Problem Definitions and Evaluation Criteria for the CEC 2021 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization

Ali Wagdy Mohamed[1,2], Anas A Hadi[3], Ali Khater Mohamed[4], Prachi Agrawal[5], Abhishek Kumar[6], P. N. Suganthan[7]

[1]Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt.

[2]Wireless Intelligent Networks Centre (WINC), School of Engineering and Applied Sciences, Nile University, Giza, Egypt.

[3]College of Computing and Information Technology, King Abdul-Aziz University P. O. Box 80200, Jeddah 21589, Saudi Arabia

[4]Faculty of Computer Science, October University for Modern Sciences and Arts (MSA), 6th October city, Giza 12451, Egypt

[5]Department of Mathematics and Scientific Computing, National Institute of Technology Hamirpur, Himachal Pradesh 177005 India

[6]Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, Varanasi, 221005, India

[7]School of Electrical Electronic Engineering, Nanyang Technological University, Singapore

aliwagdy@gmail.com, anas1401@gmail.com , akhater@msa.eun.eg, prachiagrawal202@gmail.com,

abhishek.kumar.eee13@iitbhu.ac.in, epnsugan@ntu.edu.sg

Technical Report

November 2020

Single objective optimization algorithms are the foundation upon which more complex methods, like multi-objective, niching and constrained optimization algorithms, are built. Consequently, improvements to single objective optimization algorithms are important because they can impact other domains as well. These algorithmic improvements depend in part on feedback from trials conducted with single objective benchmark functions, which themselves are the elemental building blocks for more complex tasks, like dynamic, niching, composition and computationally expensive problems. As algorithms improve, ever more challenging functions must be developed. This interplay between methods and problems drives progress, so we have developed the CEC'21 Special Session on Real-Parameter Optimization to promote this symbiosis.

Improved methods and problems sometimes require updating traditional testing criteria. In recent years, many novel optimization algorithms have been proposed to solve the bound-constrained, single objective problems offered in the CEC'05[1], CEC'13[2], CEC'14[3], CEC'17[4] , and CEC'20[5] special sessions on Real-Parameter Optimization. In this competition, the benchmark objective functions are parameterized by including the operators such as bias, rotation, and translation. The main motive behind the parameterization is to test the effect of all combinations of the operators on all benchmark functions.

Parametrized benchmarking is a step towards obtaining multi-faceted insight into algorithmic performance and the optimization problems[6]. For this, 10 scalable benchmark problems are proposed with these binary operators.

Participants are required to send their final results to the organizers in the format specified in this technical report. Based on these results, organizers will present a comparative analysis that includes statistical tests on convergence performance to compare algorithms with similar final solutions.

Participants may not explicitly use the equations of the test functions, e.g. to compute gradients. This competition also excludes surrogate and meta-models. Papers on novel concepts that help us to understand problem characteristics are also welcome. The C and MATLAB codes for CEC'21 test suite can be downloaded from the website below:

https://github.com/P-N-Suganthan

1. Introduction to the CEC'21 Benchmark Suite

1.1. Some Definitions:

All test functions are minimization problems defined as follows:

$$\text{Min } f(\mathbf{x}), \mathbf{x} = [x_1, x_2, \ldots, x_D]^{\mathrm{T}}$$

D: number of dimensions.

$\boldsymbol{o}_{i1} = [o_{i1}, o_{i2}, \ldots, o_{iD}]^{\mathrm{T}}$: the shifted global optimum (defined in "shift_data_x.txt"), which is randomly distributed in $[-80,80]^D$. All test functions are shifted to $\boldsymbol{o}$ and are scalable.

Search range: $[-100,100]^D$. For convenience, the same search ranges are defined for all test functions.

$\mathbf{M}_i$: rotation matrix. Different rotation matrix are assigned to each function and each basic function.

Considering that linkages seldom exists among all variables in real-world problems, CEC'20 randomly divides variables into subcomponents. The rotation matrix for each set of subcomponents is generated from standard normally distributed entries by Gram-Schmidt ortho-normalization with condition number *c* that is equal to 1 or 2.

1.2. Summary of the CEC'21 Test Suite

	No.	Functions	$F_i^* = F_i(\mathbf{x}^*)$
Unimodal Function	1	Shifted and Rotated Bent Cigar Function (CEC 2017[4] F1)	100
Basic Functions	2	Shifted and Rotated Schwefel's Function (CEC 2014[3] F11)	1100
	3	Shifted and Rotated Lunacek bi-Rastrigin Function (CEC 2017[4] F7)	700
	4	Expanded Rosenbrock's plus Griewangk's Function (CEC2017[4] f_{19})	1900
Hybrid Functions	5	Hybrid Function 1 (N = 3) (CEC 2014[3] F17)	1700
	6	Hybrid Function 2 (N = 4) (CEC 2017[4] F16)	1600
	7	Hybrid Function 3 (N = 5) (CEC 2014[3] F21)	2100
Composition Functions	8	Composition Function 1 (N = 3) (CEC 2017[4] F22)	2200
	9	Composition Function 2 (N = 4) (CEC 2017[4] F24)	2400
	10	Composition Function 3 (N = 5) (CEC 2017[4] F25)	2500
Search range: $[-100,100]^D$			

***Please Note: These problems should be treated as black-box problems. The explicit equations of the problems are not to be used.**

1.3. Definitions of the Basic Functions

1) Bent Cigar Function

$$f_1(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2 \tag{1}$$

2) Rastrigin's Function

$$f_2(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right) \tag{2}$$

3) High Conditioned Elliptic Function

$$f_3(x) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} x_i^2 \tag{3}$$

4) HGBat Function

$$f_4(\mathbf{x}) = \left| \left(\sum_{i=1}^{D} x_i^2 \right)^2 - \left(\sum_{i=1}^{D} x_i \right)^2 \right|^{1/2} + \left(0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i \right) / D + 0.5 \tag{4}$$

5) Rosenbrock's Function

$$f_5(\mathbf{x}) = \sum_{i=1}^{D-1} \left(100\left(x_i^2 - x_{i+1} \right)^2 + \left(x_i - 1 \right)^2 \right) \tag{5}$$

6) Griewank's Function

$$f_6(\mathbf{x}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1 \tag{6}$$

7) Ackley's Function

$$f_7(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e \tag{7}$$

8) Happycat Function

$$f_8(\mathbf{x}) = \left|\sum_{i=1}^{D}x_i^2 - D\right|^{1/4} + \left(0.5\sum_{i=1}^{D}x_i^2 + \sum_{i=1}^{D}x_i\right)/D + 0.5 \tag{8}$$

