Evaluation Criteria for CEC 2026 Competition and Special Session on 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@gs.zzu.edu.cn, 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 scoring comparison approach, authors also need to include one or more algorithms taken from the literature. CEC paper is expected to include only the final results after exhausting the maximum number of function evaluations. One submission is expected to address either single or multi-objective problems. All results should be saved using high precision. Authors of 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 CSOPs and CMOPs

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 (Max_FEs). For constrained single objective problems (CSOPs), we record both the best-so-far error value (Min_EV) and the population's lowest overall constraint violation (LCV),

$$LCV = min\colon CV(P_i), \qquad i = 1, 2 \dots, NP, \tag{1}$$

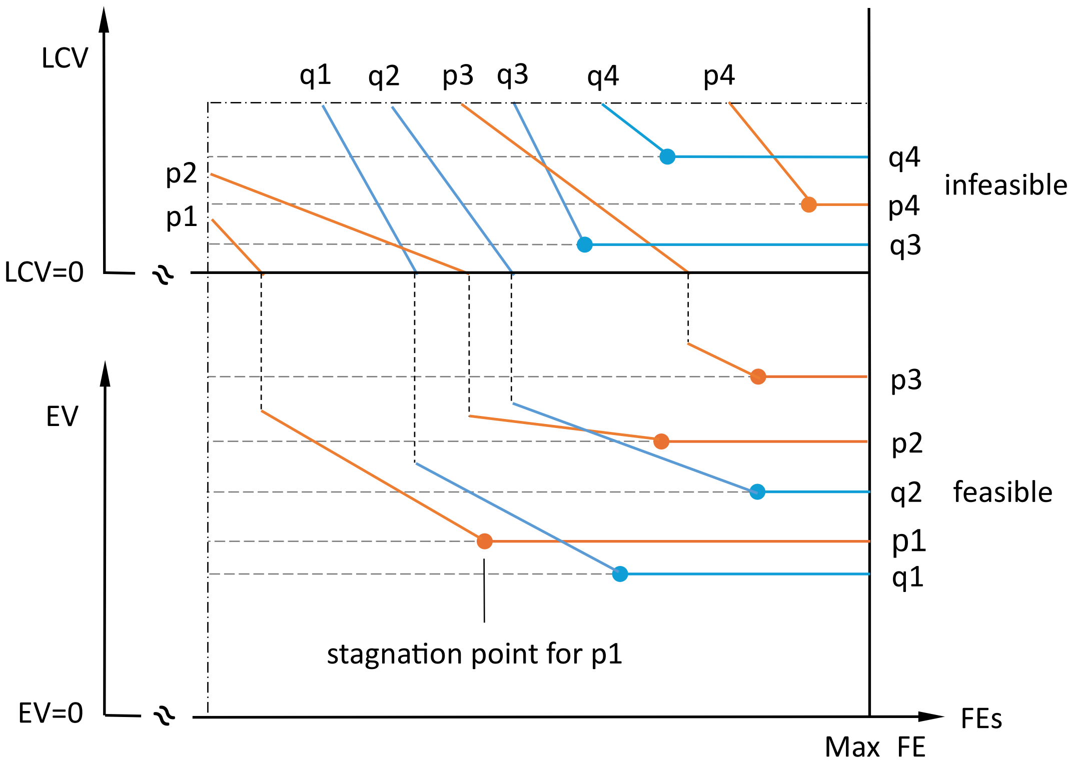
where *NP* is the population size and $CV(P_i)$ is the overall constraint violation value of the i^{th} individual in the population *P*. For constrained multi-objective problems (CMOPs), we record the population's Inverted Generational Distance (IGD), along with the population's mean overall constraint violation (MCV) (2).

$$MCV = \frac{\sum_{i=1}^{PF} CV(P_i)}{PF}, \tag{2}$$

In Eq. 2, PF is the size of the final Pareto front-1. We describe the scoring method below as it applies to CSOPs, but the same method also applies to CMOPs except that Min_EV and LCV are replaced by IGD and MCV, respectively.

