutionary Computation (CEC) Special Sessions and Competitions on Real-Parameter Single-Objective Optimization (A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan, 2020), (A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan, 2021). In the CEC competition, almost every year, a new set of benchmark functions

[a] https://orcid.org/0000-0001-5555-878X
[b] https://orcid.org/0000-0001-5864-3533
[c] https://orcid.org/0000-0002-7595-2845

Herzog, J., Brest, J. and Bošković, B.
A Comparison of the State-of-the-Art Evolutionary Algorithms with Different Stopping Conditions.
DOI: 10.5220/0012182200003595
In *Proceedings of the 15th International Joint Conference on Computational Intelligence (IJCCI 2023)*, pages 222-229
ISBN: 978-989-758-674-3; ISSN: 2184-3236

is utilized to compare the newest proposed state-of-the-art evolutionary algorithms under the same set of conditions (A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan, 2021). The problem occurs when some algorithms are budget dependent (Tušar et al., 2017), meaning that some of their parameters are set based on the stopping condition *maxFEs* (maximum number of function evaluations). Some of the evolutionary algorithms use *maxFEs* as a control parameter in the mechanism of the population size reduction. There, it affects the population size and consequently influences the optimization process. Examples of such state-of-the-art algorithms are L-SHADE (Tanabe and Fukunaga, 2014), iL-SHADE (Brest et al., 2016) and MadDE (Biswas et al., 2021).

It is even more difficult to predict algorithm's performance while using a different budget from the one being used in the experiment of the given/cited paper or competition (Hansen et al., 2021). Furthermore, a programming language can play a significant role in determining the prevailing algorithm on a given benchmark. This holds true in cases, when the observed variable is runtime or speed (Herzog et al., 2022), (Ravber et al., 2022).

The question which arises is following: how comparable are the state-of-the-art algorithms based on different predetermined stopping conditions? Are some of the state-of-the-art evolutionary algorithms adapted for a certain stopping condition?

In this paper, it is shown how different predetermined stopping conditions affect the chosen evolutionary algorithms in reaching a certain quality of solutions. For the purpose of this analysis, four state-of-the-art algorithms were chosen: EA4eig (Bujok and Kolenovsky, 2022), NL-SHADE-RSP (Biedrzycki et al., 2022), NL-SHADE-LBC (Stanovov et al., 2022), and S-LSHADE-DP (Van Cuong et al., 2022). All four evolutionary algorithms were compared on the CEC 2022 Benchmark functions. We present the statistical analysis and describe how the ranking of the chosen algorithms changes when a different maximum number of function evaluations is being applied. We ran the algorithms with the original settings from the competition (A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan, 2021) and then we modified the maximum number of function evaluations to see whether there are any differences between the algorithms' ranks or their performances.

The paper is organized as follows. In Section 2, the related work is described. In Section 3, the experiment and analysis are provided. Section 4 finally concludes our paper.

2 RELATED WORK

It is extremely difficult to deduce, which algorithm is the best for a chosen optimization problem. In this section, we will present a few of the newer approaches which tackle the stochastic algorithm's analysis and we will emphasize what they focus on.

The questions which occur in terms of the performance analysis are for example the following, how fast can an algorithm reach a wanted solution quality and with what budget this can be achieved (Bartz-Beielstein et al., 2020). The answers are provided by the fixed-target or fixed-budget approaches. One of the most common measurements of evaluating an algorithm is the number of function evaluations (*NFEs*). For example, the aim of some methods is to achieve the smallest number of function evaluations in reaching the optimal solution. However, some of the analyses focus more on the speed or runtime of the algorithm (Herzog et al., 2022). The measurement of time can be sensitive to some factors, such as the programming language, hardware, or even of the workload of the CPUs. This aspect makes algorithms more difficult to compare (Bartz-Beielstein et al., 2020).

However, all research is aimed towards establishing a fair comparison of evolutionary algorithms and choosing the best one for a given optimization problem. Furthermore, the analyses should be also aimed towards choosing the best algorithm for real-world problems (Bartz-Beielstein et al., 2020).