9) Discus Function

$$f_9(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^{D}x_i^2 \tag{9}$$

10) Lunacek bi-Rastrigin Function

$$f_{10}(\mathbf{x}) = \min\left(\sum_{i=1}^{D}(\hat{x}_i - \mu_0)^2, dD + s\sum_{i=1}^{D}(\hat{x}_i - \mu_1)^2\right) + 10\left(D - \sum_{i=1}^{D}\cos(2\pi\hat{z}_i)\right) \tag{10}$$

$$\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D+20} - 8.2}, d = 1$$

$$y = \frac{10(x-o)}{100}, \frac{x_i}{x_i} = 2\,\mathrm{sign}\left(x_i^*\right)y_i + \mu_0, \text{ for } i = 1,2,\ldots,D$$

$$z = \Lambda^{100}\left(\hat{x} - \mu_0\right)$$

11) Modified Schwefel's Function

$$f_{11}(x) = 418.9829 \times D - \sum_{i=1}^{D}g\left(z_i\right) \tag{11},$$

$$z_i = x_i + 4.209687462275036\mathrm{e}+002$$

$$g(z_i) = \begin{cases} z_i \sin\left(|z_i|^{1/2}\right) & \text{if } |z_i| \le 500 \\ \left(500 - \mathrm{mod}(z_i, 500)\right)\sin(\sqrt{|500 - \mathrm{mod}(z_i, 500)|}) - \frac{(z_i - 500)^2}{10000D} & \text{if } z_i > 500 \\ \left(\mathrm{mod}(|z_i|, 500) - 500\right)\sin(\sqrt{|\mathrm{mod}(|z_i|,|, 500) - 500|}) - \frac{(z_i + 500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$$

12) Expanded Schaffer's Function

$$\text{Schaffer's Function: } g(x,y) = 0.5 + \frac{\left(\sin^2(\sqrt{x^2+y^2}) - 0.5\right)}{\left(1 + 0.001\left(x^2+y^2\right)\right)^2}$$

$$f_{12}(\mathbf{x}) = g(x_1, x_2) + g(x_2, x_3) + \ldots + g(x_{D-1}, x_D) + g(x_D, x_1) \tag{12}$$

13) Expanded Rosenbrock's plus Griewangk's Function

$$f_{13}(\mathbf{x}) = f_6\left(f_5(x_1, x_2)\right) + f_6\left(f_5(x_2, x_3)\right) + \ldots + f_6\left(f_5(x_{D-1}, x_D)\right) + f_6\left(f_5(x_D, x_1)\right) \tag{13}$$

14) Weierstrass Function

$$f_{14}(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\left(x_i+0.5\right)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\cdot 0.5\right)\right] \tag{14}$$

$$a = 0.5, b = 3, k\max = 20$$

1.4. Definitions of the CEC'21 Test Suite

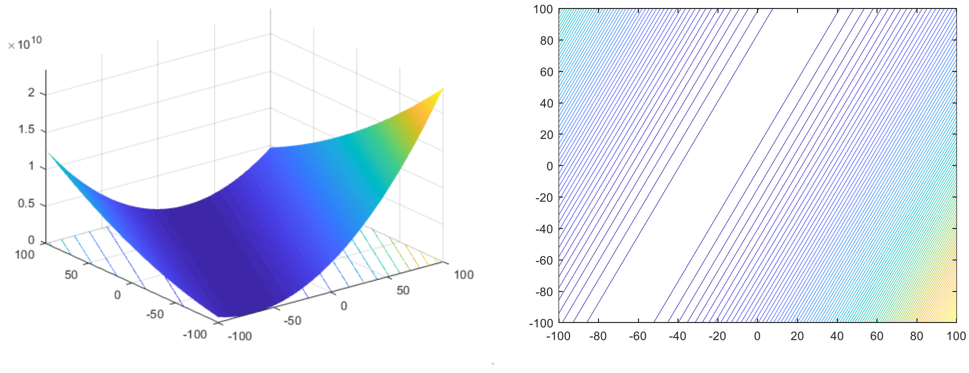
A. Basic Functions

1) Bent Cigar Function

$$F_1(\mathbf{x}) = f_1\left(\mathrm{M}\left(\mathbf{x}-o_1\right)\right)+F_1^* \tag{15}$$

Properties:

- Unimodal
- Non-separable
- Smooth but narrow ridge

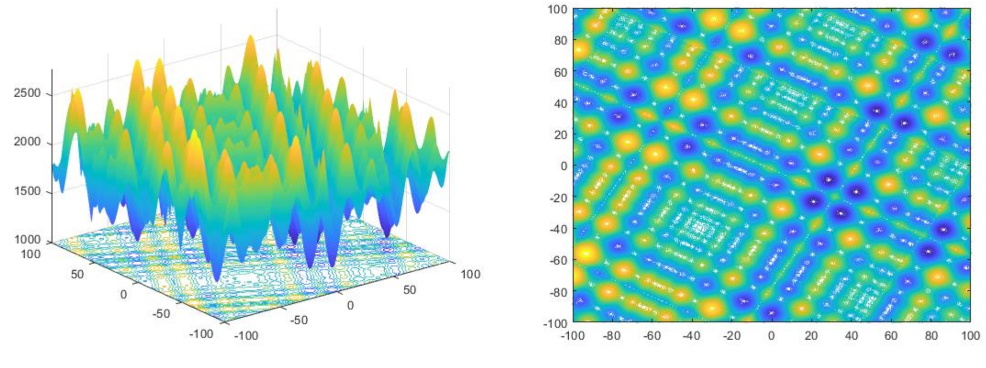


(a) 3-*D* map for 2-*D* function (b) Contour map for 2-*D* function

Figure 1 Bent Cigar Function

2) Shifted and Rotated Schwefel's Function (the same as F11 in CEC2014[3])

$$F_2(\mathbf{x}) = f_{11}\left(\mathbf{M}\left(\frac{1000\left(\mathbf{x}-\boldsymbol{o}_2\right)}{100}\right)\right)+F_2^* \tag{16}$$