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). If both trials are feasible, the accuracy score *A* awards a point (1.0) to the trial with the better final Min_EV, or a half point (0.5) to both trials if their final error values (EVs) are equal. If both trials are infeasible, then the trial with the lower LCV receives a point, or both trials receive a half point if their LCVs are equal. Finally, if one trial is feasible and the other is infeasible, then the feasible trial receives the point. 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 both their final EVs and LCVs along the ordinate at Max_FEs, 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 scored 7, 5, 2 and 0 points, respectively, for an accuracy score of A_Q = 14. In this case, both algorithms are equally accurate because $A_P = A_Q$.



Fig, 1: A convergence plot showing 8 trials from two algorithms and their stagnation points. In this example, all trials are initially infeasible, but only 5 become feasible before reaching Max_FE.

Most trials, like those in Fig. 1, stop improving before reaching Max_FE. 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 EV or LCV. For example, if we compare feasible trials, like p_1, p_2 and q_1 in Fig. 1, we find that trial q_1 is faster than trial p_2 because it reached p_2's final EV *before* p_2 did, but q_1 is slower than p_1 because it reached p_1's final EV *after* p_1 did. Similarly, if we compare infeasible trials, like q_4 and p_4, we find that p_4 is slower than q_4 because it reached q_4's LCV after q_4 did. Finally, when one trial is feasible and the other is infeasible, like trials p3 and q3 in Fig. 1, the infeasible trial q3 is faster than feasible trial p3 in this example because p3 was slower to reach q3's LCV. All of the remaining feasible trials, p1, p2, q1 and q2, were faster than any of the infeasible trials.

The speed score S awards the faster trial a point, or a half point if they both reached the higher of their two EVs or LCVs 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, 4, 2 and 0 points, respectively, while trials q_1, q_2, q_3 and q_4 score 6, 5, 3 and 1 point, respectively. Since the speed scores for algorithms P and Q are $S_P = 13$ and $S_Q = 15$, respectively, we conclude that algorithm Q is marginally faster than algorithm P.

In this idealized example, the final problem scores for P and Q are $S_P + A_P = 27$ and $S_Q + A_Q = 29$, which indicates that algorithm Q had slightly better bi-objective performance on this pseudo-problem. An algorithm's final score for the competition is the sum of its k problem scores.

Test Problems: The 28 constrained, 30-dimensional, real-parameter optimization problems in CEC2017 [4] are the test problems. The code can be found at the PlatEMO [1] website. Please note that the problems should be treated as black box problems.

Number of Trials/Problem: 25 independent runs.

Maximum Number of Function Evaluations: Max_FE = 20000*D where D is the dimensionality of the optimization problems. Authors should clearly state whether they use training. If so, all FEs used during training must be counted within the total FE budget. We will also include speed in the ranking procedure.

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

Sampling Points: Record the best-so-far Min_EV and LCV at initialization and every 10*D evaluations. Since the maximum number of function evaluations is 20000*D, record 2001 EVs and LCVs for trials with at least one feasible solution. When the whole population is infeasible, record the population's LCV value and enter "NaN" as the result for Min_EV.

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

[1] website: https://github.com/BIMK/PlatEMO or https://github.com/P-N-Suganthan/CEC2017).

1) $T_1 = (\sum_{i=1}^{28} t_i^1)/28$, where t_i^1 is the time to execute 10000 evaluations of problem i.
2) $T_2 = (\sum_{i=1}^{28} t_i^2)/28$, 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 your results as shown in Table 1, in which the first entry is for the evaluation of the initial population. The cumulative FEs at each sampling point should be saved in the first column. Additionally, the corresponding Min_EV and LCV results should be saved in the second and third columns, respectively. So, for a function, one run requires one file in .mat format. For each algorithm, 28 files should be zipped and sent to the organizers, where 28 is the total number of test problems.

Please note that if no feasible solution exists at a sampling point, enter “NaN” for the Min_EV result. Additionally, for the LCV value in Table 1, CV *should be directly calculated through the constraint functions*, rather than the author’s normalized CV result.

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 required format to the organizers as soon as possible after submitting the final version of their paper.

