Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization

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

[1] School of Electrical Engineering, Zhengzhou University, Zhengzhou, China
[2] School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China
[3] School of EEE, Nanyang Technological University, Singapore

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

Technical Report 201311, Computational Intelligence Laboratory,
Zhengzhou University, Zhengzhou China
And
Technical Report, Nanyang Technological University, Singapore

December 2013

Research on the single objective optimization algorithms is the basis of the research on the more complex optimization algorithms such as multi-objective optimizations algorithms, niching algorithms, constrained optimization algorithms and so on. All new evolutionary and swarm algorithms are tested on single objective benchmark problems. In addition, these single objective benchmark problems can be transformed into dynamic, niching composition, computationally expensive and many other classes of problems.

In the recent years various kinds of novel optimization algorithms have been proposed to solve real-parameter optimization problems, including the CEC'05 and CEC'13 Special Session on Real-Parameter Optimization[1][2]. Considering the comments on the CEC'13 test suite, we organize a new competition on real parameter single objective optimization.

For this competition, we are developing benchmark problems with several novel features such as novel basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. **This competition excludes usage of surrogates or meta-models.** There is a sub-competition to test the algorithms with a very small number of function evaluations in order emulate the computationally expensive optimization scenario. This sub-competition encourages the usage of surrogates and other approximation approaches.

This special session is devoted to the approaches, algorithms and techniques for solving real parameter single objective optimization without making use of the exact equations of the test functions. We encourage all researchers to test their algorithms on the CEC'14 test suite which includes 30 benchmark functions. The participants are required to send the final results in the format specified in the technical report to the organizers. The organizers will present an overall analysis and comparison based on these results. We will also use statistical tests on convergence performance to compare algorithms that generate similar final solutions eventually. Papers on novel concepts that help us in understanding problem characteristics are also welcome.

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 CEC’14 Benchmark Suite

1.1 Some Definitions:

All test functions are minimization problems defined as following:

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

D: dimensions.

$\boldsymbol{o}_{i1} = [o_{i1}, o_{i2}, ..., o_{iD}]^{\mathrm{T}}$: the shifted global optimum (defined in “shift_data_x.txt”), which is randomly distributed in $[-80,80]^D$. Different from CEC’13, each function has a shift data for CEC’14.

All test functions are shifted to ***o*** and scalable.

For convenience, the same search ranges are defined for all test functions.

Search range: $[-100,100]^D$.

$\mathbf{M}_i$: rotation matrix. Different from CEC’13, different rotation matrix are assigned to each function and each basic function.

Considering that in the real-world problems, it is seldom that there exist linkages among all variables. In CEC’14 the variables are divided into subcomponents randomly. The rotation matrix for each subcomponents are 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'14 Test Suite

Table I. Summary of the CEC'14 Test Functions

	No.	Functions	$F_i^*=F_i(x^*)$
Unimodal Functions	1	Rotated High Conditioned Elliptic Function	100
	2	Rotated Bent Cigar Function	200
	3	Rotated Discus Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Ackley's Function	500
	6	Shifted and Rotated Weierstrass Function	600
	7	Shifted and Rotated Griewank's Function	700
	8	Shifted Rastrigin's Function	800
	9	Shifted and Rotated Rastrigin's Function	900
	10	Shifted Schwefel's Function	1000
	11	Shifted and Rotated Schwefel's Function	1100
	12	Shifted and Rotated Katsuura Function	1200
	13	Shifted and Rotated HappyCat Function	1300
	14	Shifted and Rotated HGBat Function	1400
	15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
	16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid Function 1	17	Hybrid Function 1 (N=3)	1700
	18	Hybrid Function 2 (N=3)	1800
	19	Hybrid Function 3 (N=4)	1900
	20	Hybrid Function 4 (N=4)	2000
	21	Hybrid Function 5 (N=5)	2100
	22	Hybrid Function 6 (N=5)	2200
Composition Functions	23	Composition Function 1 (N=5)	2300
	24	Composition Function 2 (N=3)	2400
	25	Composition Function 3 (N=3)	2500
	26	Composition Function 4 (N=5)	2600
	27	Composition Function 5 (N=5)	2700
	28	Composition Function 6 (N=5)	2800
	29	Composition Function 7 (N=3)	2900
	30	Composition Function 8 (N=3)	3000
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) High Conditioned Elliptic Function

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} \boldsymbol{x}_i^2 \tag{1}$$

2) Bent Cigar Function

$$f_2(\boldsymbol{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2 \tag{2}$$

3) Discus Function

$$f_3(\boldsymbol{x}) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2 \tag{3}$$

4) Rosenbrock's Function

$$f_4(\boldsymbol{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2), \tag{4}$$

5) Ackley's Function

$$f_5(\boldsymbol{x}) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i)) + 20 + e \tag{5}$$

6) Weierstrass Function

$$f_6(\boldsymbol{x}) = \sum_{i=1}^{D} (\sum_{k=0}^{k\max} [a^k \cos(2\pi b^k (x_i + 0.5))]) - D \sum_{k=0}^{k\max} [a^k \cos(2\pi b^k \cdot 0.5)] \tag{6}$$

a=0.5, b=3, $kmax$=20

7) Griewank's Function

$$f_7(\boldsymbol{x}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{7}$$

8) Rastrigin's Function

$$f_8(\boldsymbol{x}) = \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i) + 10) \tag{8}$$

