Evaluation Criteria for CEC 2026 Competition and Special Session on Bound Constrained Single and Multi-Objective Optimization Considering both Accuracy and Speed

Xuanxuan Ban [a], Hongyu Lin [a], Kangjia Qiao [a], Peng Chen [a],

Kenneth V. Price [b], Caitong Yue [a], Jing Liang [a,c],

Vladimir Stanovov [d] and Ponnuthurai N. Suganthan [e]

a. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, 450001, China.
xxuanban@163.com, linhongyuayu@163.com, qiaokangjia@yeah.net, ty1220899231@163.com

b. Vacaville, CA, USA. pricekenneth459@gmail.com

c. School of Electrical Engineering and Automation, Henan Institute of Technology, Xinxiang, 453003, China. liangjing@zzu.edu.cn

d. Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk 660037, Russia, vladimirstanovov@yandex.ru

e. Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha 2713, Qatar. p.n.suganthan@qu.edu.qa

Technical Report

December, 2025

CEC Reviewers' Requirement: CEC reviewers expect novel contributions in every submission. In addition, to be able to use the proposed comparison approach, authors also need to include one or more algorithms taken from the literature. The CEC paper is expected to include only the final results after exhausting the maximum number of function evaluations. One submission is expected to address only one of the two cases, either single objective, or multi-objective. All results should be saved using high precision. Authors of the accepted papers need to send the full results by email with a Readme.txt file. If you have any questions, you can send an email to p.n.suganthan@qu.edu.qa

1. Comparing the Bi-Objective Performance of Algorithms on SOPs and MOPs

Let n be the number of trials run by each of m algorithms on one of k problems. Each trial terminates when it reaches the maximum number of function evaluations ($FE_{\max}$) at which point its best-so-far function value (FV) is recorded (for multi-objective problems, FV is the Inverted Generational Distance (IGD) from a reference set). An algorithm's total score on a given problem is the sum of its speed score S and its accuracy score A. Both S and A are computed by performing $mn(mn-1)/2$ pairwise comparisons between all trials from the combined set of mn trials (a trial is not compared to itself).

The accuracy score A awards a point (1.0) to the trial with the better final FV, or a half point (0.5) to both trials if their final FVs are equal. Once all comparisons have been performed, an algorithm's accuracy score is the total number of points won by its trials.

For example, Fig. 1 shows the case where $m = 2$ and $n = 4$ for a total of 8 trials—4 for algorithm P and 4 for algorithm Q. When we compare all trials by their final FVs along the ordinate at $FE_{\max}$, we find that algorithm P's trials p_1, p_2, p_3 and p_4 scored 6, 4, 3 and 1 point, respectively, for an accuracy score of $A_P = 14$. Similarly, algorithm Q's trials q_1, q_2, q_3 and q_4 of algorithm Q scored 7, 5, 2 and 0 points, respectively, for a score of $A_Q = 14$. In this case, both algorithms are equally accurate because $A_P = A_Q$.

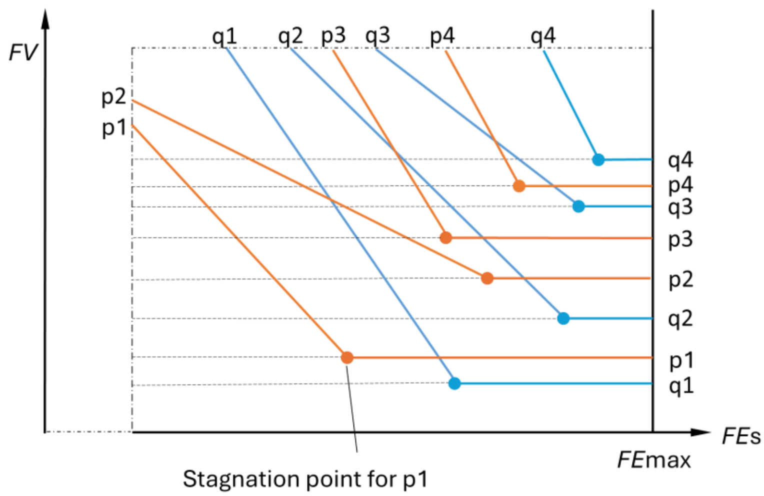

Figure 1: A convergence plot showing 8 trials from two algorithms and their stagnation points.

Most trials, like those in Fig. 1, stop improving before reaching $FE_{\max}$. By searching a trial's history, we can find its stagnation point, i.e. the *FE* cut-point at which it first reached its final *FV*. We can then say that the faster of two trials is the one that reached the higher of their two final *FV*s first. For example, Fig. 1 shows that trial q_1 is faster than trial p_2 because it reached p_2's final *FV* before p_2 did, but q_1 is slower than p_1 because it reached p_1's final *FV* after p_1 did. The speed score *S* awards the faster trial a point, or a half point to each trial if they both reached the higher of their two *FV*s at the same *FE* cut-point. Once all trials have been compared, *S* is the total number of points accrued by an algorithm's trials.

When all trials in Fig. 1 are compared by their speed, trials p_1, p_2, p_3 and p_4 score 7, 5, 4 and 2 points, respectively, while trials q_1, q_2, q_3 and q_4 score 6, 3, 1 and 0 points, respectively. Since the speed scores for algorithms P and Q are $S_P = 18$ and $S_Q = 10$, respectively, we conclude that algorithm P is faster than algorithm Q.

For this idealized example, the final problem scores for P and Q are $S_P + A_P = 32$ and $S_Q + A_Q = 24$, which indicates that algorithm P had the better bi-objective performance on this pseudo-problem. An algorithm's final score for the competition is the sum of its *k* problem scores.

