Guidelines for Real-World Multi-Objective Constrained Optimisation Competition

Abhishek Kumar[a], Guohua Wu[b], Mostafa Z. Ali[c], Qizhang Luo[b], Rammohan Mallipeddi[d,*], Ponnuthurai Nagaratnam Suganthan[e], Swagatam Das[f]

[a]Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, Varanasi, 221005, India.
[b]School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China.
[c]School of Computer Information Systems, Jordan University of Science & Technology, Jordan 22110.
[d]School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea.
[e]School of Electrical Electronic Engineering, Nanyang Technological University, Singapore 639798.
[f]Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India.

For this competition, we develop a set of 50 real-world multi-objective constrained problems [1] with different dimensions and number of objectives to vary from 2 to 34 and 2 to 5. The developed problems contain a wide variety of constraints. In [1], a brief description and baseline results of these problems are reported. This additional attachment provides the basic guidelines of the experimental setting and presentation of results (for manuscript and competition) for the participants. In addition, performance measures used to evaluate the performance of algorithms are also provided in this attachment.

1. Experimental Setting

Number of Trials/problem: 25 independent trials.
Maximum Function Evaluation:

$$Max_{FEs} = \begin{cases} 2 \times 10^4, & \text{if } (M == 2)\ \&\ (D \leq 10) \\ 8 \times 10^4, & \text{elseif } (M == 2)\ \&\ (D > 10) \\ 2.6250 \times 10^4, & \text{elseif } (M == 3)\ \&\ (D \leq 10) \\ 1.05 \times 10^5, & \text{elseif } (M == 3)\ \&\ (D > 10) \\ 3.575 \times 10^4, & \text{elseif } (M == 4)\ \&\ (D \leq 10) \\ 1.43 \times 10^5, & \text{elseif } (M == 4)\ \&\ (D > 10) \\ 5.3 \times 10^4, & \text{elseif } (M == 5)\ \&\ (D \leq 10) \\ 2.12 \times 10^5, & \text{elseif } (M == 5)\ \&\ (D > 10) \end{cases} \tag{1}$$

where Max_{FEs} is maximum allowed function evaluations, M is the number of objectives of the problem and D is dimension (number of decision variables) of problem.
Population Size: You are free to have an appropriate population size to suit your algorithm. However, size of the final population/ archive must be the less than the maximum limit, otherwise, a compulsory truncation will be applied in final statistics for fair comparisons. The maximum limit of population size is max{100, 30M}. while not exceeding the Max_{FEs}.
Search Range: Search range for each problems are provided in *Source Code*.
Initialization: Uniform random initialization within the search range. Random seed is based on time, Matlab users

*Corresponding author
Email address: `mallipeddi.ram@gmail.com` (Rammohan Mallipeddi)

can use rand ('state', sum(100*clock)).
Parameter Setting: We discourage participants searching for a distinct set of parameters for each problem/dimension/etc. Adaptive or dynamic parameter settings independent of each problem number are allowed. Please provide details on the following whenever applicable.

1. All parameters to be adjusted.
2. Corresponding dynamic ranges.
3. Guidelines on how to adjust the parameters.
4. Estimated cost of parameter tuning in terms of number of function evaluations.
5. Actual parameter values used.

Algorithm Complexity: Following procedures are suggested to calculate the algorithmic complexity.

1. $T_1 = \frac{\sum_{i=1}^{50} t_{1i}}{57}$, where t_{1i} is the computation time required to evaluate function for 100000 times for problem i.
2. $T_2 = \frac{\sum_{i=1}^{50} t_{2i}}{57}$, where t_{2i} is the computation time required by algorithm for 100000 function evaluations for problem i.
3. The algorithmic complexity is evaluated using T_1, T_2, and $\frac{T_2 - T_1}{T_1}$.

PC configuration in terms of CPU, OS, RAM, Environment (MATLAB, PYTHON, etc), and Algorithm's name need to mention before algorithm complexity.
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.
Objective Function: Objective functions of benchmark problems are treated as blackbox problems. The explicit equation of objective function (provided in [1]) must not be used.
Constraints: Constraints can be treated as white-box. Explicit constraint equations (provided in [1]) can be used. Authors can manipulate the constraint equations. But, evaluations of a constraint equation or its derivatives must also be counted as one function evaluation.