9) Modified Schwefel's Function

$$f_9(\boldsymbol{x}) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i), \qquad z_i = x_i + 4.209687462275036\text{e+}002$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{if } |z_i| \le 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|500 - \text{mod}(z_i, 500)|}) - \dfrac{(z_i - 500)^2}{10000D} & \text{if } z_i > 500 \\ (\text{mod}(|z_i|, 500) - 500) \sin(\sqrt{|\text{mod}(|z_i|, 500) - 500|}) - \dfrac{(z_i + 500)^2}{10000D} & \text{if } z_i < -500 \end{cases} \tag{9}$$

10) Katsuura Function

$$f_{10}(\boldsymbol{x}) = \frac{10}{D^2} \prod_{i=1}^{D} (1 + i \sum_{j=1}^{32} \frac{|2^j x_i - round(2^j x_i)|}{2^j})^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} \tag{10}$$

11) HappyCat Function

$$f_{11}(\boldsymbol{x}) = \left| \sum_{i=1}^{D} x_i^2 - D \right|^{1/4} + (0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i) / D + 0.5 \tag{11}$$

12) HGBat Function

$$f_{12}(\boldsymbol{x}) = \left| (\sum_{i=1}^{D} x_i^2)^2 - (\sum_{i=1}^{D} x_i)^2 \right|^{1/2} + (0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i) / D + 0.5 \tag{12}$$

13) Expanded Griewank's plus Rosenbrock's Function

$$f_{13}(\boldsymbol{x}) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + ... + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1)) \tag{13}$$

14) Expanded Scaffer's F6 Function

Scaffer's F6 Function: $g(x, y) = 0.5 + \dfrac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$

$$f_{14}(\boldsymbol{x}) = g(x_1, x_2) + g(x_2, x_3) + ... + g(x_{D-1}, x_D) + g(x_D, x_1) \tag{14}$$

1.4 Definitions of the CEC'14 Test Suite

A. Unimodal Functions:

1) Rotated High Conditioned Elliptic Function

$$F_1(\boldsymbol{x}) = f_1(\mathbf{M}(\boldsymbol{x}-\boldsymbol{o}_1)) + F_1{}^* \tag{15}$$

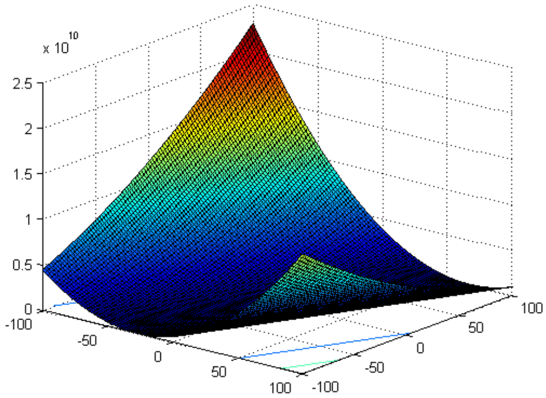

Figure 1. 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Non-separable
- Quadratic ill-conditioned

2) Rotated Bent Cigar Function

$$F_2(\boldsymbol{x}) = f_2(\mathbf{M}(\boldsymbol{x}-\boldsymbol{o}_2)) + F_2{}^* \tag{16}$$

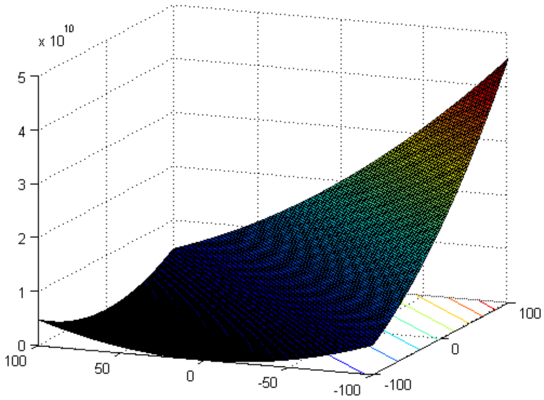

Figure 2. 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Non-separable
- Smooth but narrow ridge

3) Rotated Discus Function

$$F_3(\boldsymbol{x}) = f_3(\mathbf{M}(\boldsymbol{x} - \boldsymbol{o}_3)) + F_3{}^* \tag{17}$$

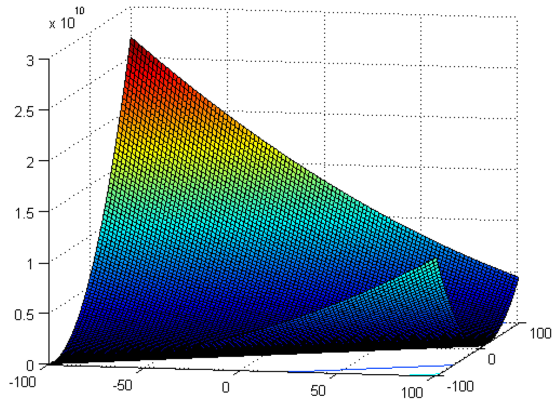

Figure 3. 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Non-separable
- With one sensitive direction

