Problem Definitions and Evaluation Criteria forComputational Expensive Optimization

B. Liu[1], Q. Chen[2] and Q. Zhang[3], J. J. Liang[4], P. N. Suganthan[5], B. Y. Qu[6]

[1]Department of Computing, Glyndwr University, UK
[2]Facility Design and Instrument Institute, China Aerodynamic Research and Development Center, China
[3]Department of Computer Science, City University of Hong Kong, Hong Kong& School of Computer Science and Electronic Engineering, University of Essex, UK.
[4] School of Electrical Engineering, Zhengzhou University, Zhengzhou, China
[5]School of EEE, Nanyang Technological University, Singapore
[6]School of Electric and Information Engineering,Zhongyuan University of Technology,Zhengzhou, China

b.liu@glyndwr.ac.uk, qingfu.zhang@cityu.edu.hk,chenqin1980@gmail.com
liangjing@zzu.edu.cn, epnsugan@e.ntu.edu.sg, qby1984@hotmail.com

Many real-world optimization problems require computationally expensive computer or physical simulations for evaluating their candidate solutions. Often, traditional gradient-based mathematical programming methods cannot be applied directly to these problems since analytic formulations are unavailable. Evolutionary algorithms (EA) cannot directly solve them either since a large number of function evaluations are unaffordable. In recent years, various kinds of novel methods for computationally expensive optimization problems have been proposed and surrogate model assisted evolutionary algorithm (SAEA) is attracting more and more attention.

To promote research on expensive optimization, we propose to organize a competition focusing on small- to medium-scale (from 10 decision variables to 30 decision variables) real parameter bound constrained single-objective computationally expensive optimization. We encourage all participants to test their algorithms on the CEC 14 expensive optimization test suite which includes 24 black-box benchmark functions (8 popular test problems with 10, 20 and 30 dimensions). The participants are required to send the final results in the format given in the technical report to the organizers.The organizers will conduct an overall analysis and comparison. Special attention will be paid to which algorithm has advantages on which kind of problems.

The C and Matlab codes for CEC'14 test suite can be downloaded from the website given below:

http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014

1. Introduction to the 24 CEC'14 expensive optimization test problems

1.1 Summary of CEC'14 expensive optimization test problems

Eight popular test functions are used. The test suites include unimodal / multi-modal, continuous / discrete and separable / non-separable functions. All test functions are scalable and 10 decision variables, 20 decision variables and 30 decision variables are used. The test problems are summarized in Table I.

Table I. Summary of the CEC' 14 expensive optimization test problems

No	Functions	Dimensionality	Search ranges	$f_i^* = f_i(x^*)$
1	Sphere function	10, 20, 30	[-5.12,5.12]	0
2	Ellipsoid function	10, 20, 30	[-5.12,5.12]	0
3	Rotated Ellipsoid function	10, 20, 30	[-5.12,5.12]	0
4	Step function	10, 20, 30	[-5.12,5.12]	0
5	Ackley's function	10, 20, 30	[-32,32]	0
6	Griewank's function	10, 20, 30	[-600,600]	0
7	Rosenbrock's function	10, 20, 30	[-2.048,2.048]	0
8	Rastrigin's function	10, 20, 30	[-5.12,5.12]	0

Please notice: These problems should be treated as black-box optimization problems and without any prior knowledge. Neither the analytical equations nor the problem landscape characters extracted from analytical equations are allowed to be used, except the continuous / integer decision variables. However, the dimensionality and the number of available function evaluations can be considered as known values and can be used.

