Evaluation criteria for CEC 2025 competition and special session on 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],
Ponnuthurai N. Suganthan [d]

a. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, 450001, China, qiaokangjia@gs.zzu.edu.cn, xxuanban@163.com, 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. KINDI Computing Research Center, College of Engineering, Qatar University, Doha 2713, Qatar, p.n.suganthan@qu.edu.qa

Technical Report

December, 2024

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. 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 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 query, you can send an email to p.n.suganthan@qu.edu.qa.

1. Introduction to the U-score approach

Traditionally, algorithm performance evaluation embraces one of the two complementary paradigms. The first, often referred to as the ’fixed target’ scenario, records of the number of function evaluations (FEs) necessary for a trial to achieve a predetermined minimum function error value (Min_EV). The second, termed the ’fixed cost’ scenario, records the function error value (EV) of a trial once it exhausts a stipulated maximum number of function evaluations (Max_FEs). Nonetheless, a notable lacuna has persisted in the assessment landscape, with a dearth of methodologies concurrently considering both Min_EV and Max_FEs. To resolve the difficulty of pre-specifying a target value-to-reach (TGT) for each test problem, we equate TGT to the mean of the results achieved by all algorithms, thereby having a balanced pass-fail rate always.

The incorporation of both convergence accuracy and speed within the U-score approach [1] provides a comprehensive perspective on algorithmic performance, thereby facilitating effective comparative analyses and rankings across a multitude of algorithms by considering each run of each algorithm. Noteworthy is the fact that in the context of a binary competition involving just two algorithms, the U-score approach effectively simplifies to the Mann-Whitney U statistic.

Based on the U-score approach, we set up four groups of algorithmic ranking competitions, i.e., 1) Constrained single objective optimization problems. 2) Constrained multi-objective optimization problems.

2. Constrained single objective optimization problems (COPs)

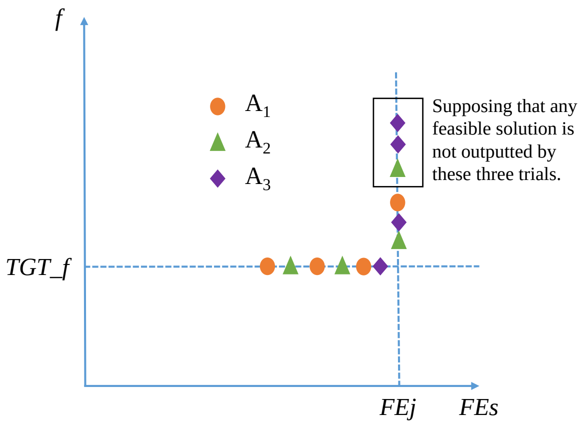

Figure 1: Three algorithms, A1–A3, run four trials each on a COP. A single run terminates when it reaches Max_FEs. TGT_f and FE_j will be determined later. All trial results can be ordered from the best to the worst.

U-score approach for constrained single objective optimization problems: For constrained single objective optimization problems, the minimum objective function value f_{min} is the performance indicator at each sampling point, if the corresponding solution is feasible. The feasibility condition

Trial	●	▲	●	▲	●	◆	▲	◆	●	▲	◆	◆	*SR*	U-score
Ranks	12	11	10	9	8	7	6	5	4	3	2	1	78	
A1	12		10		8				4				34	24
A2		11		9			6			3			29	19
A3						7		5			2	1	15	5

The "correction factor" (*cf*) is $n(n+1)/2 = 4 * 5/2 = 10$, where n denotes the number of trails. SR denotes the sum of ranks. The scores of algorithms are calculated by the "SR" minus the "*cf*" according to the U-score algorithm.

Figure 2: U-score ranks for CMOPs.

is defined in [3]. To exemplify the U-score ranking method's confluence of convergence speed and accuracy on COPs, an illustrative example is depicted in Figure 1. This figure portrays three ranking algorithms, designated as A1 to A3, each assigned a distinct color and shape. For each algorithm, four distinct runs were executed, yielding a total of 12 trials. These trials are stratified into two categories: 1) those that successfully converged to the *TGT_f* target value, and 2) those that fail to attain this target within the stipulated Max_FEs. (PS: As the TGT_f is undefined as yet, authors are asked to execute to the Max_FEs and save results using high precision.)

