# Evaluation criteria for CEC 2024 competition and special session on numerical optimization considering accuracy and speed

Kangjia Qiao [a], Xupeng Wen [b], Xuanxuan Ban [a], Peng Chen [a], Kenneth V. Price [e], Ponnuthurai N. Suganthan [d], Jing Liang [a,c], Guohua Wu [b], Caitong Yue [a]

*a*. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, 450001, China, qiaokangjia@gs.zzu.edu.cn, xxuanban@163.com, ty1220899231@163.com

*b*. School of Traffic & Transportation Engineering, Central South University, Changsha, 410073, China, wenxupeng@csu.edu.cn, guohuawu@csu.edu.cn

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

*d*. KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Box:2713, Qatar, p.n.suganthan@qu.edu.qa

*e*. Vacaville, CA, USA, pricekenneth459@gmail.com

Technical Report

November, 2023

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 FEs.

One submission is expected to address only one of the four cases. All results should be saved using a high precision. If you have any query, you can send an email.

## 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, entails the meticulous recording 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, involves recording 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.

The incorporation of both convergence accuracy and speed within the U-score approach [3] engenders a comprehensive perspective on algorithmic performance, thereby facilitating effective comparative analyses and rankings across a multitude of algorithms while considering each run of each algorithms. 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) Bound constrained single objective optimization problems. 2) Constrained single objective optimization problems. 3) Bound constrained multi-objective optimization problems. 4) Constrained multi-objective optimization problems.

## 2. Bound constrained single objective optimization problems (SOP)

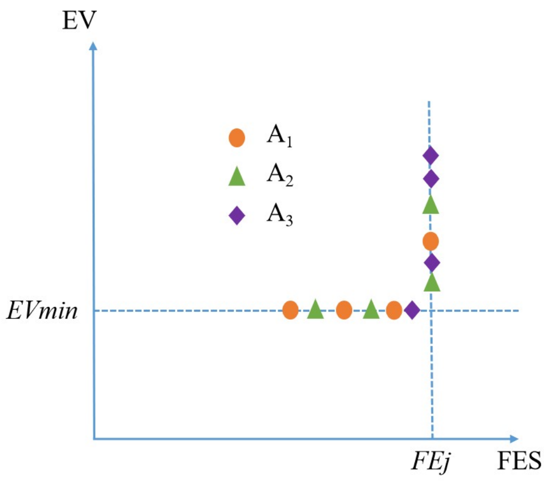


**Figure 1:** Three algorithms, A1–A3, run four trials each on an SOP. A single run terminates when it reaches Max_FEs. $TGT\_EV$ and $FE_j$ will be determined later. All trials' results can be ordered from the best to the worst.

**Test Problems:** The 29 real-parameter numerical optimization problems with $30D$ in CEC2017 [1] are adopted as test problems, and the codes can be downloaded from the website: https://github.com/P-N-Suganthan/CEC2017-BoundContrained.

| 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.

**Number of Trials/Problem:** 25 independent runs. (Do not run many 25 runs to pick the best run).
**Maximum 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 value around every 10*$D$ evaluations will be recorded for each run. For example, the maximum function evaluation Max_FEs is 10000*$D$, then 1000 EV values should be saved.
**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 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) $T_2 = (\sum_{i=1}^{29} t_2^1)/29$, $t_i^2$ 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 $(T_2 - T_1)/T_1$

**Recording 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.
**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, the 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

**Table 1**
Results saved in "Name_S01_Min_EV.mat"

| | Run 1 | Run 2 | Run 3 | ... | Run 25 |
|---|---|---|---|---|---|
| Min_EV at 10*D FEs | | | | | |
| Min_EV at 20*D FEs | | | | | |
| Min_EV at 30*D FEs | | | | | |
| ... | | | | | |
| ... | | | | | |
| Min_EV at generation Max_gens | | | | | |

to the *TGT_EV*, target error value, and 2) those that, failing to attain this target within the stipulated Max_FEs, concluded upon reaching the limit of Max_FEs.

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 diminished *Min_EV* values.

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 single objective optimization problems

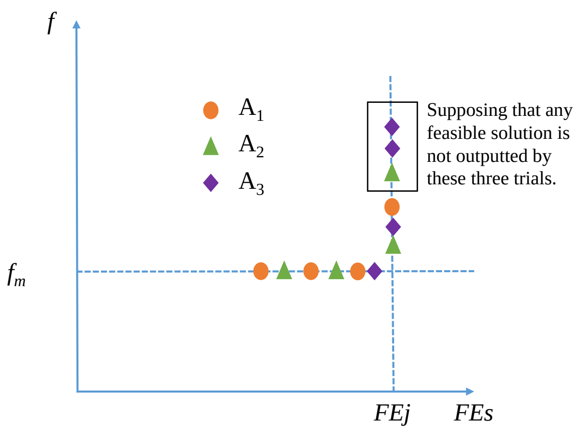


**Figure 3:** Three algorithms, A1–A3, run four trials each on a COP. A single run terminates when it reaches Max_FEs. $TGT\_f$ and $FE_j$ will be determined later. All trial results can be ordered from the best to the worst.

| 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.