Table 1

Results saved in “PaperID_CPJ.mat” where J=1, 2...,28 problems

FEs	Run 1		Run 2		...	Run 25	
	Min_EV	LCV	Min_EV	LCV		Min_EV	LCV
After Initialization							
Sampling Point 1 at 1*10*D* FEs							
Sampling Point 2 at 2*10*D* FEs							
...							
Last Sampling Point at Max_FE							

2. Constrained Multi-Objective Optimization Problems (CMOPs)

Test Problems: The latest constrained multi-objective optimization problems with scalable decision space constraints (SDC problems) [2] are the test problems. The SDC benchmark contains 15 problems. The codes can be downloaded from the website[2]. Please note that the problems should be treated as black box problems.

Number of Trials: 30 independent runs.

Maximum Number of Function Evaluations: For each function, the maximum number of evaluations is Max_FE = 200000. Authors need to clearly state whether they use training. If so, all FEs used during training must be counted within the total FE budget. We will include speed in the ranking procedure.

Pareto Front Size: The final Pareto front (i.e. front 1) should have a size of 100. Compute the IGD results using the maximal 100 feasible individuals. The recommended population size is 100.

Parameter Setting: The dimension is set to D = 30 for each SDC function.

Sampling Points: The IGD values should be recorded at initialization and every 200 function evaluations. Since the maximum number of evaluations Max_FE = 200000, save 1001 IGD values.

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

1) $T_1 = \left(\sum_{i=1}^{15} t_i^1\right)/15$, where t_i^1 is the time to execute 10000 evaluations of problem i.

2) $T_2 = \left(\sum_{i=1}^{15} t_i^2\right)/15$, 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: To compare and evaluate the algorithms participating in the competition, it is necessary that the authors email the results to the organizers in the format shown in Table 2 after submitting the final version of the accepted papers.

[2] https:// github.com/cilabzzu/Codes/blob/main/SDC, which should be ran on PlatEMO (website: https://github.com/BIMK/PlatEMO). Please note that the latest PlatEMO has contained the codes of SDC functions, and you can directly use these codes on PlatEMO.

In Table 2, at each sampling point, i.e. every 200 FEs, IGD and MCV should be computed and saved in the second and third columns, respectively. Please note that if no feasible solution exists at a sampling point, the IGD result should be expressed by "NaN". Additionally, when computing the MCV value for Table 2, CV should be *directly calculated through the constraint functions*, rather than the author's normalized CV result. For one algorithm, 15 files in .mat format (one for each problem) should be zipped using the paper number as the file name and sent to the organizers.

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 final results in the prescribed format to the organizers as soon as possible after submitting the final version of paper.

Table 2

Results saved in "PaperID_SDCJ.mat" where J=1,2,...,15 problems.

FEs	Run 1		Run 2		...	Run 30	
	IDG	MCV	IDG	MCV		IDG	MCV
After Initialization							
Sampling Point 1 at 200 FEs							
Sampling Point 2 at 400 FEs							
...							
Sampling Point 1000 at 200K FEs							

References

[1] Price, K.V., Kumar, A., Suganthan, P.N., 2023. Trial-based dominance for comparing both the speed and accuracy of stochastic optimizers with standard non-parametric tests. Swarm and Evolutionary Computation, 101287, Vol. 78, April.

[2] Qiao, K., Liang, J., Yu, K., Yue, C., Lin, H., Zhang, D., Qu, B., 2023a. Evolutionary constrained multi-objective optimization: Scalable high dimensional constraint benchmarks and algorithm. IEEE Transactions on Evolutionary Computation, 1–1doi:10.1109/TEVC.2023.3281666.

[3] Qiao, K., Wen, X., Ban, X., Chen, P., Price, K., Suganthan, P., Liang, J., Wu, G., Yue, C., 2023b. Evaluation criteria for CEC 2024 competition and special session on numerical optimization considering accuracy and speed. Zhengzhou University, Central South University, Henan Institute of Technology, Qatar University, Technical Report.

[4] Wu, G., Mallipeddi, R., Suganthan, P.N., 2017. Problem definitions and evaluation criteria for the cec 2017 competition on constrained real parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report.