In adherence to the stipulated procedures and criteria, the specific rankings of algorithms A1 to A3 are delineated within Figure 2, which presents the tabulated U-score results. To be specific, the scoring of algorithms A1, A2, and A3 is determined through the summation of their respective rankings. Evidently, algorithm A1 emerges as the victor, amassing a total score of 24 based on the U-score approach, thereby securing the topmost rank. In juxtaposition, A2 secures the second rank, and A3 the third. This outcome underscores the supremacy of algorithm A1, attributed to its swifter convergence velocity and its propensity to attain lower f values.

Please note that some algorithms might not output any feasible solution at a particular sampling point, that is, the entire population is infeasible. If two or more algorithms do not output any feasible solution at a sampling point, we will rank these trials based on the lowest value of the overall constraint violation (LCV) of the best solution. The algorithm with smaller LCV value is better. The formula for calculating LCV for a run of an algorithm is as follows:

$$LCV = \mathbf{min:}CV(P_i), i = 1, ..., NP \tag{1}$$

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. Please note that when recording the LCV value in Table 1, CV should be directly calculated through the constraint functions, rather than the author's normalized CV result.

Test Problems: The 28 constrained real-parameter optimization problems with $30D$ in CEC2017 [3] are adopted as test problems. The code can be found from the PlatEMO [1]

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

Number of Trials/Problem: 25 independent runs.

Maximum Number of Function Evaluations: Max_FEs = 20000*D, where D is the dimensionality of the optimization problems.

Population Size: You are free to have an appropriate population size to suit your algorithm while not exceeding the Max_FEs.

Sampling Points: Record f_{min} values and LCV every 10*D evaluations. For example, if the maximum number of function evaluations Max_FEs is 20000*D, then 2000 f_{min} values are recorded for trials with one or more feasible solutions. When the whole population is infeasible, the lowest LCV value of the population should be saved at the respective sampling points.

Target Error Values: The target error value will be determined after the competition. Hence, all algorithms should be executed until Maximum number of Function Evaluations (Max_FEs) 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:

1) $T_1 = (\sum_{i=1}^{28} t_i^1)/28$, t_i^1 is the computing time of 10000 evaluations for problem i.

2) $T_2 = (\sum_{i=1}^{28} t_i^2)/28$, 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: 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. Meanwhile, the corresponding f_{min} 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. Please note that if no feasible solution exists at one sampling point, the f_{min} result should be expressed by "NaN".

Thus, for each algorithm, 28 files should be zipped and sent to the organizers, where 28 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 introduced format to the organizers after submitting the final version of paper as soon as possible.

3. Constrained multi-objective optimization problems (CMOPs)

U-score approach for constrained multi-objective optimization problems: For constrained multi-objective optimization problems, we introduce the Inverted Generational Distance (*IGD*)

Table 1
Results saved in "PaperID_CPJ.mat" where J=1,2,...,28 problems

FEs	Run1		Run2		...	Run25	
	f_{min}	LCV	f_{min}	LCV		f_{min}	LCV
at Initialisation FEs							
Sampling Point 1, FEs=1*10D							
Sampling Point 2, FEs=2*10D							
...							
Last Sampling Point, Max_FEs							

values as an indicator. To exemplify the U-score ranking method's confluence of convergence speed and accuracy on CMOPs, an illustrative example is depicted in Figure 3. This figure portrays three ranking algorithms, designated as A1 to A3, each assigned a distinct color and shape. For each algorithm, four distinct runs were executed, yielding a total of 12 trials. These trials are stratified into two categories: 1) those that successfully converged to the TGT_IGD target value, and 2) those that fail to attain this target within the stipulated Max_FEs. (PS: As the TGT_IGD is undefined as yet, authors are asked to execute to the Max_FEs and save results using high precision.)