2. Performance Indicator

In general, performance indicators are used to assess the quality of the obtained Pareto fronts in case of Constrained Multi-objective Optimization Problems (CMOPs). Here, we utilize the Hypervolume Indicator (HV) for giving a score to the Pareto fronts obtained by all algorithms as HV has been the only Pareto-compliant indicator available currently in the literature [2]. A larger value of HV of a given Pareto front indicates the better approximation of the original Pareto front of the given problem. Usually, Pareto front of the real-world problem is not known. This is the main reason for not utilizing the other performance indicator which requires a set of reference vectors. As suggested in [3, 4], we set the reference vector of length M to $[1.1, 1.1, ..., 1.1]^T$ for the calculation of HV in the normalized objective-space. For normalization of objective-space, we use approximated ideal and nadir points of actual objective-space and the normalized i-th objective function value, $f_i(\bar{x})$, for a solution $\bar{x}$ can be obtained by the following equation.

$$\hat{f}_i(\bar{x}) = \frac{f_i(\bar{x}) - f_i^{ideal}}{f_i^{nadir} - f_i^{ideal}} \tag{2}$$

where $\hat{f}_i(\bar{x})$ is the normalized i-th objective function value at solution $barx$; f_i^{ideal} and f_i^{nadir} are the ideal and nadir points of i-th dimension of the original objective-space. respectively. Here, we use two algorithms, SASS [5] and sCMAgES [6] to calculate the ideal and nadir points of all objectives of all problems of the proposed test-suite as these algorithms are the top-ranked algorithms of *Special Session & Competition on Real-world Constrained Optimization* organised at WCCI 2020 and GECCO 2020 [7].

3. Preparation of Statistics

Record the HV value for the achieved Pareto solutions and achieved minimum constraint violation v_m after $0.1Max_{FEs}$, $0.2Max_{FEs}$, $0.3Max_{FEs}$,..., $0.9Max_{FEs}$ and Max_{FEs} function evaluations for each problems. To calculate the v_m for solution x, following equation must be used.

$$v_m = min\{v(x_i)\},\ i = 1, 2...N_p \tag{3}$$

where,

$$v(x) = \frac{\sum_{i=1}^{p} G_i(x) + \sum_{j=p+1}^{m} H_j(x)}{m}, \tag{4}$$

$$G_i(x) = \begin{cases} g_i(x), & \text{if } g_i(x) > 0 \\ 0, & \text{if } g_i(x) \leq 0. \end{cases},$$

and

$$H_j(x) = \begin{cases} |h_j(x)|, & \text{if } |h_j(x)| - 0.0001 > 0 \\ 0, & \text{if } |h_j(x)| - 0.0001 \leq 0. \end{cases}$$

Moreover, record the Pareto front and solutions of each run. Calculate Feasibility Rate (FR) for each problems over 25 trails using following procedures.

$$\text{FR} = \frac{\text{Total trials having } v_m = 0}{\text{Total trials}}, \tag{5}$$

4. Presentation of Results

The simulation results obtained for the different optimization problems should be reported in the specified formats (in the manuscript and for the competition).

4.1. Presentation of results in the conference manuscript

For each problem, the results need to be presented in the following format in the manuscript.

Table 1: Outcomes at FEs = Max_{FEs} for Problems RCM01-RCM08.

		RCM01	RCM02	RCM03	RCM04	RCM05	RCM06	RCM07	RCM08
Best	hv								
	v_m								
Median	hv								
	v_m								
Mean	hv								
	v_m								
Worst	hv								
	v_m								
Std	hv								
	v_m								
FR									

∗The sorting method:

1. Sort runs having $v_m = 0$ in front of runs having $v_m > 0$;
2. Sort runs having $v_m = 0$ according to their HV values hv;
3. Sort runs having $v_m > 0$ according to their v_m.

4.2. Presentation of results for the competition

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.
Create four txt document with the name "AlgorithmName_FunctionNo._HV.txt", "AlgorithmName_FunctionNo._CV.txt", "AlgorithmName_FunctionNo._TrailNo_PS.txt" and "AlgorithmName_FunctionNo._TrailNo_PF.txt" for each problem. For example, PSO results for problem RCM05, the files name should be "PSO_RCM05_HV.txt" and "PSO_RC05_CV.txt". Then save the results matrix (the blocking part) as Table 2 and Table 3 in the file.