(a) 3-*D* map for 2-*D* function (b) Contour map for 2-*D* function

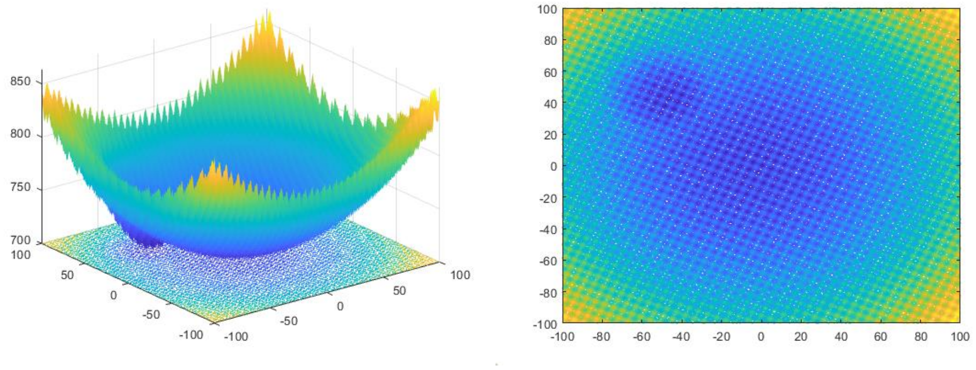
Figure 2 Shifted and Rotated Schwefel's Function

Properties:

- Multi-modal
- Non-separable
- Local optima's number is huge and the penultimate local optimum is far from the global optimum.

3) Shifted and Rotated Lunacek bi-Rastrigin Function (the same as F7 in CEC2017[4])

$$F_3(x) = f_{10}\left(\mathbf{M}\left(\frac{600(x-o_3)}{100}\right)\right) + F_3^* \tag{17}$$



(a) 3-*D* map for 2-*D* function (b) Contour map for 2-*D* function

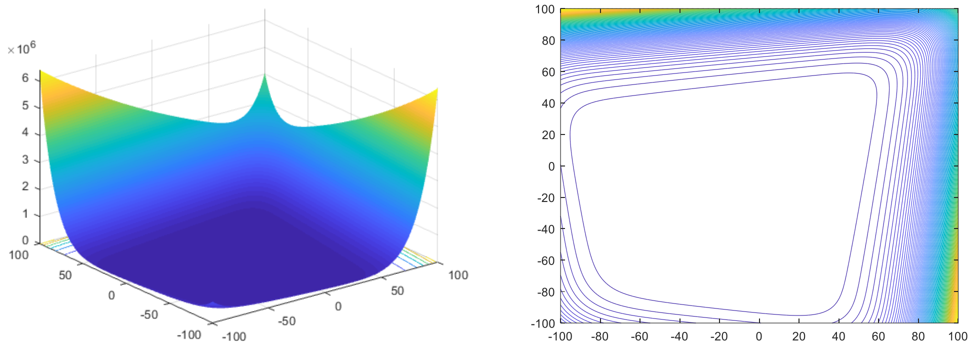
Figure 3 Shifted and Rotated Lunacek bi-Rastrigin Function

Properties:

- Multi-modal
- Non-separable
- Asymmetrical
- Continuous everywhere yet differentiable nowhere

4) Expanded Rosenbrock's plus Griewangk's Function (the same as f_{19} in CEC2017[4])

$$F_4(\mathbf{x}) = f_6\left(f_5\left(x_1, x_2\right)\right) + f_6\left(f_5\left(x_2, x_3\right)\right) + \ldots + f_6\left(f_5\left(x_{D-1}, x_D\right)\right) + f_6\left(f_5\left(x_D, x_1\right)\right) + F_4^* \tag{18}$$



(a) 3-*D* map for 2-*D* function (b) Contour map for 2-*D* function

Figure 4 Expanded Rosenbrock's plus Griewangk's Function

Properties:

- ➢ Non-separable
- ➢ Optimal point locates in flat area

B. Hybrid Functions

Considering that in the real-world optimization problems, different subcomponents of the variables may have different properties[7]. In this set of hybrid functions, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents.

$$F(\mathbf{x}) = g_1\left(\mathbf{M}_1 z_1\right) + g_2\left(\mathbf{M}_2 z_2\right) + \ldots + g_N\left(\mathbf{M}_N z_N\right) + F^*(\mathbf{x}) \tag{19}$$

$F(\mathbf{x})$: hybrid function

$g_i(\mathbf{x})$: i^{th} basic function used to construct the hybrid function

N: number of basic functions

$z = \left[z_1, z_2, \ldots, z_N\right]$

$z_1 = \left[y_{S_1}, y_{S_2}, \ldots, y_{S_{n_1}}\right], z_2 = \left[y_{S_{n_1+1}}, y_{S_{n_1+2}}, \ldots, y_{S_{n_1+n_2}}\right], \ldots, z_N = \left[y_{S_{\sum_{i=1}^{N-1} n_i+1}}, y_{S_{\sum_{i=1}^{N-1} n_i+2}}, \ldots, y_{S_D}\right]$

$y = x - o_i, S = \text{randperm}(1:D)$

p_i: used to control the percentage of $g_i(\mathbf{x})$

n_i: dimension for each basic function $\sum_{i=1}^{N} n_i = D$

$n_1 = \lceil p_1 D \rceil, n_2 = \lceil p_2 D \rceil, \ldots, n_{N-1} = \lceil p_{N-1} D \rceil, n_N = D - \sum_{i=1}^{N-1} n_i$

Properties:

- Multi-modal or Unimodal, depending on the basic function
- Non-separable subcomponents
- Different properties for different variables subcomponents

5) Hybrid Function 1 (the same as F17 in CEC2014[3])

$N = 3$

$p = [0.3, 0.3, 0.4]$

g_1: Modified Schwefel's Function f_{11}

g_2: Rastrigin's Function f_2

g_3: High Conditioned Elliptic Function f_3

6) Hybrid Function 2 (the same as F16 in CEC2017[4])

$N = 4$

$p = [0.2, 0.2, 0.3, 0.3]$

g_1: Expanded Schaffer Function f_{12}

g_2: HGBat Function f_4

g_3: Rosenbrock's Function f_5

g_4: Modified Schwefel's Function f_{11}

7) Hybrid Function 3 (the same as F21 in CEC2014[3])

$N = 5$

$p = [0.1, 0.2, 0.2, 0.2, 0.3]$

g_1: Expanded Schaffer Function f_{12}

g_2: HGBat Function f_4

g_3: Rosenbrock's Function f_5

g_4: Modified Schwefel's Function f_{11}

g_5: High Conditioned Elliptic Function f_3

C. Composition Functions

$$F(\mathbf{x}) = \sum_{i=1}^{N} \left\{ \omega_i^* \left[\lambda_i g_i(\mathbf{x}) + bias_i \right] \right\} + F^* \tag{20}$$