The anytime approach (Hansen, Nikolaus and Auger, Anne and Brockhoff, Dimo and Tušar, Tea, 2022), argues that the appropriate measurement to provide a quantitative and meaningful performance assessment is the number of blackbox evaluations to reach a predefined target value. They call this measure the runtime of the algorithm. It is also budget-free since the authors support the claim that benchmarking for a single budget seems inefficient. Furthermore, it does not provide enough information about the solver or the optimization problem. The approach is able to assess the performance anytime. The main focus is on using quantitative performance measures on a ratio scale and runtime measurements.

It is no longer enough to choose only one algorithm for a given set of benchmark functions (Wolpert and Macready, 1997), but to choose the best algorithm for each problem instance. This problem can be mitigated by using the automated algorithm selection (Cenikj et al., 2022). This approach emphasizes that different instances are best solved when different algorithms are being used (Kerschke et al., 2019). Researchers have been focusing mostly on the characteristics of an algorithm, however the focus should also be on the characteristics of an optimization problem. In (Herzog et al., 2023), it is shown how to predict a stopping condition for a specific optimization problem and stochastic solver with a certain probability. The prediction model is based on the statistical distribution of two variables (runtime and number of function evaluations). Based on the smaller dimensions of an optimization problem and a chosen stochastic solver, one is able to predict in what runtime or number of function evaluations a solution of wanted quality will be reached with any probability for larger dimensions.

An interest has been taken recently in exploratory landscape analysis, which characterizes optimization problem instances with numerical features, which describe different sides of the problem instances (Nikolikj et al., 2022). The approaches changed also in terms of not only choosing an algorithm suitable for a benchmark but vice-versa. The SELECTOR (Cenikj et al., 2022) approach focuses on selecting a representative set of benchmark functions to provide a reproducible and replicable statistical comparison.

No matter which method is used, a solid statistical analysis should be its basis (Birattari and Dorigo, 2007). The most common approach to determine whether there is a statistical significance between two solvers is using parametric or non-parametric statistical tests. However, for the parametric tests some assumptions should be checked before applying them to the data. These assumptions are: normality of the data, homogeneity of variances and independence of the observations (Carrasco et al., 2020). Violations of these assumptions may lead to an incorrect conclusion, which can result in misinterpretation due to the lack of statistical knowledge.

Due to several pitfalls and mishaps, which can occur with the incorrect use of the statistical tests, the researchers have gained interest in Bayesian techniques (Benavoli et al., 2017). Bayesian methods provide interpretable information based on the underlying performance distribution (Calvo et al., 2019).

Researchers use established and novel methods with the intention of determining which algorithm prevails on a given set of benchmark functions. By comparing different approaches using the benchmark functions, researchers aim to identify the strengths and weaknesses of each algorithm and gain comprehensive insights into their performances.

3 EXPERIMENT

In this section, we provide the experimental part of this paper. The main aim of this section is to present how modifying the stopping condition, the maximum number of function evaluations affects the algorithms' performance. The intention is also to determine through a statistical analysis whether the ranking of the algorithms changes.

For this analysis, the four most recent state-of-the-art algorithms were chosen. The chosen state-of-the-art algorithms are the following: the winner of the CEC 2022 Competition EA4eig (Bujok and Kolenovsky, 2022) with Eigen crossover, NL-SHADE-RSP (Biedrzycki et al., 2022), which uses a midpoint of a population to estimate the optimum, NL-SHADE-LBC (Stanovov et al., 2022) with linear parameter adaptation bias and S-LSHADE-DP (Van Cuong et al., 2022) with dynamic perturbation for population diversity. The three solvers NL-SHADE-RSP, NL-SHADE-LBC, and S-LSHADE-DP were implemented in C++ programming language, and only EA4eig was implemented in Matlab 2021b. The experiment was carried out on a personal computer with GNU C++ compiler version 9.3.0, Intel(R) Core(TM) i5-9400 with 3.2 GHz CPU and 6 cores under Linux Ubuntu 20.04 for solvers NL-SHADE-RSP, NL-SHADE-LBC and S-LSHADE-DP and on a personal computer with Windows 11 in Matlab 2021b for the solver EA4eig.