In adherence to the stipulated procedures and criteria, the specific rankings of algorithms A1 to A3 are delineated within Figure 4, which presents the tabulated U-score results. To be specific, the scoring of algorithms A1, A2, and A3 is determined through the summation of their respective rankings. Evidently, algorithm A1 emerges as the victor, amassing a total score of 24 based on the U-score approach, thereby securing the topmost rank. In juxtaposition, A2 secures the second rank, and A3 the third. This outcome underscores the supremacy of algorithm A1, attributed to its swifter convergence velocity and its propensity to attain lower *IGD* values.

Please note that some algorithms might not output any feasible solution on one run, that is, the entire population is infeasible. If at least two trials do not output any feasible solutions as such in Figure 3, we will rank these trials based on the mean value of overall constraint violation value of the populations (*MCV*). The trial with a smaller *MCV* value is better. The formulate of calculating *MCV* is as follows:

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

where PF is the required final front-1 size and $CV(P_i)$ is the constraint violation value of the i^{th} individual in the population P. Please note that when recording the MCV value in Table 2, CV should be directly calculated through the constraint functions, rather than the author's normalized CV result.

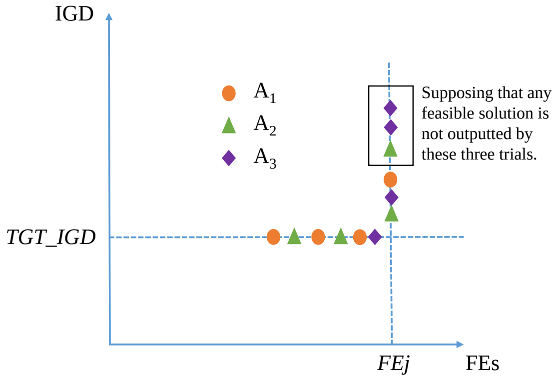

Figure 3: Three algorithms, A1–A3, run four trials each on a CMOP. A single run terminates when reaches Max_FEs. TGT_IGD and FE_j will be determined later. All trial results can be ordered from the best to the worst.

Trial	●	▲	●	▲	●	◆	▲	◆	●	▲	◆	◆	*SR*	U-score
Ranks	12	11	10	9	8	7	6	5	4	3	2	1	78	
A1	12		10		8				4				34	24
A2		11		9			6			3			29	19
A3						7		5			2	1	15	5

The "correction factor" (*cf*) is $n(n+1)/2 = 4 * 5/2 = 10$, where n denotes the number of trails. SR denotes the sum of ranks. The scores of algorithms are calculated by the "SR" minus the "*cf*" according to the U-score algorithm.

Figure 4: U-score ranks for CMOPs.

Test Problems: The latest constrained multiobjective optimization problems with scalable decision space constraints (SDC problems) [2] are adopted as test problems. SDC benchmark contains 15 problems. The codes can be downloaded from the website[2].

Number of Trials: 30 independent runs.

Maximum Number of Function Evaluations: The maximum number of evaluations are set to 200000 for each function.

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

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

Sampling Points: The *IGD* values will be recorded once every 200 function evaluations. For example, if the maximum number of evaluations Max_FEs is 200000, then 1000 *IGD* values are saved.

Target IGD Values: The target IGD value will be determined after the competition. Hence, all algorithms should be executed until the Maximum number of Function Evaluations (Max_FEs) are consumed. Please note that the minimal IGD value is unknown for multiobjective optimization

[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.

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

	Run1		Run2		...	Run 30	
	IGD	MCV	IGD	MCV		IGD	MCV
at initialization FEs							
Sampling point 1							
Sampling point 2							
...							
...							
Sampling point 1000							

problems. So, the mean or median IGD value of all trials from all algorithms participating in the competition will be set as the target IGD value.

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

1) $T_1 = (\sum_{i=1}^{15} t_i^1)/15$, t_i^1 is the computing time of 10000 evaluations for problem i.

2) $T_2 = (\sum_{i=1}^{15} t_i^2)/15$, 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 in the format as shown in Table 2 to the organizers, after submitting the final version of the accepted papers.

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", while *MCV* value should be recorded.

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 after submitting the final version of paper as soon as possible.

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., 2023. Evolutionary constrained multiobjective optimization: Scalable high-dimensional constraint benchmarks and algorithm. IEEE Transactions on Evolutionary Computation , 1–1doi:10.1109/TEVC.2023.3281666.

[3] 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 .