$F(\mathbf{x})$: composition function

$g_i(\mathbf{x})$: i^{th} basic function used to construct the composition function

N: number of basic functions

$\boldsymbol{o}_i$: new shifted optimum position for each $g_i(\mathbf{x})$, define the global and local optima's position

$bias_i$: defines which optimum is global optimum

σ_i : used to control each $g_i(\mathbf{x})$'s coverage range, a small σ_i gives a narrow range for that $g_i(\mathbf{x})$

λ_i : used to control each $g_i(\mathbf{x})$'s height

ω_i : weight value for each $g_i(\mathbf{x})$, calculated as below:

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^{D} \left(x_j - o_{ij} \right)^2}} \exp\left(-\frac{\sum_{j=1}^{D} \left(x_j - o_{ij} \right)^2}{2D\sigma_i^2} \right) \tag{21}$$

Then normalize the weight $\omega_i = w_i / \sum_{i=1}^{n} w_i$

So when $\mathbf{x} = \boldsymbol{o}_i$, $\omega_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$ for $j = 1, 2, \ldots, N$, $f(\mathbf{x}) = bias_i + f^*$.

The local optimum which has the smallest bias value is the global optimum. The composition function merges the properties of the sub-functions better and maintains continuity around the global/local optima.

Functions $Fi' = Fi - F_i^*$ are used as g_i. In this way, the function values of global optima of g_i are equal to 0 for all composition functions in this report.

In CEC'14[3], the hybrid functions are also used as the basic functions for composition functions (Composition Function 7 and Composition Function 8). With hybrid functions as the basic functions, the composition function can have different properties for different variables subcomponents.

Please Note: All the basic functions that have been used in composition functions are shifted and rotated functions.

8) Composition Function 1 (the same as F22 in CEC2017[4])

$N=3$

$\sigma=[10,20,30]$

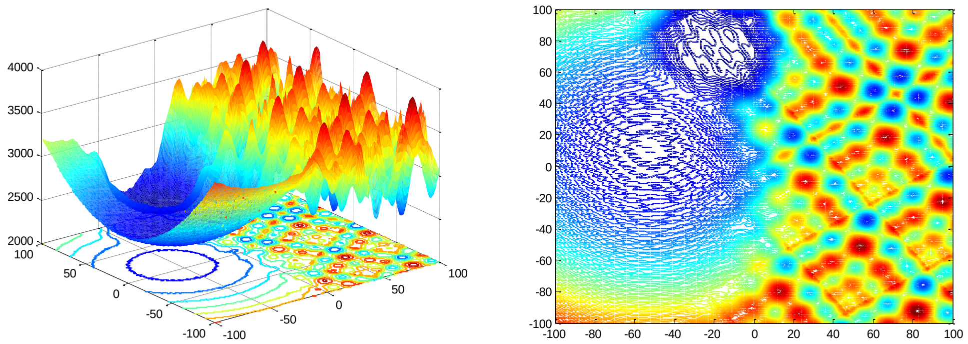
$\lambda=[1,10,1]$

$bias=[0,100,200]$

g_1: Rastrigin's Function f_2

g_2: Griewank's Function f_6

g_3: Modified Schwefel's Function f_{11}



(a) 3-D map for 2-D function (b) Contour map for 2-D function

Figure 5 Composition Function 1

Properties:

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

9) Composition Function 2 (the same as F24 in CEC2017[4])

$N=4$

$\sigma=[10,20,30,40]$

$\lambda=[10,1e-6,10,1]$

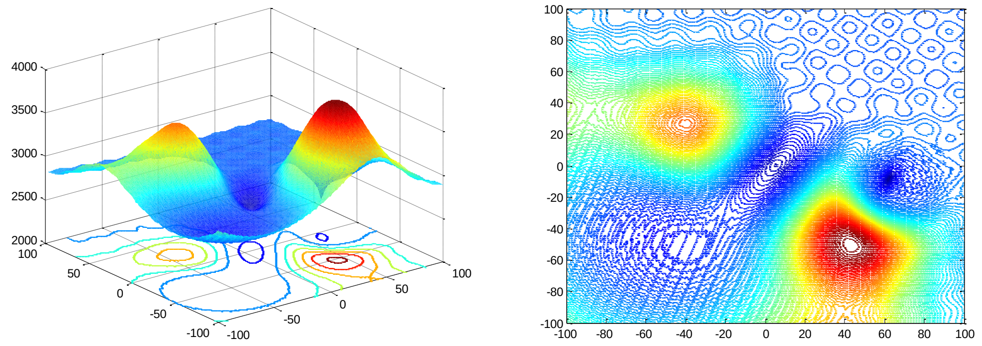
$bias=[0,100,200,300]$

g_1: Ackley's Function f_7

g_2: High Conditioned Elliptic Function f_3

g_3: Griewank's Function f_6

g_4: Rastrigin's Function f_2



(a) 3-*D* map for 2-*D* function (b) Contour map for 2-*D* function

Figure 6 Composition Function 2

10) Composition Function 3 (the same as F25 in CEC2017[4])

N=5

$\sigma = [10, 20, 30, 40, 50]$

$\lambda = [10, 1, 10, 1e-6, 1]$

$bias = [0, 100, 200, 300, 400]$

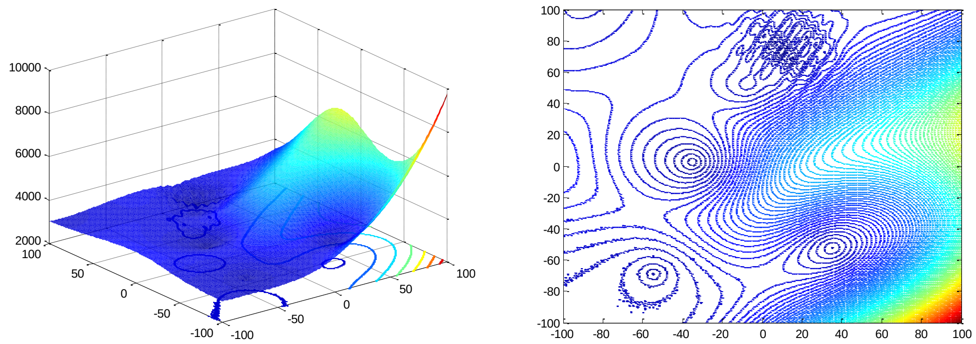
g_1: Rastrigin's Function f_2

g_2: Happycat Function f_8

g_3: Ackley's Function f_7

g_4: Discus Function f_9

g_5: Rosenbrock's Function f_5



(a) 3-*D* map for 2-*D* function (b) Contour map for 2-*D* function

Figure 7 Composition Function 3

Properties:

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

2. Parametrized Benchmark

The benchmarks are vital to the improvement in global metaheuristics. The two-benchmark series CEC[4] and COCO[8] are proposed to evaluate the real parameter metaheuristic algorithms. In this competition, the benchmark functions are considered by applying the different transformations such as bias, rotation and shift[6]. The all possible combinations of operators such as bias (exist or does not exist), rotation (exists and does not exist), shift (exists and does not exist), bias (exists) but rotation (does not exist) and so on are proposed. The resulting set is known as parameterized benchmark such that the main goal is to be achieved. The main goal is to test the effect of all possible combinations of transformations in the benchmark functions as the best transformation can be chosen.