**U-score approach for constrained single objective optimization problems:** For constrained single objective optimization problems (COPs), the minimum objective function value $f_{min}$ as the performance indicator at each sampling point, if the corresponding solution is feasible. Feasibility condition is defined in [5]. To exemplify the U-score ranking method's confluence of convergence speed and accuracy on COPs, the 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_f* target value, and 2) those that, failing to attain this target within the stipulated Max_FEs.

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 diminished $f$ value.

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 trials do not output any feasible solution (as such in Figure 4), we will rank these trials based on the lowest value of overall constraint violation ($LCV$) value of the 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 constraint violation value of the $i$-(th) individual in the population $P$. Please note that When recording the $LCV$ value in Table 2, $CV$ 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 [5] are adopted as test problems, and the code can be found from the PlatEMO (website: https://github.com/BIMK/PlatEMO or https://github.com/P-N-Suganthan/CEC2017).

**Number of Trials/Problem:** 25 independent runs.

**Maximum Function Evaluation:** 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:** The $f$ values and $LCV$ around every $10*D$ evaluation will be recorded. For example, if the maximum function evaluation 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 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) $T_2 = (\sum_{i=1}^{28} t_2^1)/28$, $t_i^2$ 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:** The operation results can be saved as Table 2, in which the $Initial_N P$ indicates the initialize population size and the $N_k$ denotes the offspring size at the $k$-th generation. So, the cumulative FEs at each generation should be saved in the first column. Meanwhile, the corresponding $LCV$ and $f_{min}$ 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 exist in one generation, the $f_{min}$ result should be expressed by "NaN".

Thus, for each algorithm, $25 \times 28 = 700$ files should be zipped and sent to the organizers, where 28 represents the total number of test functions, and 25 represents the number of trials.

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.

**Table 2**
Results saved in "PaperID_C01_run1.mat"

| | FEs | $LCV$ | $f_{min}$ |
|---|---|---|---|
| FEs at $Initial_{NP}$ | | | |
| FEs at $\sum_{k=1}^{2} N_k$ | | | |
| FEs at $\sum_{k=1}^{3} N_k$ | | | |
| ... | | | |
| ... | | | |
| FEs at $\sum_{k=1}^{Max\_gens} N_k$ | | | |

## 4. Bound constrained multi-objective optimization problems

**U-score approach for unconstrained multi-objective optimization problems:** For unconstrained multi-objective optimization problems (MOP), 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, the illustrative example is depicted in Figure 5. 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 are stratified into two categories: 1) those that successfully converged to the *TGT_IGD* target value, and 2) those that, failing to attain this target within the stipulated Max_FEs, concluded upon reaching the limit of Max_FEs.

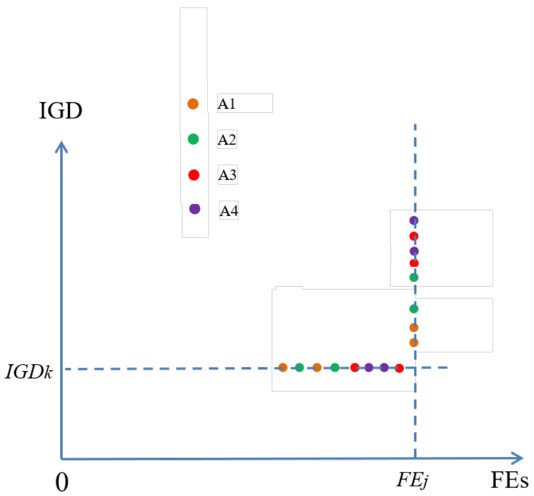


**Figure 5:** Four algorithms, A1–A4, run four trials each on MOP. A run terminates when it reaches Max_FEs. $TGT\_IGD$ and $FE_j$ will be determined later. All trial results can be ordered from best to worst.

**Table 3**
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.

Table 3 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 diminished *IGD* values.

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

**Number of Trials/Problem:** 30 independent runs.

**Maximum Function Evaluation:** The maximum evaluations are set to 100000 for each function.

**Population Size:** The population size is set to 100. If the population size is changed during the running process, please compute the final *IGD* results using maximal 100 feasible individuals.

**Sampling Points:** The *IGD* values will be recorded once every 100 function evaluations. For example, if the maximum number of evaluations *Max_FEs* is 100000, 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 Maximum 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) $T_2 = (\sum_{i=1}^{10} t_2^1)/10$, $t_i^2$ 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 $(T_2 - T_1)/T_1$

**Table 4**
Results saved in"PaperID RCM06 IGD.txt"

| | Run 1 | Run 2 | Run 3 | ... | Run 30 |
|---|---|---|---|---|---|
| *IGD* at FEs 1*$N_1$ | | | | | |
| *IGD* at FEs 2*$N_2$ | | | | | |
| *IGD* at FEs 3*$N_3$ | | | | | |
| ... | | | | | |
| ... | | | | | |
| *IGD* at FEs Max_gens*$N_{Max\_gens}$ | | | | | |

**Presentation of Results:** To compare and evaluate the algorithms participating in the competition, it is necessary that the authors send (through email) the results in the following format to the organizers.