To compare these algorithms, we chose the CEC 2022 single-objective benchmark functions. All four algorithms were used in the competition in 2022 and ranked as the first four. The benchmark consists of 12 functions: Shifted and fully Rotated Zakharov Function, Shifted and fully Rotated Rosenbrock's Function, Shifted and fully Rotated Expanded Schaffer's f6 Function, Shifted and fully Rotated Non-Continuous Rastrigin's Function, Shifted and fully Rotated Levy Function, Hybrid Function 1 (it contains $N = 3$ functions), Hybrid Function 2 ($N = 6$), Hybrid Function 3 ($N = 5$), Composition Function 1 ($N = 5$), Composition Function 2 ($N = 4$), Composition Function 3 ($N = 5$) and Composition Function 4 ($N = 6$). We made 30 independent runs for each algorithm and for two dimensions $D = 10$ and $D = 20$. We realize that the dimensions appear to be relatively low, however these are the settings provided by the competition's technical report (A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan, 2021). We used different stopping conditions for $D = 10$ and selected the following values of $maxFEs$ = 100,000; 200,000; 400,000 and 800,000 and for $D = 20$, we set the $maxFEs$ = 500,000; 1,000,000; 2,000,000 and 4,000,000.

The algorithms were observed based on how the average error of all 30 runs changes when the stopping condition *maxFEs* is modified. For each run, we are

Table 1: Rankings of the solvers according to Friedman's test and mean values for dimensions $D = 10$ with $maxFEs = 200{,}000$, and $D = 20$ with $maxFEs = 1{,}000{,}000$ for the CEC 2022 benchmark functions.

Solver	Quality of Solutions	
	$D = 10$	$D = 20$
EA4eig	2.50	2.42
NL-SHADE-LBC	2.58	**2.13**
NL-SHADE-RSP	**2.25**	2.46
S-LSHADE-DP	2.67	3.00

Table 2: Rankings of the solvers according to Friedman's test and mean values for dimensions $D = 10$ with $maxFEs = 100{,}000$, and $D = 20$ with $maxFEs = 500{,}000$ for the CEC 2022 benchmark functions.

Solver	Quality of Solutions	
	$D = 10$	$D = 20$
EA4eig	**2.08**	**2.00**
NL-SHADE-LBC	2.21	2.21
NL-SHADE-RSP	2.79	3.21
S-LSHADE-DP	2.92	2.58

recording the function error value after a certain number of function evaluations. Then we calculate the average value of the errors of all runs. The statistical analysis was done by using the non-parametric statistical tests: Friedman's test for ranking and Wilcoxon signed-rank test for a pairwise analysis to determine whether there is a significant difference between the mean values of the chosen evolutionary algorithms when using different stopping conditions.

In hindsight, we were mainly interested whether modifying the stopping criteria affects the ranking of the algorithms. For this purpose, we applied the Friedman's test. This is a non-parametric test, which can detect the statistical significance among all solvers. The Friedman's test ranks the algorithms from best to worst; the best-performing algorithm or the algorithm with the lowest mean value (quality of solution) should have the lowest rank and the largest mean value should have the highest rank.

Firstly we initially executed the algorithms using their original settings, which means the *maxFEs* for $D = 10$ was set as the $200,000$ and for the $D = 20$ the *maxFEs* was set as $1,000,000$. We ranked the solvers separately based on the dimension with Friedman's test. In Table 1, Friedman's test detects that there are no significant differences between the solvers' results, which means that their performances are comparable. The lowest rank is obtained by NL-SHADE-RSP for $D = 10$ and NL-SHADE-LBC for $D = 20$. The highest rank is obtained by S-LSHADE-DP for both dimensions. Since there is no statistical significance between the solvers, the post-hoc procedure is not needed.

To investigate the effect of decreasing the stopping condition on the algorithms' performance and rankings, we did the following. We set the *maxFEs* to a 100,000 for $D = 10$ and 500,000 for $D = 20$. As shown in Table 2, the order of ranks changes. The lowest rank is obtained by EA4eig for both dimensions. The highest rank is obtained by S-LSHADE-DP for $D = 10$ and by NL-SHADE-RSP for $D = 20$. It was determined that there is no significant difference between the solvers.

In Table 3, we show the ranking of the solvers for the increase of the stopping condition to $maxFEs = 400{,}000$ for $D = 10$ and $maxFEs = 2{,}000{,}000$ for $D = 20$. We notice that the lowest rank is obtained by NL-SHADE-LBC for both dimensions. The highest rank is obtained by S-LSHADE-DP for $D = 10$ and by NL-SHADE-RSP for $D = 20$.