In CEC'17[4] benchmark, 20 basic functions are defined with the inclusion of shift-vector o_i in the variables x, multiplication of rotation matrix M, and addition of bias F_i^* in the original objective function f_i. The mathematical equation for the test function is created as:

$$F_i(X) = f_i\big(M(x - o_i)\big) + F_i^* \tag{22}$$

Matrix M is used for the rotation transformation, which contains two possible cases as it does exist, and it does not exist.

There are some detailed variations for hybrid and composition functions, which make the full pattern slightly more complicated. The $F_i(X)$ is known parameterized benchmark function in which we have to test the effect of shift, rotation and bias transformation on the original benchmark functions. Therefore, the decomposition allows to define the binary parameters that demonstrate which transformation should be applied and ensures that predictors are standardized to the same scale. The values taken by the parameters is presented in Table 1 with the reference Equation number from which the value can be obtained.

Table 1: Parametrization of the benchmark problems

ID	Parameter	Values	Reference	Type
C_1	Bias	F_i^* or 0	Equation (22)	Controlled
C_2	Shift	o_i or 0	Equation (22)	Controlled
C_3	Rotation	M or I	Equation (22)	Controlled

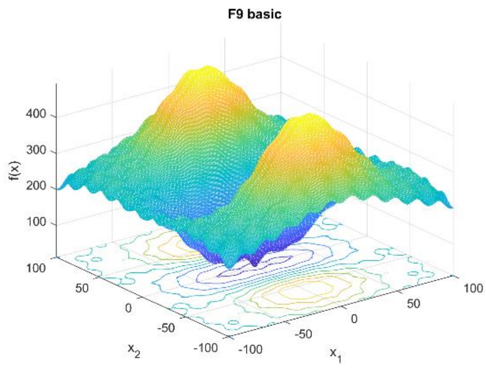
The parameters (bias, shift and rotation) can be controlled, and they can be activated or deactivated. While, some other uncontrollable parameters or features can be observed only. These parameters are problem type, separability, number of local optima and symmetry. The problem types are unimodal, simple multimodal, hybrid, and composition. Besides, the problem may be separable or non-separable. There is a few or a huge number of local optima. The shape of the problem may be symmetric or asymmetric. These parameters are observed only as their values are fixed.

There are only 8 configurations possible for each function. Therefore, to understand the binary parameters one example is illustrated as, if we want to check the effect of shift operator not the rotation and bias operator then we would set M=I (identity matrix) and $F_i^* = 0$. The detailed description of each binary operator applied on function F_i has been shown in Table 2.

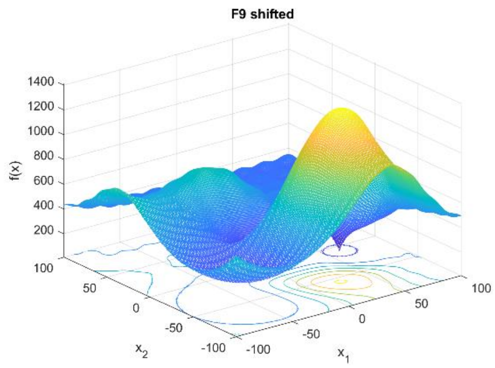
Table 2: Binary parameter values for each transformation applied on the function F_i.

Name of the Functions	Bias	Shift	Rotation
F_i Basic	0	0	I
F_i Bias	F_i^*	0	I
F_i Shift	0	o_i	I
F_i Rotation	0	0	M
F_i Bias and Shift	F_i^*	o_i	I
F_i Bias and Rotation	F_i^*	0	M
F_i Shift and Rotation	0	o_i	M
F_i Bias, Shift and Rotation	F_i^*	o_i	M

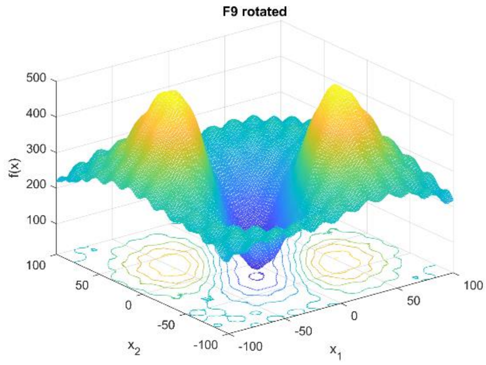
To show the effect of these configurations on the benchmark set, F_9 has been selected as an example. The 3-D maps for 2 dimensions F_9 in all 8 configurations are shown in Figure 8. In each figure, the subfigure (a) shows the basic 3-D map of function such that the no parametrization is used. The subfigures (b), (c), (d) present the function with only shift parameter, only rotation and only bias, respectively. In the subfigures (e), (f), (g), two operations are simultaneously used that are shift with rotation, shift with bias and rotation with bias, respectively. The subfigure (h) shows all the three parameters with the original function. These figures illustrate the effect of all parameters on the original benchmark functions. Moreover, their contour maps (For F_9) is also drawn in Figures 9.