2. Bound Constrained Single Objective Optimization Problems (SOPs)

Test Problems: The 29, 30-dimensional, real-parameter numerical optimization problems in CEC2017 [1] are the test problems. The codes can be downloaded from the website: https://github.com/P-N-Suganthan/CEC2017-BoundContrained. Please note that the problems should be treated as black box problems.

Number of Trials/Problem: 25 independent runs. (Do not run many 25 runs to pick the best run).

Maximum Number of Function Evaluations: $FE_{\max} = 10000*D$, where *D* is the dimensionality of the optimization problems. The authors should clearly state if they use training. If so, all *FE*s during training must be counted within the total *FE* budget. We will also include speed in the ranking procedure.

Search Range: $[-100, 100]^D$

Population Size: You are free to choose a population size that is appropriate for your algorithm.

Sampling Points: Record the best-so-far *FV* (function value—not error value) every 10**D* evaluations for each run. For example, the maximum number of function evaluations $FE_{\max}$ is 10000**D*, so save 1000 *FV*s.

Algorithm Complexity: The evaluation of algorithm complexity requires the calculation of two indicators T_1 and T_2, which are calculated as follows:

1) $T_1 = (\sum_{i=1}^{29} t_i^1)/29$, where t_i^1 is the time to execute 10000 evaluations of problem *i*.

2) $T_2 = (\sum_{i=1}^{29} t_i^2)/29$, where t_i^2 is the time taken *by the algorithm* to execute 10000 evaluations of problem *i*.

The complexity of the algorithm is computed as: $(T_2 - T_1)/T_1$.

Presentation of Results: Save the results in the form of Table 1, where $FV_{\min}$ is the *best-so-far* function value (not error value) of each run at each sampling point, i.e. every 10**D* *FE*s. Thus, for each algorithm, 29 files (one for each test function) should be zipped and sent to organizers.

Table 1

Results saved in "PaperID_FJ_Min_*FV*.mat" where J=1,2,3,...29 problems.

	Run 1	Run 2	Run 3	...	Run 25
Min_*FV* at Initialization *FE*s					
Min_*FV* at 10**D* FEs Min_*FV* at 20**D* FEs					
...					
...					
Min_*FV* at $FE_{\max}$					

After submitting the initial version of their papers, all participants are allowed to improve their algorithms until the final accepted paper submission deadline set by the conference. Authors are required to submit their results in the prescribed format to the organizers as soon as possible after submitting the final version of paper. Please refer to the template in the following link for the format of the submitted results:

https://github.com/P-N-Suganthan/ 2025-CEC/blob/main/results_data_8.25.zip.

3. Multi-Objective Optimization Problems (MOPs) without Constraints

The scoring approach for MOPs is the same as for SOPs, except that the function value *FV* is the inverted generational distance (*IGD*) from a reference set.

Test Problems: The test suite consists of the 10 multi-objective problems in [2]. Problems should be treated as black box problems.

Number of Trials/Problem: 30 independent runs per problem.

Maximum Number of Function Evaluations: The maximum number of evaluations for each function is 100000. The authors should clearly state if they use training. If so, all *FE*s used during training must be counted within the total *FE* budget. We will include speed in the evaluation procedure.

Pareto Front Size: The final PF (Front 1) is expected to have a size of 100. Compute *IGD* results using the maximal 100 feasible individuals. The recommended population size is 100.

Sampling Points: Record the best-so-far *IGD* values every 200 function evaluations. Since the maximum number of evaluations is 100000, 500 *IGD* values are saved.

Encoding: If the algorithm requires encoding, then the encoding scheme should be independent of the specific problems and governed by generic factors such as the search ranges.

Algorithm Complexity: The evaluation of algorithm complexity depends on two indicators T_1 and T_2, which are calculated as follows:

1) $T_1 = (\sum_{i=1}^{29} t_i^1)/29$, where t_i^1 is the time to execute 10000 evaluations of problem *i*.

2) $T_2 = (\sum_{i=1}^{29} t_i^2)/29$, where t_i^2 is the time taken *by the algorithm* to execute 10000 evaluations of problem *i*.

The complexity of the algorithm is computed as: $(T_2 - T_1)/T_1$.

Presentation of Results: As shown in Table 2, 501 *IGD* values for each of the 30 runs are required for each problem. For example, the results of PaperID for problem RCMJ, the files name should be "PaperID RCMJ *IGD*.txt", where *IGD* values are saved, respectively. Thus, 10 * 30 = 300 files should be zipped and sent to the organizers, where 10 is the total number of test functions, and 30 is the number of trials per problem. Please refer to the template for submitting result in the following link: https://github.com/P-N-Suganthan/2025-CEC/blob/ main/results_data_8.25.zip.

Table 2

Results saved in "PaperID RCMJ *IGD*.txt" where J=1,2,...,10 problems

	Run 1	Run 2	Run 3	...	Run 30
IGD at Initialization *FE*s					
IGD at Sampling Point 1					
IGD at Sampling Point 2					
.. .					
.. .					
IGD at Sampling Point 500, 100K *FE*s					

After submitting the initial version of their papers, all participants are allowed to improve their algorithms until the final accepted paper submission deadline set by the conference. Authors are required to submit their results in the prescribed format to the organizers as soon as possible after submitting the final version of paper.

References

[1] Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P., 2016. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization, in: Technical Report. Nanyang Technological University Singapore, pp. 1–34.

[2] Li, H., Deb, K., Zhang, Q., Suganthan, P.N., Chen, L., 2019. Comparison between moea/d and nsga-iii on a set of novel many and multi-objective benchmark problems with challenging difficulties. Swarm and Evolutionary Computation 46, 104–117.