Table 2: Information matrix saved in"PSO_RCM05_F.txt"

	Trial 1	Trial 2	::	Trial 25
HV values at $FEs = 0.1 * Max_{FES}$				
HV values at $FEs = 0.2 * Max_{FES}$				
HV values at $FEs = 0.3 * Max_{FES}$				
::				
HV values at $FEs = 0.9 * Max_{FES}$				
HV values at $FEs = 1.0 * Max_{FES}$				

Table 3: Information matrix saved in"PSO_RCM05_CV.txt"

	Trial 1	Trial 2	::	Trial 25
v_m at $FEs = 0.1 * Max_{FES}$				
v_m at $FEs = 0.2 * Max_{FES}$				
v_m at $FEs = 0.3 * Max_{FES}$				
::				
v_m at $FEs = 0.9 * Max_{FES}$				
v_m at $FEs = 1.0 * Max_{FES}$				

These file contain a (10×25) matrix. The next two files for PSO results over problem RCM05 at trial 3, the files name should be "PSO_RCM05_03_PF.txt" and "PSO_RC05_03_PS.txt", where Pareto front and Pareto solutions are saved, respectively. Thus $50 * 2 + 50 * 2 * 25 = 2600$ files should be zipped and sent to the organizers.
Notice: All participants are allowed to improve their algorithms further after submitting the initial version of their papers submitted to conference. And they are required to submit their results in the introduced format to the organizers after submitting the **final** version of paper as soon as possible.

5. Performance Measure for algorithms

For ranking the Constrained Multi-objective Optimization Evolutionary Algorithms (CMOEAs) based on the performance over the proposed benchmark suite, we propose a ranking scheme inspired from [8]. Supposing N algorithms CMOEA_1, CMOEA_2.....CMOEA_N participates in the comparative analysis done on P problems. The performance score, S, can be defined as follows:

$$S(CMOEA_i) = \frac{1}{P}\left(\sum_{j=1}^{P} \frac{1}{N-1}\left(0.5\sum_{k=1}^{N} \delta_{j,k}^{i,best} + 0.3\sum_{k=1}^{N} \delta_{j,k}^{i,mean} + 0.2\sum_{k=1}^{N} \delta_{j,k}^{i,median}\right)\right), \tag{6}$$

where,

$$\delta_{j,k}^{i,mean} = \begin{cases} 1, & \text{if } CMOEA_k \text{ significantly outperforms } CMOEA_i \text{ on a problem } k, \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

and

$$\delta_{j,k}^{i,l} = \begin{cases} 1, & \text{if } \Big((hv_j > hv_k) \wedge (v_{m,j} == v_{m,k})\Big) \vee \Big(v_{m,j} < v_{m,k}\Big), \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Here, Wilcoxon rank-sum test at a 0.05 significance level is utilized to determine the significant difference between the performance of two algorithms on a problem for $\delta_{j,k}^{i,mean}$. The lower value of S of an algorithm suggests that the algorithm performs better on the proposed test-suite.

6. Details of Real-World Problems

Table 4: Details of the 50 RWCMOPs. M is the total number of objectives, D is the total number of decision variables of the problem, ng is the number of inequality constraints and nh is the number of equality constraints