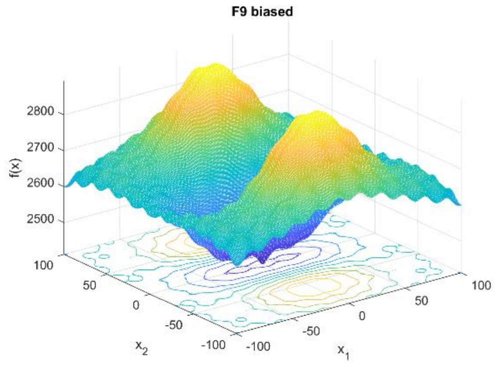
(a) 3-D map for 2-D basic function



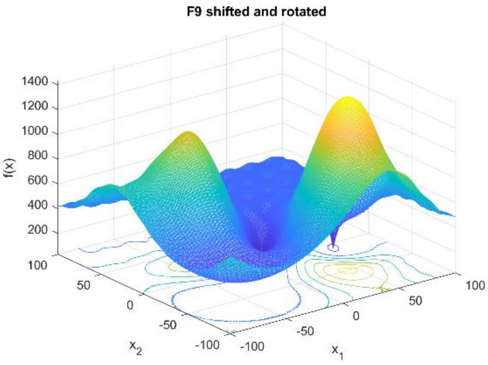
(b) 3-D map for 2-D function with shift operator



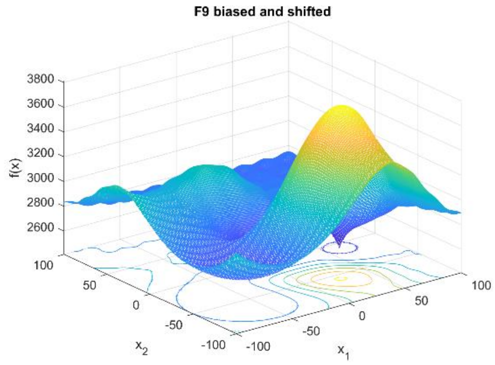
(c) 3-D map for 2-D function with rotation



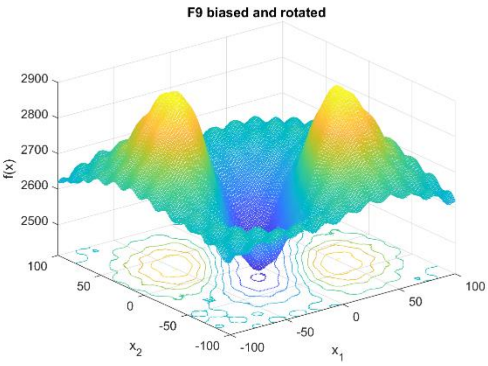
(d) 3-D map for 2-D function with bias



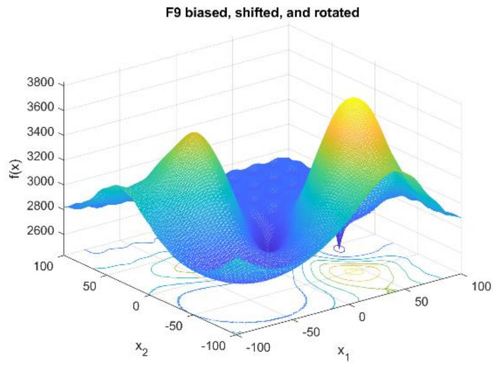
(e) 3-D map for 2-D function with shift and rotation



(f) 3-D map for 2-D function with shift and bias

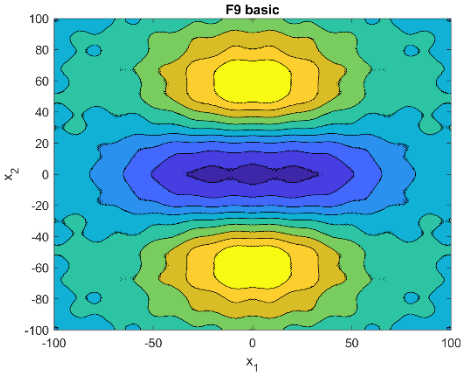


(g) 3-D map for 2-D function with bias and rotation

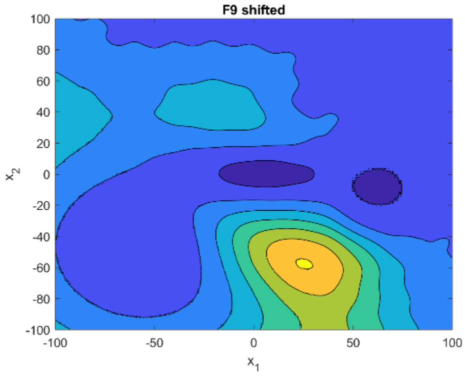


(h) 3-D map for 2-D function with shift, rotation and bias

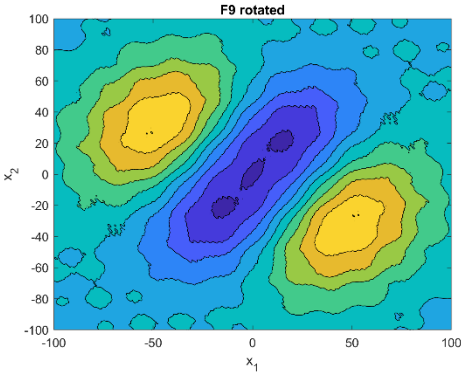
Figure 8: Composite Parameterized Function 9



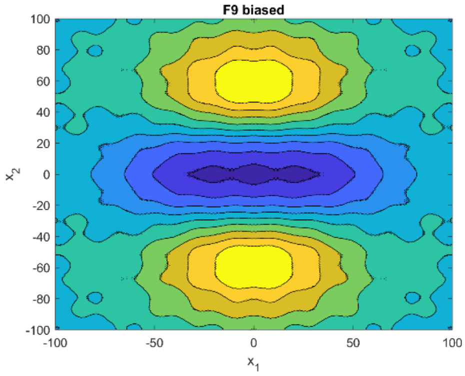
(a) Contour map for 2-D basic function



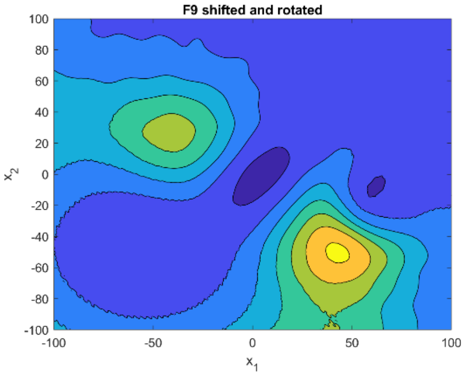
(b) Contour map for 2-D function with shift operator



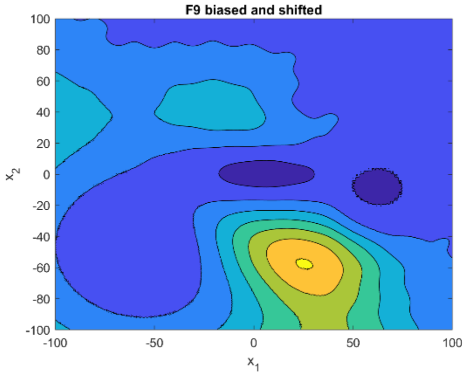
(c) Contour map for 2-D function with rotation