1.2Definitions of CEC'14 expensive optimization test problems

1) Sphere function

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$$

$$F_1(\boldsymbol{x}) = f_1(x) : D = 10, 20, 30$$

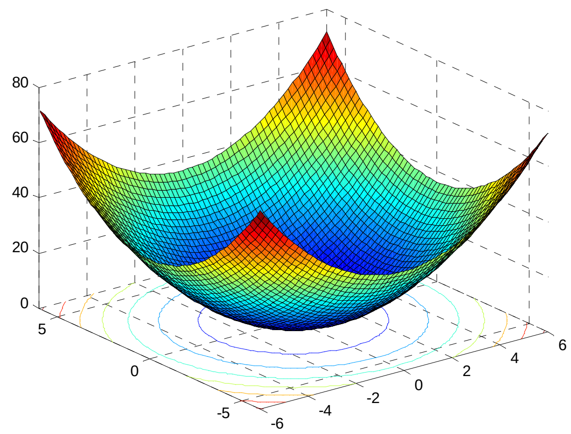

Figure 1. 3-D map for 2-D Sphere function

Properties:

- Unimodal

2) Ellipsoid function

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{D} i x_i^2$$

$$F_2(\boldsymbol{x}) = f_2(x) : D = 10, 20, 30$$

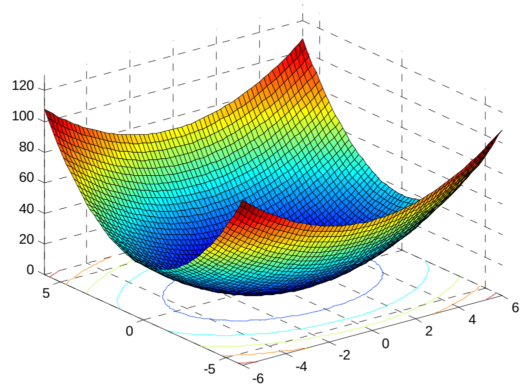

Figure 2. 3-D map for 2-D Ellipsoid function

Properties:

- ➢ Unimodal

3) Rotated Ellipsoid function

$$F_3(\boldsymbol{x}) = f_2(\mathrm{M}x) : D = 10, 20, 30$$

M is the rotation matrix.

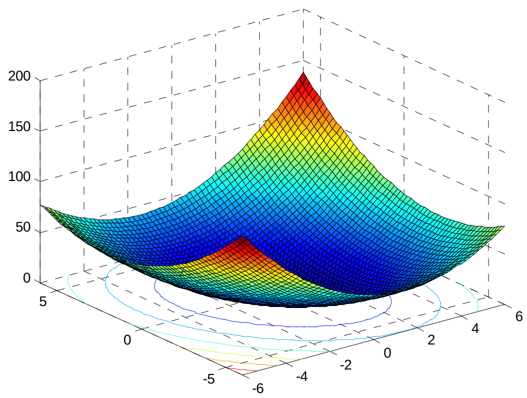

Figure 3. 3-D map for 2-D Rotated Ellipsoid function

Properties:

- ➢ Unimodal

4) Step function

$$f_4(\mathbf{x}) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$$

$$F_4(\mathbf{x}) = f_4(x) : D = 10, 20, 30$$

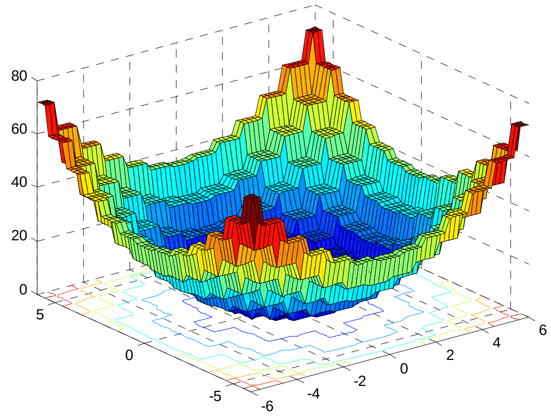

Figure 4. 3-D map for 2-D Step function

Properties:

- Unimodal
- Discontinuous

5) Ackley's function

$$f_5(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$$

$$F_5(\mathbf{x}) = f_5(x) : D = 10, 20, 30$$

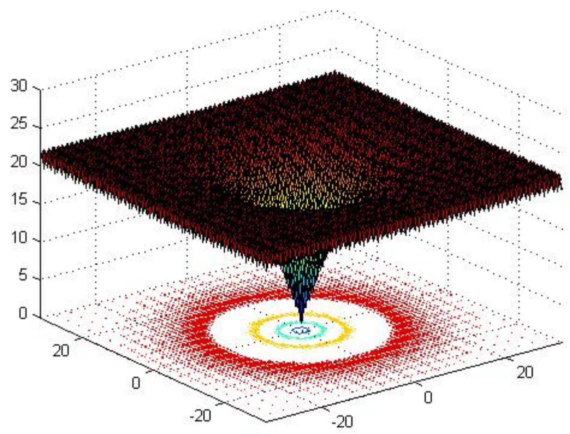

Figure 5. 3-D map for 2-D Ackley's function

Properties:

- Multi-modal

6) Griewank's function

$$f_6(\boldsymbol{x}) = \sum_{i=1}^{D} \frac{x_i^{\ 2}}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$$

$$F_6(\boldsymbol{x}) = f_6(x) : D = 10, 20, 30$$

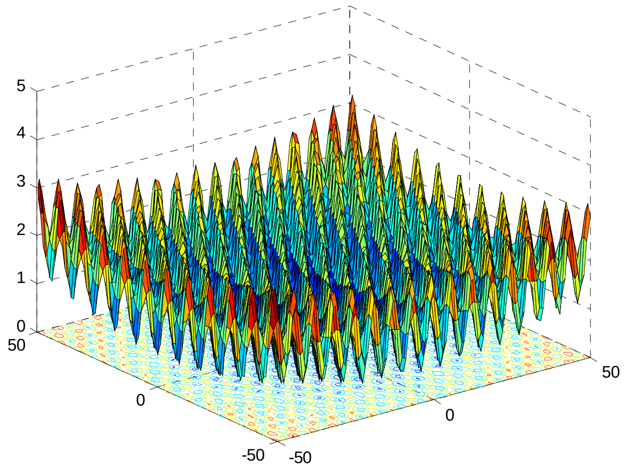

Figure 6. 3-D map for 2-D Griewank'sfunction

Properties:

- Multi-modal

7) Rosenbrock's function

$$f_7(\boldsymbol{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$F_7(\boldsymbol{x}) = f_7(x) : D = 10, 20, 30$$

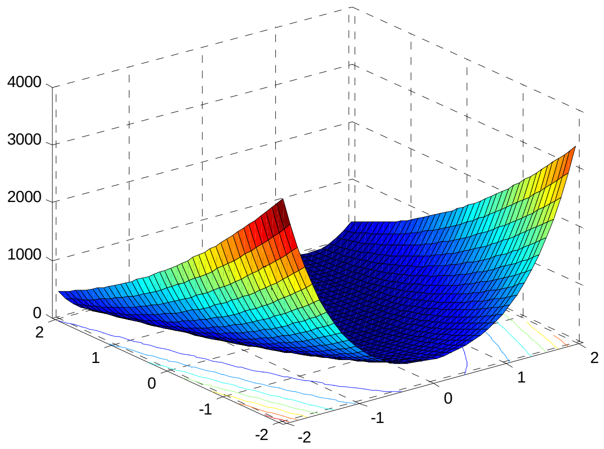

Figure 7. 3-D map for 2-D Rosenbrock'sfunction

Properties:

- Multi-modal
- Non-separable
- Having a very narrow valley from local optimum to global optimum