Prob	Name	M	D	ng	nh
	Mechanical Design Problems				
RCM01	Pressure Vessel Design	2	2	2	2
RCM02	Vibrating Platform Design	2	5	5	0
RCM03	Two Bar Truss Design	2	3	3	0
RCM04	Welded Beam Design	2	4	4	0
RCM05	Disc Brake Design	2	4	4	0
RCM06	Speed Reducer Design	2	7	11	0
RCM07	Gear Train Design	2	4	1	0
RCM08	Car Side Impact Design	3	7	9	0
RCM09	Four Bar Plane Truss	2	4	1	0
RCM10	Two Bar Plane Truss	2	2	2	0
RCM11	Water Resources Management	5	3	7	0
RCM12	Simply Supported I-beam Design	2	4	1	0
RCM13	Gear Box Design	3	7	11	0
RCM14	Multiple Disk Clutch Brake Design	2	5	8	0
RCM15	Spring Design	2	3	8	0
RCM16	Cantilever Beam Design	2	2	2	0
RCM17	Bulk Carrier Design	3	6	9	0
RCM18	Front Rail Design	2	3	3	0
RCM19	Multi-product Batch Plant	3	10	10	0
RCM20	Hydro-static Thrust Bearing Design	2	4	7	0
RCM21	Crash Energy Management for High-speed Train	2	6	4	0
	Chemical Engineering Problems				
RCM22	Haverly's Pooling Problem	2	9	2	4
RCM23	Reactor Network Design	2	6	1	4
RCM24	Heat Exchanger Nwteork Design	3	9	0	6
	Process, Design and Synthesis Problems				
RCM25	Process Synthesis Problem	2	2	2	0
RCM26	Process Synthesis and Design Problem	2	3	1	1
RCM27	Process Flow Sheeting Problem	2	3	3	0
RCM28	Two Reactor Problem	2	7	4	4
RCM29	Process Synthesis Problem	2	7	9	0
	Power Electronics Problems				
RCM30	Synchronous Optimal Pulse-width Modulation of 3-level Inverters	2	25	24	0
RCM31	Synchronous Optimal Pulse-width Modulation of 5-level Inverters	2	25	24	0
RCM32	Synchronous Optimal Pulse-width Modulation of 7-level Inverters	2	25	24	0
RCM33	Synchronous Optimal Pulse-width Modulation of 9-level Inverters	2	30	29	0
RCM34	Synchronous Optimal Pulse-width Modulation of 11-level Inverters	2	30	29	0
RCM35	Synchronous Optimal Pulse-width Modulation of 13-level Inverters	2	30	29	0
	Power System Optimization Problems				
RCM36	Optimal Sizing of Single Phase Distribution Generation with Reactive Power support for phase balancing at Main Transformer/Grid and Minimizing Active Power Loss	2	28	0	24
RCM37	Optimal Sizing of Single Phase Distribution Generation with Reactive Power support for phase balancing at Main Transformer/Grid and Minimizing Reactive Power Loss	2	28	0	24
RCM38	Optimal Sizing of Single Phase Distribution Generation with Reactive Power support for Minimizing Active and Reactive Power Loss	2	28	0	24
RCM39	Optimal Sizing of Single Phase Distribution Generation with Reactive Power support for phase balancing at Main Transformer/Grid and Minimizing Active and Reactive Power Loss	3	28	0	24
RCM40	Optimal Power Flow for Minimizing Active and Reactive Power Loss	2	34	0	26
RCM41	Optimal Power Flow for Minimizing Voltage deviation, Active and Reactive Power Loss	3	34	0	26
RCM42	Optimal Power Flow for Minimizing Voltage deviation, and Active Power Loss	2	34	0	26
RCM43	Optimal Power Flow for Minimizing Fuel Cost, and Active Power Loss	2	34	0	26
RCM44	Optimal Power Flow for Minimizing Fuel Cost, Active and Reactive Power Loss	3	34	0	26
RCM45	Optimal Power Flow for Minimizing Fuel Cost, Voltage deviation, and Active Power Loss	3	34	0	26
RCM46	Optimal Power Flow for Minimizing Fuel Cost, Voltage deviation, Active and Reactive Power Loss	4	34	0	26
RCM47	Optimal Droop Setting for Minimizing Active and Reactive Power Loss	2	18	0	12
RCM48	Optimal Droop Setting for Minimizing Voltage Deviation and Active Power Loss	2	18	0	12
RCM49	Optimal Droop Setting for Minimizing Voltage Deviation, Active, and Reactive Power Loss	2	18	0	12
RCM50	Power Distribution System Planning	2	6	0	1

References

[1] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, S. Das, A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results, in: *submitted to*, Swarm and Evolutionary Computation, Elsevier, 2020.

[2] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 862–876.

[3] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (2015) 16–37.

[4] H. Ishibuchi, Y. Setoguchi, H. Masuda, Y. Nojima, Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes, IEEE Transactions on Evolutionary Computation 21 (2016) 169–190.
[5] A. Kumar, S. Das, I. Zelinka, A self-adaptive spherical search algorithm for real-world constrained optimization problems, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, 2020, pp. 13–14.
[6] A. Kumar, S. Das, I. Zelinka, A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, 2020, pp. 11–12.
[7] A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, Swarm and Evolutionary Computation (2020) 100693.
[8] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary computation 19 (2011) 45–76.