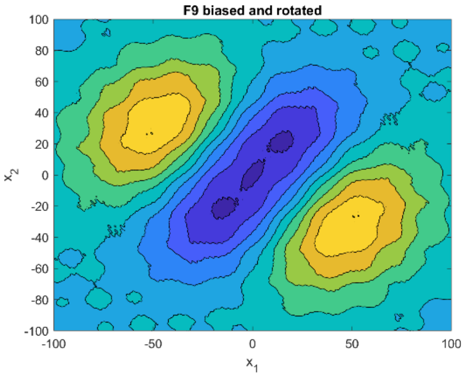
(d) Contour map for 2-D function with bias



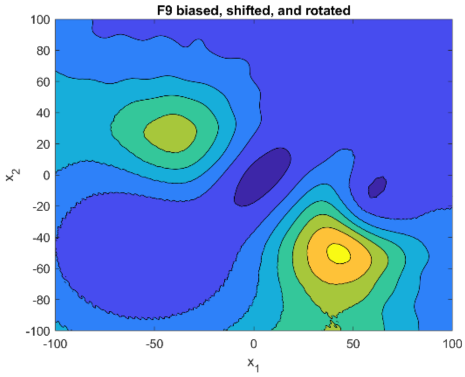
(e) Contour map for 2-D function with shift and rotation



(f) Contour map for 2-D function with shift and bias



(g) Contour for 2-D function with bias and rotation



(h) Contour for 2-D function with shift, rotation and bias

Figure 9: Contour Maps for Composite Parameterized Function 9

3. Experimental Settings and Evaluation Criteria

3.1. Experimental Settings

Problems: 10 minimization problems

Dimensions: For F1-F10, D = 10, and 20;

Runs / problem: 30

MaxFES:

	MaxFES
$D = 10$	200,000
$D = 20$	1,000,000

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

Initialization: Uniform random initialization within the search space. For fair comparison, 1000 uniform random seed was already generated and stored in `'input_data\Rand_Seeds.txt'` file and the random seed for each run is based on four factors: Problem size (D), Function No. (func_no), Runs, and Run Id(run_id) according to:

```
seed_ind=(problem_size/10*func_no*Runs+run_id)-Runs;
seed_ind=mod(seed_ind,1000);
run_seed=Rand_Seeds(seed_ind);
```

Matlab users can use:

```
rng(run_seed,'twister');
```

Global Optimum: All problems have the global optimum within the given bounds and there is no need to search outside of the given bounds for these problems.

$$F_i\left(\mathbf{x}^*\right) = F_i\left(\boldsymbol{o}_i\right) = F_i^*$$

Termination: Terminate when reaching MaxFES or the error value is smaller than 10^{-8}.

3.2. Results Record

1) **Record function error value ($F_i(\mathbf{x}) - F_i(\mathbf{x}^*)$) after** $\left\lfloor D^{\frac{k}{5}-3}\text{MaxEFEs}\right\rfloor$ (k = 0, 1, 2, 3, ... , 15) **for each run.**

For example, in problems with D = 10; the function error value after $\left\lfloor 10^{\frac{0}{5}-3} \times 200{,}000\right\rfloor \left\lfloor 10^{\frac{1}{5}-3} \times 200{,}000\right\rfloor \left\lfloor 10^{\frac{2}{5}-3} \times 200{,}000\right\rfloor \ldots \left\lfloor 10^{\frac{15}{5}-3} \times 200{,}000\right\rfloor$ for each run need to be recorded.

In this case, **16** error values are recorded for each function for each run. Sort the error values achieved after MaxFES in 30 runs from the smallest (best) to the largest (worst) and present the best, worst, mean, median and standard variance values of function error values for the 30 runs.

Please Notice: Error values smaller than 10^{-8} will be taken as zero.

2) Algorithm Complexity

a) Run the test program below:

$x = 0.55$

for i = 1: 200000

$$x = x + x; x = x/2; x = x * x; x = sqrt(x); x = \log(x); x = \exp(x); x = x/(x+2);$$

end

Computing time for the above = *T0*;

b) Evaluate the computing time just for Function 1. For 200000 evaluations of a certain dimension *D*, it gives *T1*;

c) The complete computing time for the algorithm with 200000 evaluations of the same *D* dimensional Function 1 is *T2*.

d) Execute step c five times and get five *T2* values. $\hat{T2} = \text{Mean}(T2)$

The complexity of the algorithm is reflected by: $\hat{T2}$, *T1*, *T0*, and $(\hat{T2} - T1)/T0$.

The algorithm complexities are calculated on 10, 20 dimensions, to show the algorithm complexity's relationship with dimension. Also provide sufficient details on the computing system and the programming language used. In step c, we execute the complete algorithm five times to accommodate variations in execution time due adaptive nature of some algorithms.

Please Note: Similar programming styles should be used for all *T0*, *T1* and *T2*.

(For example, if *m* individuals are evaluated at the same time in the algorithm, the same style should be employed for calculating *T1*; if parallel calculation is employed for calculating *T2*, the same way should be used for calculating *T0* and *T1*. In other word, the complexity calculation should be fair.)

3) **Parameters**

Participants must not search for a distinct set of parameters for each problem/dimension/etc. Please provide details on the following whenever applicable:

a) All parameters to be adjusted

b) Corresponding dynamic ranges

c) Guidelines on how to adjust the parameters

d) Estimated cost of parameter tuning in terms of number of FEs

e) Actual parameter values used.

4) **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.

5) **Results Format**

The participants are required to send the final results as the following format to the organizers and the organizers will present an overall analysis and comparison based on these results.

Create one txt document with the name "AlgorithmName_FunctionNo._D.txt" for each test function and for each dimension.

For example, DE results for test function 5 and D = 10, the file name should be "DE_5_10.txt".

Then save the results matrix (***the gray shadowing part***) as Table I in the file:

Table I Information Matrix for D Dimensional Function X with the configuration Y.