8) Rastrigin's function

$$f_8(\boldsymbol{x}) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$$

$$F_8(\boldsymbol{x}) = f_8(x): D = 10, 20, 30$$

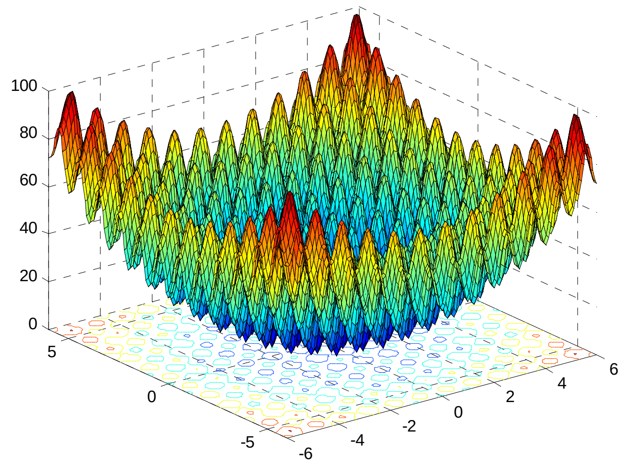

Figure 8. 3-D map for 2-D Rastrigin'sfunction

Properties:

- Multi-modal

2. Evaluation criteria

2.1 Experimental setting:

- Number of independent runs: 30
- Maximum number of exact function evaluations:
  - 10-dimensional problems: 500
  - 20-dimensional problems: 1,000
  - 30-dimensional problems: 1,500
- Initialization: Any problem-independent initialization method is allowed.
- Global optimum: All problems have the global optimum within the given bounds and there is no need to perform search outside of the given bounds for these problems.
- Termination: Terminate when reaching the maximum number of exact function evaluations or the error value ($f_i^* - f_i(x^*)$) is smaller than 10^{-8}.

2.2 Results to record:

(1) Current best function values:

Record current best function values in each iteration for each run. Sort the obtained best function values after the maximum number of exact function evaluations from the smallest (best) to the largest (worst) and present the best, worst, mean, median and standard deviation values for the 30 runs. Error values smaller than 10^{-8} are taken as zero.

(2) Algorithm complexity:

For expensive optimization, the criterion to judge the efficiency is the obtained best result vs. number of exact function evaluations. But the computational overhead on surrogate modeling and search is also considered as a secondary evaluation criterion. Considering that for different data sets, the computational overhead for a surrogate modeling method can be quite different, the computational overhead of each problem is necessary to be reported. Often, compared to the computational cost on surrogate modeling, the cost on 500, 1000 and 1500 function evaluations can almost be ignored. Hence, the following method is used:

a) Run the test program below:

```
fori=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) The average complete computing time for the algorithm = $\widehat{T}1$

The complexity of the algorithm is measured by: $\widehat{T}1/T0$.

(3) Parameters:

Participants are requested not to search for the best 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, dimensionality of the problems, etc.

(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_expensive.txt" foreach test function and for each dimension. For example, PSO results for test function 5 and D=30, the file name should be "PSO_5_30_expensive.txt".

The txt document should contain the mean and median values of current best function values of each iteration of all the 30 runs. The participant can save the results in the matrix shown in Table II and extracts the mean and median values of each iteration.

Table II Information matrix for function X

	Iteration 1	Iteration 2	…	Iteration 500/1000/1500
Run 1				
Run 2				
…				
Run 30				

Notice: All participants are allowed to improve their algorithms further after submitting the initial version of their papers to CEC2014. 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.

2.3 Results template

Language: Matlab2008a

Algorithm: Surrogate model assisted evolutionary algorithm A

Results
Notice:

Considering the length limit of the paper, only Error Values Achieved with MaxFESare need to be listed.

Table III. Results for 10D

Func.	Best	Worst	Median	Mean	Std
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					

Table IV. Results for 20D

...

Table V. Results for 30D

...

Algorithm Complexity

Table VI. Computational Complexity

Func.	$\widehat{T1}/T0$
1	
2	
3	
...	
23	
24	

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

2.4 Sorting method

The mean and median values at the maximum allowed number of evaluations will be used. For each problem, the algorithm with the best result scores 9, the second best scores 6, the third best scores 3 and all the others score 0.

$$\text{Total score} = \sum_{i=1}^{24} score_i \text{ (using mean value)} + \sum_{i=1}^{24} score_i \text{ (using median value)}$$

The top three winners will be announced.

Special attention will be paid to which algorithm has advantages on which kind of problems, considering dimensionality and problem characteristics.