Evaluation criteria for CEC 2025 competition and special session on bound constrained single and multi-objective optimization considering accuracy and speed

Xuanxuan Ban [a], Kangjia Qiao [a], Peng Chen [a], Hongyu Lin [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 one of the four cases. 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 [3] 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 two groups of algorithmic ranking competitions, i.e., 1) Bound constrained single objective optimization problems. 2) Bound constrained multi-objective optimization problems.

2. Bound constrained single objective optimization problems (SOPs)

U-score approach for single objective optimization problems [3]: To exemplify the U-score ranking method's confluence of convergence speed and accuracy on SOPs, 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 target error value TGT_EV and 2) those that fail to attain this target within the stipulated Max_FEs, concluded upon reaching the limit of Max_FEs. (PS: As the Min_EV 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 Min_EV.

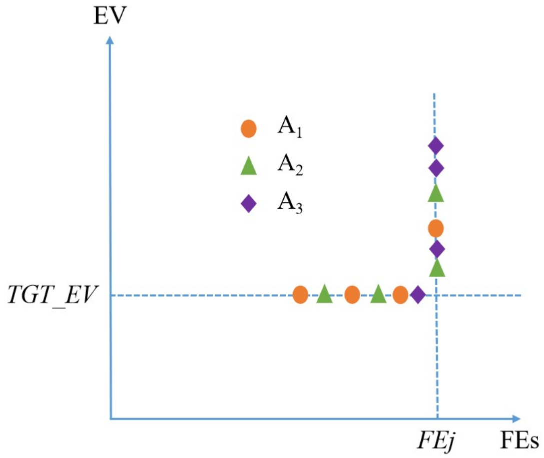

Figure 1: Three algorithms, A1–A3, run four trials each on an SOP. A single run terminates when it reaches Max_FEs. All trials' results can be ordered from the best to the worst. TGT_EV and FE_j will be determined later.

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 algorithms A1, A2 and A3.

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.

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

Maximum Number of Function Evaluations: Max_FEs = 10000*D, where D is the dimensionality of the optimization problems.

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

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

Sampling Points: The best EV (Error Value) every 10*D evaluations will be recorded for each run. For example, the maximum number of function evaluations Max_FEs is 10000*D, then 1000 EVs should be saved.

Table 1
Results saved in "PaperID_FJ_Min_EV.mat" where J=1,2,3,...29 problems.

	Run 1	Run 2	Run 3	...	Run 25
Min_EV at Initialisation FEs					
Min_EV at 10*D FEs					
Min_EV at 20*D FEs					
...					
...					
Min_EV at Max_FEs					

Target Error Values: The target error value, TGT_EV for each problem, will be determined after the competition. Hence, all algorithms should be executed until the 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) $T_1 = (\sum_{i=1}^{29} t_i^1)/29$, 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 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: The results can be saved in the form of Table 1, where Min_EV 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 the 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.

3. Bound constrained multi-objective optimization problems (MOPs)

U-score approach for unconstrained multi-objective optimization problems: For unconstrained multi-objective optimization problems, we use the Inverted Generational Distance (IGD) value as the performance indicator. To exemplify the U-score ranking method's confluence of convergence speed and accuracy on MOP, an illustrative example is depicted in Figure 3. This figure portrays four ranking algorithms, designated as A1 through A4, each assigned a distinct color. For each algorithm, four distinct runs were executed, yielding a total of 16 trials. These trials

Table 2
U-score ranks for MOEAs

Trial	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	*SR*	U-score[1]
Ranks	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	136	
A1	16		14						8	7							45	**35**
A2		15		13							6	5					39	**29**
A3					12			9					4		2		27	**17**
A4						11	10							3		1	25	**15**

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

are stratified into two categories: 1) those that successfully converged to the TGT_IGD 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.)

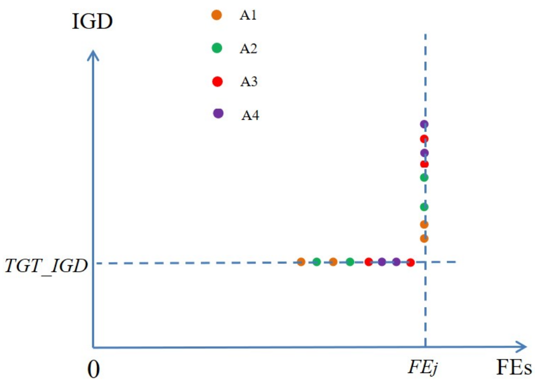

Figure 3: Four algorithms, A1–A4, run four trials each on an MOP. A run terminates when it reaches Max_FEs. TGT_IGD and FE_j will be determined later. All trial results can be ordered from best to worst.

Table 2 presents the tabulated U-score results, wherein the scoring of algorithms A1, A2, A3, and A4 is determined through the summation of their respective rankings. Evidently, algorithm A1 emerges as the victor, amassing a total score of 35 based on the U-score approach, thereby securing the topmost rank. In juxtaposition, A2 secures the second rank, A3 the third, and A4 the fourth. This outcome underscores the supremacy of algorithm A1, attributed to its swifter convergence velocity and its propensity to attain lower *IGD* values.

Test Problems: We adopt the benchmark of [2] including 10 multi-objective problems to rank the optimizers of MOPs without constraints.

Number of Trials/Problem: 30 independent runs.

Table 3
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					

Maximum Number of Function Evaluations: The maximum number of evaluations are set to 100000 for each function.

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 *IGD* values will be recorded once every 200 function evaluations. For example, if the maximum number of evaluations Max_FEs 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 (Max_FEs) 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 3 to the organizers after submitting the final version of the accepted paper.

According to Table 3, 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 3.

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.

[3] 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 .