***.txt	Run1	Run2	…	Run30
Function error values when FES= $\left\lfloor D^{\frac{0}{5}-3}\text{MaxEFEs} \right\rfloor$				
Function error values when FES= $\left\lfloor D^{\frac{1}{5}-3}\text{MaxEFEs} \right\rfloor$				
Function error values when FES= $\left\lfloor D^{\frac{2}{5}-3}\text{MaxEFEs} \right\rfloor$				
Function error values when FES= $\left\lfloor D^{\frac{3}{5}-3}\text{MaxEFEs} \right\rfloor$				
… …				
Function error values when FES= $\left\lfloor D^{\frac{14}{5}-3}\text{MaxEFEs} \right\rfloor$				
Function error values when FES= $\left\lfloor D^{\frac{15}{5}-3}\text{MaxEFEs} \right\rfloor$				

For instance, for function F1 with D=10; 8 configurations are possible; therefore, 8 tables will be created for D=10. Similarly, for D=20; 8 tables will also be generated. Thus, total of 16 tables must be presented for function F1.

Therefore, 16 (F1) +16(F2) +… +16 (F10) = 160 files should be zipped and sent to the organizers. Each file contains a 16*30 matrix.

Notice: All participants are allowed to improve their algorithms further after submitting the initial version of their papers to CEC2021. 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.

3.3. Results Template

Language: Matlab 2020b

Algorithm: Differential Evolution (DE)

Results

Notice:

Considering the length limit of the paper, only Error Values Achieved with MaxFES are need to be listed. While the authors are required to send all results to the organizers for a better comparison among the algorithms.

Table II Results for 10 D (Basic)

Func.	Best	Worst	Median	Mean	Std
1					
2					
3					
4					
5					
8					
9					
10					

Table III Results for 10D (Shift Operator)

Func.	Best	Worst	Median	Mean	Std
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Table IV Results for 10D(Rotation)
Table V Results for 10D(Translation)
Table VI Results for 10D (Shift and Rotation)
Table VII Results for 10D (Shift and Translation)
Table VIII Results for 10D (Rotation and Translation)
Table IX Results for 10D (Shift, Rotation and Translation)
Table X Results for 20D (Basic)
Table XI Results for 20D (Shift Operator)
Table XII Results for 20D (Rotation)
Table XIII Results for 20D(Translation)
Table XIV Results for 20D (Shift and Rotation)
Table XV Results for 20D (Shift and Translation)
Table XVI Results for 20D (Rotation and Translation)
Table XVII Results for 20D (Shift, Rotation and Translation)

Algorithm Complexity

Table XVIII Computational Complexity

	T0	T1	$T2$	$(T2-T1)/T0$
D = 10				
D = 20				

3.4. Evaluation Criteria

Algorithms are evaluated with a score that is composed of two parts, *Score*1 and *Score*2, both of which assign equal weights to 10 and 20 dimensional results. *Score*1 is based on sums of normalized error values, while *Score*2 is composed of sums of ranks. Each score contributes 50% to the total *Score*, which has a maximum value of 100.

In particular, *Score*1 begins as an average of 2 sums of normalized functional error values:

$$SNE = 0.5\sum_{m=1}^{m=8}\sum_{i=1}^{i=10} ne_{i,m}^{10D} + 0.5\sum_{m=1}^{m=8}\sum_{i=1}^{i=10} ne_{i,m}^{20D} \quad (22)$$

where ne is an algorithm's normalized error value for a given function, configuration and dimension and SNE is the average of all normalized error values over all functions, configurations and dimensions. For this competition, ne is defined as:

$$ne = \frac{f(\mathbf{x}_{best})-f(\mathbf{x}^*)}{f(\mathbf{x}_{\mathrm{best}})_{\mathrm{max}}-f(\mathbf{x}^*)}\ , \quad (23)$$

where $f(\mathbf{x}_{\mathrm{best}})$ is the algorithm's best result out of 30 trials, $f(\mathbf{x}^*)$ is the function's known optimal value and $f(\mathbf{x}_{\mathrm{best}})_{\mathrm{max}}$ is the largest $f(\mathbf{x}_{\mathrm{best}})$ among all algorithms for the given function/dimension combination. Once *SNE* has been determined for all algorithms, *Score*1 is computed as:

$$Score1 = \left(1 - \frac{SNE - SNE_{\mathrm{min}}}{SNE}\right) \times 50\ , \quad (24)$$

where SNE_{min} is the minimal sum of normalized errors among all algorithms. *Score*2 begins as an average of 2 sums of ranks (*SR*):

$$SR = 0.5\sum_{m=1}^{m=8}\sum_{i=1}^{i=10} rank_{i,m}^{10D} + 0.5\sum_{m=1}^{m=8}\sum_{i=1}^{i=10} rank_{i,m}^{20D}\ , \quad (25)$$

where *rank* is the algorithm's rank among all algorithms for a given function, configuration and dimension that is based on its *mean* error value (not normalized). Once *SR* has been determined for all algorithms, *Score*2 is computed as:

$$Score2 = \left(1 - \frac{SR - SR_{\mathrm{min}}}{SR}\right) \times 50\ , \quad (26)$$

where SR_{min} is the minimal sum of ranks among all algorithms. The final *Score* is the sum of *Score*1 and *Score*2:

$$Score = Score1 + Score2 \quad (27)$$

The entries will be ranked based on the score.

References

[1] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger & S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," Technical Report, Nanyang Technological University, Singapore, May 2005 and KanGAL Report #2005005, IIT Kanpur, India, 2005.

[2] J. J. Liang, B. Y. Qu, P. N. Suganthan, Alfredo G. Hernández-Díaz, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization", Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, January 2013.

[3] J. J. Liang, B. Y. Qu, P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Real-Parameter Optimization", Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, January 2014.

[4] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, B. Y. Qu, " Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization", Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, October 2016.

[5] C. T. Yue, K. V. Price, P. N. Suganthan, J. J. Liang, M. Z. Ali, B. Y. Qu, N. H. Awad, & P.P Biswas, “Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization”, Technical Report. Rep., Zhengzhou University and Nanyang Technological University, 2019.

[6] K. R. Opara, A. A. Hadi, A. W. Mohamed, “Parametrized Benchmarking: an outline of the idea and a feasibility study”. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. GECCO ’20 Companion, July 8–12, 2020, Cancún, Mexico https://doi.org/10.1145/3377929.3389944

[7] Xiaodong Li, Ke Tang, Mohammad N. Omidvar, Zhenyu Yang, and Kai Qin, "Benchmark Functions for the CEC’2013 Special Session and Competition on Large-Scale Global Optimization", Technical Report, 2013.

[8] N. Hansen, A. Auger, S. Finck, and R. Ros. 2009. Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental setup. Technical Report. INRIA. http://coco.gforge.inria.fr