In Table 4, the ranking of the algorithms is shown when *maxFEs* is increased to $maxFEs = 800{,}000$ for $D = 10$ and $maxFEs = 4{,}000{,}000$ for $D = 20$. The lowest rank is obtained by S-LSHADE-DP for $D = 10$ and by EA4eig for the $D = 20$. No matter the stopping condition, there is no significant difference between the solvers. It can be observed that by increasing the stopping condition, the differences between the ranks of the algorithms get closer.

Additionally, we show the convergence graphs of all four solvers for $D = 20$. We observed the average error and how it changes for a specific maximum number of function evaluations. For the values on the *x*-axis, we calculated them based on the rules from CEC competition (A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan, 2021). The convergence graphs are shown in Figs. 2 to 4. As it is shown in Figs. 2 to 4, the EAeig has a good convergence rate at the beginning and reaches a good solution, however, it slows down and is unable to find the optimal solution. The convergence graphs are a good indicator of the influence of the stopping condition on the algorithm, for example in Fig. 3, it is clear that NL-SHADE-LBC reaches a better solution with a larger number of function evaluations than other algorithms.

In Tables 5 to 9, we show each separate solver and the quality of the solutions it has reached on a certain benchmark function for different stopping conditions. It is evident that even with a high number of function evaluations maximum number of function evaluations, stochastic solvers do not converge to the optimum for each benchmark function. Still, we suggest using a bigger value as a *maxFEs*, since this will enable the stochastic solvers to reach a better quality of solutions.

Table 3: Rankings of the solvers according to the Friedman's test and mean values for dimensions $D = 10$ with $maxFEs = 400{,}000$, and $D = 20$ with $maxFEs = 2{,}000{,}000$ for the CEC 2022 benchmark functions.

Solver	Quality of Solutions	
	$D = 10$	$D = 20$
EA4eig	2.33	2.33
NL-SHADE-LBC	**2.00**	**1.96**
NL-SHADE-RSP	2.75	3.04
S-LSHADE-DP	2.92	2.67

Table 4: Rankings of the solvers according to the Friedman's test and mean values for dimensions $D = 10$ with $maxFEs = 800{,}000$, and $D = 20$ with $maxFEs = 4{,}000{,}000$ for the CEC 2022 benchmark functions.

Solver	Quality of Solutions	
	$D = 10$	$D = 20$
EA4eig	2.58	**1.96**
NL-SHADE-LBC	2.58	2.28
NL-SHADE-RSP	2.58	3.17
S-LSHADE-DP	**2.25**	2.89

Table 5: Average errors obtained on the 12 benchmark functions for the $D = 10$ for different stopping conditions for EA4eig.

F	EA4eig			
	$1E+05$	$2E+05$	$4E+05$	$8E+05$
f_1	0	0	7.97E-03	0
f_2	6.64E-01	1.46	1.33E+00	1.20E+00
f_3	0	0	0	0
f_4	9.29E-01	1.26028	4.64E-01	3.98E-01
f_5	0	0	0	0
f_6	5.99E-02	0.0174	3.64E-03	1.57E-03
f_7	0	0	8.01E-03	0
f_8	3.75E-01	7.09E-02	7.19E-02	2.51E-02
f_9	1.86E+02	1.86E+02	1.86E+02	1.86E+02
f_{10}	1.00E+02	1.00E+02	1.00E+02	1.00E+02
f_{11}	9.25E-09	0	0	0
f_{12}	1.47E+02	1.47E+02	1.47E+02	1.47E+02

4 CONCLUSION

In this paper, we focused on comparison of the state-off-the-art algorithms by utilizing the fixed-budget approach. The approach is usually used when comparing the evolutionary algorithms and their performances. An algorithm needs to reach a wanted solution in the given budget (number of function evaluations). This number is predetermined. Since state-of-the-art algorithms use *maxFEs* as an additional parameter, which influences the optimization process, for example in the mechanism such as, the population linear size reduction, it is also important how we set it. We followed the CEC 2022 competition

Table 6: Average errors obtained on the 12 benchmark functions for the $D = 10$ for different stopping conditions for NL-SHADE-LBC.

