Evaluation criteria for CEC 2026 competition and special session on bound constrained single and multi-objective optimization considering 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 U-score 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 multiobjective. 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. Bi-Objective Performance of Algorithms

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). A total score of an algorithm 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 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, we find that the trials p_1, p_2, p_3 and p_4 of algorithm P scored 6, 4, 3 and 1 point, respectively, for an accuracy score of $A_P = 14$. Similarly, the 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 $A_P = A_Q$.

Most trials, like those in Fig. 1, stop improving before reaching $FE_{\max}$. By searching the history of a trial, 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 FVs first. For example, Fig. 1 shows that trial q_1 is faster than trial p_2 because it reached the final FV before p_2 did, but q_1 is slower than $p1$ because it reached the 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 FVs at the same FE cut-point. Once all trials have been compared, S is the total number of points accrued by the trials of an algorithm.

For example, 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 final score of an algorithm for the competition is the sum of its k problem scores.

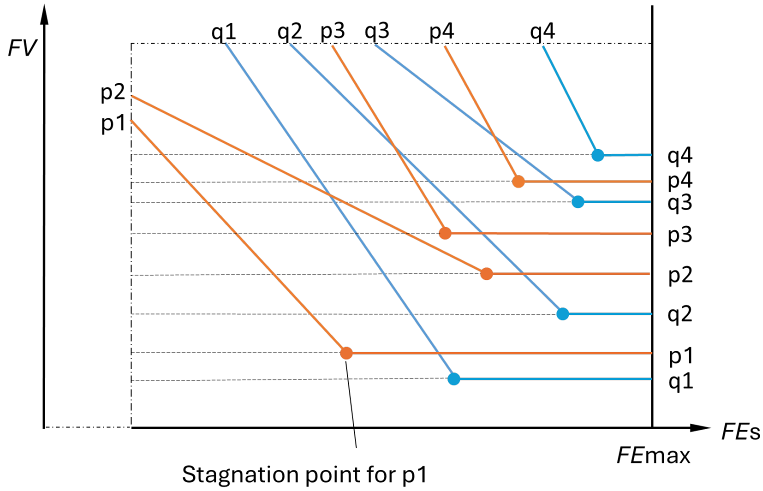

Figure 1: A convergence plot showing 8 trials from two algorithms and their stagnation points.

2. Bound constrained single objective optimization problems (SOPs)

Test Problems: The 29 real-parameter numerical optimization problems with $30D$ in CEC2017 [1] are adopted as 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 blackbox 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 whether they use training. If so, all FEs during training must be counted within the total FEs budget. We will also include speed in the ranking procedure.

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

Population Size: You are free to have an appropriate population size to suit your algorithm while not exceeding the $FE_{\max}$s.

Sampling Points: The best-so-far FV (Function Value) every $10*D$ evaluations will be recorded for each run. For example, the maximum number of function evaluations $FE_{\max}$s is $10000*D$, then 1000 FVs should be saved.

Target Error Values: The target error value, TGT_FV for each problem, will be determined after the competition. Hence, all algorithms should be executed until the Maximum number of Function Evaluations ($FE_{\max}$s) are consumed.

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

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 Initialisation FEs					
Min_*FV* at 10*D FEs					
Min_*FV* at 20*D FEs					
...					
...					
Min_*FV* at $FE_{\max}$s					

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

2) $T_2 = (\sum_{i=1}^{29} t_i^2)/29$, t_i^2 is the complete computing time of the algorithm with 10000 evaluations for the problem i.

The complexity of the algorithm is reflected by: T_1, T_2 and $(T_2 - T_1)/T_1$

Presentation of Results: The results can be saved in the form of Table 1, where $FV_{\min}$ is the best error value of each run at each sampling point. The value should be recorded every $10{*}D$ FEs. Thus, for each algorithm, 29 files should be zipped and sent to organizers, where 29 represents the total number of test functions.

Note that all participants are allowed to improve their algorithms further after submitting the initial version of their papers 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 after submitting the final version of paper as soon as possible. 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. Bound constrained multi-objective optimization problems (MOPs)

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: We adopt the benchmark of [2] including 10 multi-objective problems to rank the optimizers of MOPs without constraints. Please note that the problems should be treated as blackbox problems.

Number of Trials/Problem: 30 independent runs.

Maximum Number of Function Evaluations: The maximum number of evaluations are set to 100000 for each function. The authors should clearly state whether they use training. If so, all FEs

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 Initialisation FEs					
IGD at Sampling Point 1					
IGD at Sampling Point 2					
...					
...					
IGD at Sampling Point 500, 100K FEs					

used during the training must be counted within the total FEs 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 maximal 100 feasible individuals. The recommended population size is 100.

Sampling Points: The best-so-far *IGD* values will be recorded once every 200 function evaluations. For example, if the maximum number of evaluations $FE_{\max}$s is 100000, then 500 *IGD* values are saved.

Target *IGD* Values: The target *IGD* value will be determined after the competition. Hence, all algorithms should be executed until Maximum number of Function Evaluations ($FE_{\max}$s) are consumed.

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 requires the calculation of two indicators T_1 and T_2, which are calculated as follows:

1) $T_1 = (\sum_{i=1}^{10} t_i^1)/10$, t_i^1 is the computing time of 10000 evaluations for problem i.

2) $T_2 = (\sum_{i=1}^{10} t_i^2)/10$, t_i^2 is the complete computing time for the algorithm with 10000 evaluations for problem i.

The complexity of the algorithm is reflected by: T_1, T_2 and $(T_2 - T_1)/T_1$

Presentation of Results: To compare and evaluate the algorithms participating in the competition, it is necessary that the authors email the results as shown in Table 2 to the organizers after submitting the final version of the accepted paper.

According to 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 Inverted Generational Distance values are saved, respectively. Thus, 10 * 30 =

300 files should be zipped and sent to the organizers, where 10 represents the total number of test functions, and 30 represents the number of trials per problem.

Note that all participants are allowed to improve their algorithms further after submitting the initial version of their papers until the final accepted paper submission deadline set by the conference. Authors are required to submit their results in the introduced format to the organizers after submitting the final version of paper as soon as possible. In summary, the results should be saved as shown in Table 2. 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`.

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.