From Table 4, the $N_k$ denotes the $k$-th generation's population size, suppose the maximum number of generations *Max_gen* is 500, then the table contain a (500 × 30) matrix, where 500 represents the number of *IGD* values (Generations), and 30 represents the total runs. For example, the results of PaperID for problem RCM06 at run 3, the files name should be "PaperID RCM06 03 IGD.txt", where Inverted Generational Distance values are saved, respectively.

Thus, for each algorithm, $1 + 50 = 51$ files should be zipped and sent to the organizers, where 50 represents the total number of test functions, and 1 represents the total results of the algorithm obtained containing $10 \times 30 \times 500$, saved as "PaperID.mat".

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 operation results can be saved as Table 4.

## 5. Constrained multi-objective optimization problems

**U-score approach for constrained multi-objective optimization problems:** For constrained multi-objective optimization problems (CMOPs), we introduce the Inverted Generational Distance (*IGD*) values as indicators. To exemplify the U-score ranking method's confluence of convergence speed and accuracy on CMOPs, the illustrative example is depicted in Figure 6. 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, failing to attain this target within the stipulated Max_FEs, concluded upon reaching the limit of Max_FEs.

In adherence to the stipulated procedures and criteria, the specific rankings of algorithms A1 to A3 are delineated within Figure 7, 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 diminished *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 6, 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:

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

where $NP$ is the population 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 5, *CV* should be directly calculated through the constraint functions, rather than the author's normalized *CV* result.

**Test Problems:** The latest constrained multiobjective optimization problems with scalable decision space constraints (SDC problems) [4] are adopted as test problems. SDC benchmark contains 15 problems, and the codes can be downloaded from the website[1].

**Number of Trials:** 30 independent runs.

**Maximum Function Evaluation:** The maximum evaluations are set to 200000 for each function.

[1] 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.

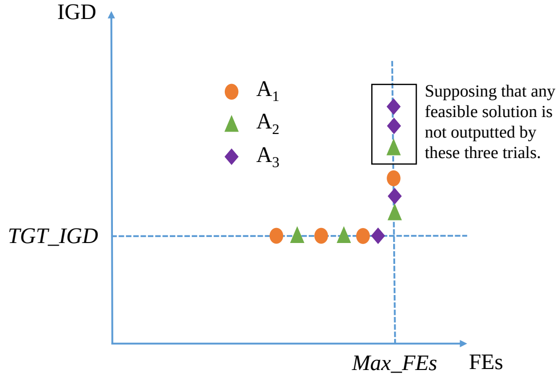


**Figure 6:** Three algorithms, A1–A3, run four trials each on a CMOP. A single run terminates when reaches Max_FEs. $TGT\_IGD$ and $FE_j$ will be determined later. All trial results can be ordered from the best to the worst.

| 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 7:** U-score ranks for CMOPs.

**Population Size:** The population size *NP* is set to 100. If the population size is changed during the running process, please compute the final *IGD* results using maximal 100 feasible individuals.

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

**Sampling Points:** The *IGD* values will be recorded after generating offspring and implementing environmental selection in each generation.

**Target IGD Values:** The target IGD value will be determined after the competition. Hence, all algorithms should be executed until Maximum Function Evaluations (Max_FEs) are consumed. Please note that the minimal IGD value is unknown for multiobjective optimization problems. So, the average 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) $T_2 = (\sum_{i=1}^{15} t_2^1)/15$, $t_i^2$ is the complete computing time for the algorithm with 10000 evaluations for problem $i$.

**Table 5**
Results saved in "PaperID_SDC1_run1.mat"

| C_FEs | MCV | IGD |
|---|---|---|
| $IN$ | | |
| $\sum_{k=1}^{1} N_k$ | | |
| $\sum_{k=1}^{2} N_k$ | | |
| | | |
| | | |
| $\sum_{k=1}^{Max\_gens} N_k$ | | |

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 send (through email) the results in the following format to the organizers.

In Table 5, the $IN$ indicates the initialize population size and the $N_k$ denotes the offspring size at the $k$-th generation (i.e., the used FEs at the $k$-th generation, and C_FEs indicates the cumulative FEs). So, the cumulative FEs at each generation should be saved in the first column. Meanwhile, the corresponding *MCV* and *IGD* 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 exist in one generation, the *IGD* result should be expressed by "NaN".

Thus, for each algorithm, $30 \times 15 = 450$ files should be zipped using the paper number as the file names and sent to the organizers, where 15 represents the total number of test functions, and 30 represents the number of trials.

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

## 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, April .

[4] 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.

[5] 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 .