F	NL-SHADE-LBC			
	$1E+05$	$2E+05$	$4E+05$	$8E+05$
f_1	0	0	0	0
f_2	0.133	0.133	0.133	0.133
f_3	0	0	0	0
f_4	4.2	1.79	1.3	0.829
f_5	0	0	0	0
f_6	0.19	0.012	0.02	0.156
f_7	0.02	0	0	0
f_8	0.28	0.046	0.0176	2.74E-02
f_9	2.29E+02	2.29E+02	2.29E+02	1.86E+02
f_{10}	1.03E+02	100	100.12	100.113
f_{11}	0	0	0	0
f_{12}	164.9241	165	164.925	164.85

Table 7: Average errors obtained on the 12 benchmark functions for the $D = 10$ for different stopping conditions for NL-SHADE-RSP.

F	NL-SHADE-RSP			
	$1E+05$	$2E+05$	$4E+05$	$8E+05$
f_1	0	0	0	0
f_2	0	0	0	0
f_3	0	0	0	0
f_4	12.62	9.27	6.53	5.59
f_5	0	8.48	3.76E-01	0.0817
f_6	0.31	1.80E-01	0.06	0.046
f_7	0	3.70E-06	0	0
f_8	0.64	2.20E-01	0.11	0.06
f_9	2.29E+02	2.29E+02	2.28E+02	2.29E+02
f_{10}	1.42E+00	1.30E-01	4.11E-01	4.40
f_{11}	0	0	3.29E-07	0
f_{12}	164.95	1.60E+02	164.34	163.60

with the four state-of-the-art evolutionary algorithms EA4eig, NL-SHADE-LBC, NL-SHADE-RSP, and S-LSHADE-DP. Through our experiment, we established that the ranks of the algorithms change with increasing and/or decreasing the *maxFEs*. However, we show that there is no statistical significance between them. This indicates that the solvers are comparable and close in their performances. Therefore, it is difficult to choose the best performing algorithm for the selected benchmark. In future work, our focus will be on comparing the stochastic solvers, whose performances are very close.

Table 8: Rankings of the solvers according to the Friedman's test and mean values for dimensions $D = 10$ and the $maxFEs = 800{,}000$, and $D = 20$ $maxFEs = 4{,}000{,}000$ for the CEC 2022 benchmark functions.

Solver	Quality of Solutions	
	$D = 10$	$D = 20$
EA4eig	2.58	**1.96**
NL-SHADE-LBC	2.58	2.28
NL-SHADE-RSP	2.58	3.17
S-LSHADE-DP	**2.25**	2.89

Table 9: Average errors obtained on the 12 benchmark functions for the $D = 10$ for different stopping conditions for S-LSHADE-DP.

F	S-LSHADE-DP			
	$1E+05$	$2E+05$	$4E+05$	$8E+05$
f_1	0	0	0	0
f_2	0	0	0	0
f_3	0	0	0	0
f_4	6.28	4.72	3.54	3.00
f_5	0	0	0	0
f_6	2.66E-01	2.60E-01	2.43E-01	2.28E-01
f_7	0	0	0	0
f_8	1.49E+00	1.89E-01	1.20E01	0.02E-01
f_9	2.29E+02	2.27E+02	2.22E+02	2.06E+02
f_{10}	1.37E+00	1.25E-02	0	0
f_{11}	0	0	0	0
f_{12}	1.62E+02	1.62E+02	1.61E+02	1.61E+02

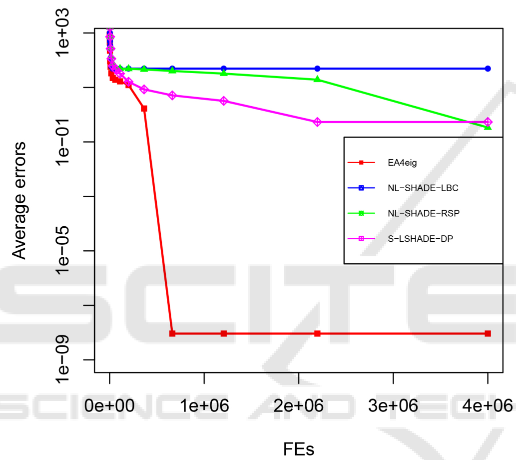

Figure 1: The convergence graph of all four chosen solvers on CEC 2022 Benchmark function f_2 for the $D = 20$.

