Guideline for the CEC 2015 Competition on Single Objective Multi-Niche Optimization

B. Y. Qu[1], J. J. Liang[2], P. N. Suganthan[3], Q. Chen[4]

[1] School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China

[1] School of Electrical Engineering, Zhengzhou University, Zhengzhou, China

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

[4]Facility Design and Instrument Institute, China Aerodynamic Research and Development Center, China

qby1984@hotmail.com, liangjing@zzu.edu.cn, epnsugan@ntu.edu.sg, chenqin1980@gmail.com

January 2015

1. Summary of the CEC’15 Multi-Niche Test Suite:

Table I. Summary of the CEC’15 CEC’15 Multi-Niche Test Functions

	No.	Functions	Dimension	Goal optima No. global/local*	$F_i^*=F_i(x^*)$
Expanded Scalable Function	1	Shifted and Rotated Expanded Two-Peak Trap	5	1/15	100
			10	1/55	
			20	1/210	
	2	Shifted and Rotated Expanded Five-Uneven-Peak Trap	2	4/21	200
			5	32/0	
			8	256/0	
	3	Shifted and Rotated Expanded Equal Minima	2	25/0	300
			3	125/0	
			4	625/0	
	4	Shifted and Rotated Expanded Decreasing Minima	5	1/15	400
			10	1/55	
			20	1/210	
	5	Shifted and Rotated Expanded Uneven Minima	2	25/0	500
			3	125/0	
			4	625/0	
	6	Shifted and Rotated Expanded Himmelblau’s Function	4	16/0	600
			6	64/0	
			8	256/0	
	7	Shifted and Rotated Expanded Six-Hump Camel Back	6	8/0	700
			10	32/0	
			16	256/0	
	8	Shifted and Rotated Modified Vincent Function	2	36/0	800
			3	216/0	
			4	1296/0	
Composition Function	9	Composition Function 1	10, 20, 30	5	900
	10	Composition Function 2	10, 20, 30	5	1000
	11	Composition Function 3	10, 20, 30	5	1100
	12	Composition Function 4	10, 20, 30	5	1200
	13	Composition Function 5	10, 20, 30	5	1300
	14	Composition Function 6	10, 20, 30	5	1400
	15	Composition Function 7	10, 20, 30	5	1500
Search Range: $[-100,100]^D$ level of accuracy = 0.1 Error values smaller than 10^{-8} will be taken as zero.					

2. Code

You can download the C, JAVA and Matlab codes for CEC'15 niching test suite from the website given below:

http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2015/CEC2015.htm

3. Details of the benchmark suite

Please refer to:

B. Y. Qu, J. J. Liang, Z. Y. Wang and P. N. Suganthan "Performance evaluation of novel multi-model problems," submitted to *Swarm and Evolutionary Computation*, 2015

4. Requirement

1. These problems should be treated as **black-box** problems. The explicit equations of the problems are not to be used.
2. Goal Peaks global/local here is the total number of global and local solutions required. Some test functions have numerous poor quality local optima while algorithms are expected to capture the best local solutions as required.

Note That the first 8 problems with 3 variable dimensions are compulsive for basic competition. The rest 7 composition functions are highly recommended for further comparison.

5. Experimental Setting

Problems: 15 minimization problems

Dimensions: Refer to Table I

Runs / problem: 50 (**Do not run many 50 runs to pick the best run**)

MaxFES: $2000 * D * \sqrt{q}$. Here q is the goal optima number. (For example, for 5D function 1, q=4, MaxFES=2000*5*2=20000).

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

Initialization: Uniform random initialization within the search space. Random seed is based on time, Matlab users can use rand('state', sum(100*clock)).

Global Optima: All problems have the required number of optima within the given bounds and it is NOT allowed to perform search outside of the given bounds for these problems, as solutions outside of the bounds are regarded as infeasible.

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

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

6. Performance Metric

The performance of all multimodal algorithms is measured in terms of the following criteria:

1. For the first 8 extended simple problems, average number of optima found is used as the assessment criterion using the given level of accuracy. A level of accuracy, typically $0<\varepsilon<1$, is a value indicating how close the computed solutions to the known global peaks are. If the difference from a computed solution to a known global optimum is below ε, then the peak is considered to have been found.
2. For the 7 composition functions, a different measurement method is used. For each problems, all the algorithms need to provide 5 best optima (list the parameters as well as the objective values) of median run that mutually separated at least by *D* in Euclidean distance (*D* is the dimension). The results are compared based on these solutions.