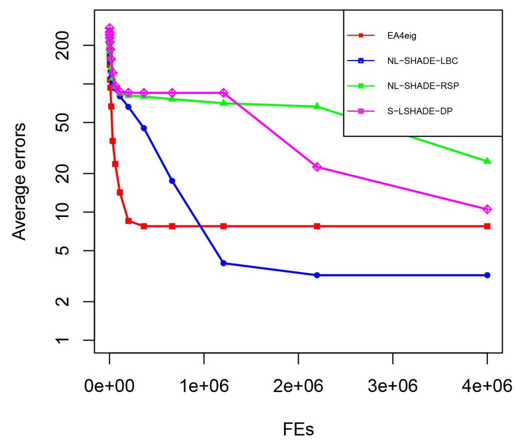

Figure 2: The convergence graph of all four chosen solvers on CEC 2022 Benchmark function f_4 for $D = 20$.

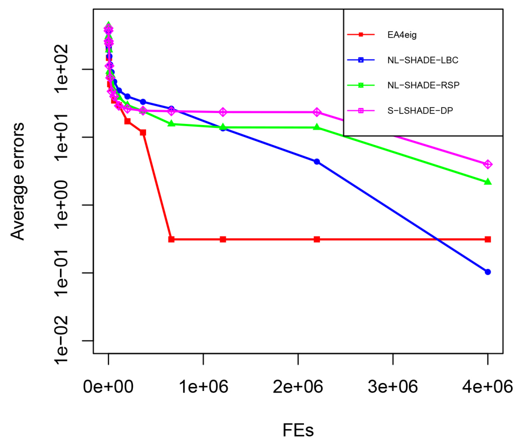

Figure 3: The convergence graph of all four chosen solvers on CEC 2022 Benchmark function f_7 for $D = 20$.

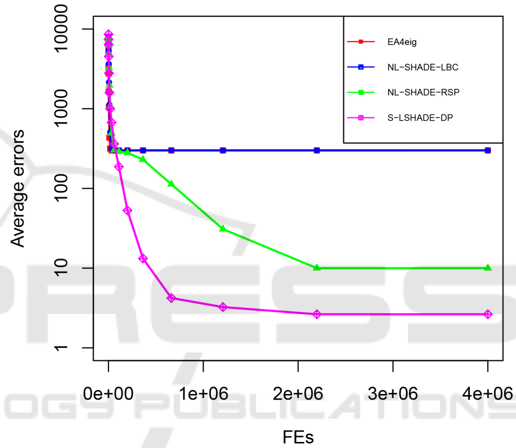

Figure 4: The convergence graph of all four chosen solvers on CEC 2022 Benchmark function f_{11} for $D = 20$.

ACKNOWLEDGEMENTS

This work was supported by the Slovenian Research Agency (Computer Systems, Methodologies, and Intelligent Services) under Grant P2-0041.

REFERENCES

A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan (December 2021). Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical report, Nanyang Technological University, Singapore.

A. W. Mohamed, A. A. Hadi, A. K. Mohamed, P. Agrawal, A. Kumar, P. N. Suganthan (November 2020). Problem Definitions and Evaluation Criteria for the CEC

2021 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical report, Nanyang Technological University, Singapore.

Bartz-Beielstein, T., Doerr, C., Berg, D. v. d., Bossek, J., Chandrasekaran, S., Eftimov, T., Fischbach, A., Kerschke, P., La Cava, W., Lopez-Ibanez, M., et al. (2020). Benchmarking in optimization: Best practice and open issues. *arXiv preprint arXiv:2007.03488*.

Benavoli, A., Corani, G., Demšar, J., and Zaffalon, M. (2017). Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. *The Journal of Machine Learning Research*, 18(1):2653–2688.

Biedrzycki, R., Arabas, J., and Warchulski, E. (2022). A version of nl-shade-rsp algorithm with midpoint for cec 2022 single objective bound constrained problems. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Birattari, M. and Dorigo, M. (2007). How to assess and report the performance of a stochastic algorithm on a benchmark problem: mean or best result on a number of runs? *Optimization letters*, 1:309–311.

Biswas, S., Saha, D., De, S., Cobb, A. D., Das, S., and Jalaian, B. A. (2021). Improving Differential Evolution through Bayesian Hyperparameter Optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 832–840. IEEE.

Brest, J., Maučec, M. S., and Bošković, B. (2016). iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1188–1195. IEEE.

Brest, J., Zamuda, A., Fister, I., and Boskovic, B. (2014). Some improvements of the self-adaptive jde algorithm. In *2014 IEEE Symposium on Differential Evolution (SDE)*, pages 1–8. IEEE.

Bujok, P. and Kolenovsky, P. (2022). Eigen crossover in cooperative model of evolutionary algorithms applied to cec 2022 single objective numerical optimisation. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Calvo, B., Shir, O. M., Ceberio, J., Doerr, C., Wang, H., Bäck, T., and Lozano, J. A. (2019). Bayesian performance analysis for black-box optimization benchmarking. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1789–1797.

Carrasco, J., Garcia, S., Rueda, M., Das, S., and Herrera, F. (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation*, 54:100665.

Cenikj, G., Lang, R. D., Engelbrecht, A. P., Doerr, C., Korošec, P., and Eftimov, T. (2022). Selector: selecting a representative benchmark suite for reproducible statistical comparison. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 620–629.

Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., and Brockhoff, D. (2021). COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144.

Hansen, Nikolaus and Auger, Anne and Brockhoff, Dimo and Tušar, Tea (2022). Anytime performance assessment in blackbox optimization benchmarking. *IEEE Transactions on Evolutionary Computation*, 26(6):1293–1305.

Herzog, J., Brest, J., and Bošković, B. (2023). Analysis based on statistical distributions: A practical approach for stochastic solvers using discrete and continuous problems. *Information Sciences*, 633:469–490.

Herzog, J., Brest, J., and Bošković, B. (2022). Performance Analysis of Selected Evolutionary Algorithms on Different Benchmark Functions. In *Bioinspired Optimization Methods and Their Applications: 10th International Conference, BIOMA 2022, Maribor, Slovenia, November 17–18, 2022, Proceedings*, pages 170–184. Springer.

Jansen, T. (2020). Analysing stochastic search heuristics operating on a fixed budget. *Theory of evolutionary computation: recent developments in discrete optimization*, pages 249–270.

Kerschke, P., Hoos, H. H., Neumann, F., and Trautmann, H. (2019). Automated algorithm selection: Survey and perspectives. *Evolutionary computation*, 27(1):3–45.

Meng, Z., Zhong, Y., and Yang, C. (2021). Cs-de: Cooperative strategy based differential evolution with population diversity enhancement. *Information Sciences*, 577:663–696.

Nikolikj, A., Trajanov, R., Cenikj, G., Korošec, P., and Eftimov, T. (2022). Identifying minimal set of Exploratory Landscape Analysis features for reliable algorithm performance prediction. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Piotrowski, A. P., Napiorkowski, J. J., and Piotrowska, A. E. (2020). Population size in particle swarm optimization. *Swarm and Evolutionary Computation*, 58:100718.

Ravber, M., Moravec, M., and Mernik, M. (2022). Primerjava evolucijskih algoritmov implementiranih v različnih programskih jezikih. *Elektrotehniski Vestnik*, 89(1/2):46–52. (In Slovene).

Stanovov, V., Akhmedova, S., and Semenkin, E. (2022). NL-SHADE-LBC algorithm with linear parameter adaptation bias change for CEC 2022 Numerical Optimization. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 01–08. IEEE.

Tanabe, R. and Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE congress on evolutionary computation (CEC)*, pages 1658–1665. IEEE.

Tušar, T., Hansen, N., and Brockhoff, D. (2017). Anytime benchmarking of budget-dependent algorithms with the COCO platform. In *IS 2017-International multiconference Information Society*, pages 1–4.

Van Cuong, L., Bao, N. N., Phuong, N. K., and Binh, H. T. T. (2022). Dynamic perturbation for population diversity management in differential evolution. In *Pro-*

ceedings of the Genetic and Evolutionary Computation Conference Companion, pages 391–394.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Zhu, X., Ye, C., He, L., Zhu, H., and Chi, T. (2023). Ensemble of Population-Based Metaheuristics With Exchange of Individuals Chosen Based on Fitness and Lifetime for Real Parameter Single Objective Optimization.