7. Results Record

1) For functions 1-8, calculate **Average Number of Optima Found,** according to section 6 and present the **best, worst, mean** and **standard variance** values of these four performance metrics for the 50 runs.

2) For functions 9-15, since the participants are required to search for the optima based on the distance among optima and the exact positions of these goal optima are not provided, the performance metrics cannot be calculated. For these problems, the mean error values ($F_i(\mathbf{x})$-$F_i(\mathbf{x}^*)$) of 5 best optima of each run should be calculated and the **best, worst, mean** and **standard variance** values of these mean error values should be presented in table for comparison.

3) Algorithm Complexity

a) Run the test program below:

for *i*=1:1000000

x= 0.55 + (double) *i*;

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 8**. 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 8** is *T2*.

d) Execute step c **five** times and get **five** *T2* values. $\hat{T}2$ =Mean(*T2*)

The complexity of the algorithm is reflected by: $\hat{T}2$, *T1*, *T0*, and ($\hat{T}2$ -*T1*)/*T0*

The algorithm complexities are calculated on 10, 20, 30 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.)

4) 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.

5) 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 required format to the organizers and the organizers will present an overall analysis and comparison based on these results.

a) Record the obtained q solutions and the corresponding function error value ($F_i(\boldsymbol{x})$-$F_i(\boldsymbol{x}^*)$) at MaxFES for each run.

Create one txt document with the name "AlgorithmName_FunctionNo._D_RunsNo..txt" for each run.

For example, PSO results for test function 5 and D=30, the file name for the first run should be "PSO_5_30_1.txt".

Then save the results matrix (***the gray shadowing part, a (D+1)*q matrix***) as Table II in the file. Thus there should be **50 txt files** for each function of a certain dimension.

Table II. Results Matrix for *D* Dimensional Function X of ^{ith} run

$f(\boldsymbol{x}_1)$	x_{11}	x_{12}	…	x_{1D}
$f(\boldsymbol{x}_1)$	x_{21}	x_{22}	…	x_{2D}
…	…	…	…	…
$f(\boldsymbol{x}_q)$	x_{q1}	x_{q2}	…	x_{qD}

Please Note:

1. Error value smaller than 10^{-8} will be taken as zero. Predefined level of accuracy = 0.1.

2. Please check the solutions to make sure they are in the predefined search range. Final solutions out of the bound are not acceptable.

b) FES used for finding each optimum (satisfying the predefined level of accuracy).

In this case, *q* FES values are recorded for each function for each run. If the optimum is not found in the end of the run, record FES=Inf in the results.

Create one txt document with the name "AlgorithmName_FunctionNo._D_FES..txt" for each run.

For example, PSO results for test function 5 and *D*=30, the file name for the first run should be "PSO_5_30_FES.txt".

Then save the results matrix (***the gray shadowing part, a* 50*q *matrix***) as Table III in the file. Thus there should be **one txt file** for each function of a certain dimension. With the 51 error matrix mentioned in (a), ***51 txt files* are required to submitted to the organizer for a function of a certain dimension in total.**

Table III. FES matrix for *D* Dimensional Function X

***FES.txt	1	2	…	*q*
Run 1				
Run 2				
...				
Run 51				

Notice: All participants are allowed to improve their algorithms further after submitting the initial version of their papers to CEC2015. 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.

8. Results Template

Language: Matlab 2013a

Algorithm: Particle Swarm Optimizer (PSO)

Results

Table IV. Average Number of Optima Found (Functions 1-8)

Func.	Dimension	Best	Worst	Mean	Std
	5				
1	10				
	20				
	2				
2	5				
	8				
	2				
3	3				
	4				
	5				
4	10				
	20				
	2				
5	3				
	4				
	4				
6	6				
	8				
	6				
7	10				
	16				
	10				
8	20				
	30				

Table V. Average Error Values for Functions 9-15

Func.	Dimension	Best	Worst	Mean	Std
	10				
9	20				
	30				
	10				
10	20				
	30				
	10				
11	20				
	30				

12	10
	20
	30
13	10
	20
	30
14	10
	20
	30
15	10
	20
	30

Algorithm Complexity

Table XVI. Computational Complexity

	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
D=10				
D=20				
D=30				

Parameters

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.