B. Multimodal Functions

4) Shifted and Rotated Rosenbrock's Function

$$F_4(\boldsymbol{x}) = f_4(\mathbf{M}(\frac{2.048(x - o_4)}{100}) + 1) + F_4{}^* \tag{18}$$

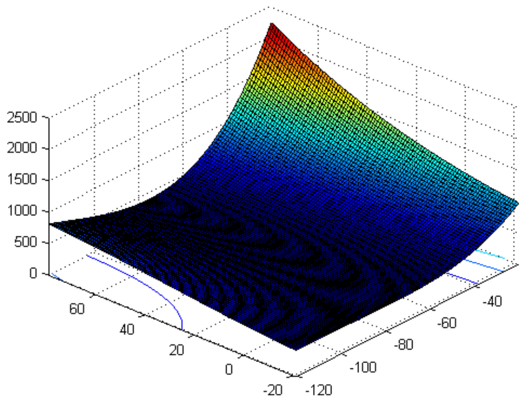

Figure 4(a). 3-*D* map for 2-*D* function

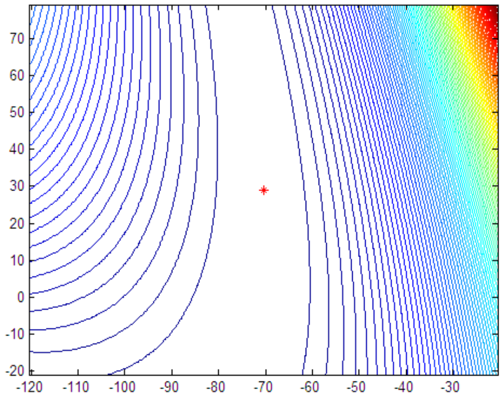

Figure 4(b).Contour map for 2-*D* function

Properties:

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

5) Shifted and Rotated Ackley's Function

$$F_5(\boldsymbol{x}) = f_5(\mathbf{M}(\boldsymbol{x} - \boldsymbol{o}_5)) + F_8 * \tag{19}$$

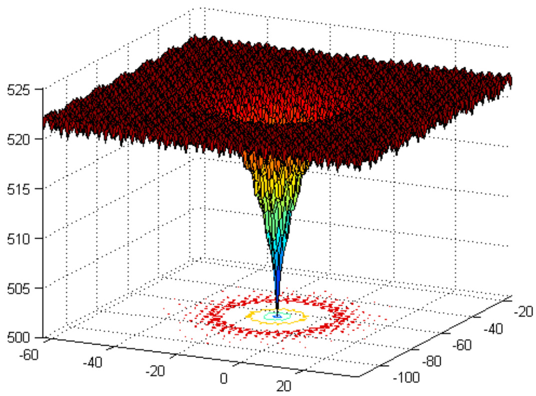

Figure 5. 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable

6) Shifted and Rotated Weierstrass Function

$$F_6(\boldsymbol{x}) = f_6(\mathbf{M}(\frac{0.5(\boldsymbol{x} - \boldsymbol{o}_6)}{100})) + F_6 * \tag{20}$$

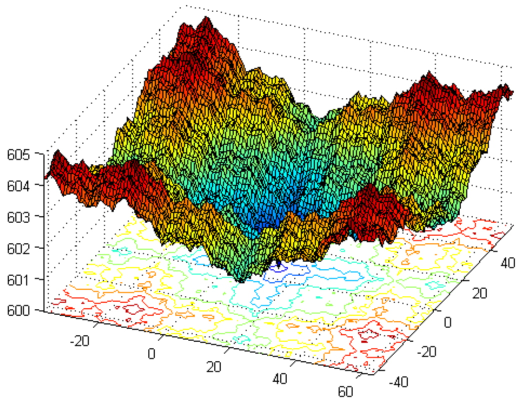

Figure 6. 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable
- Continuous but differentiable only on a set of points

7) Shifted and Rotated Griewank's Function

$$F_7(\mathbf{x}) = f_7(\mathbf{M}(\frac{600(\mathbf{x}-\mathbf{o}_7)}{100})) + F_7^* \tag{21}$$

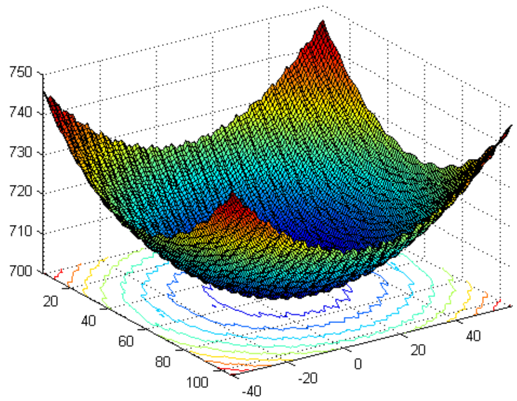

Figure 7(a). 3-*D* map for 2-*D* function

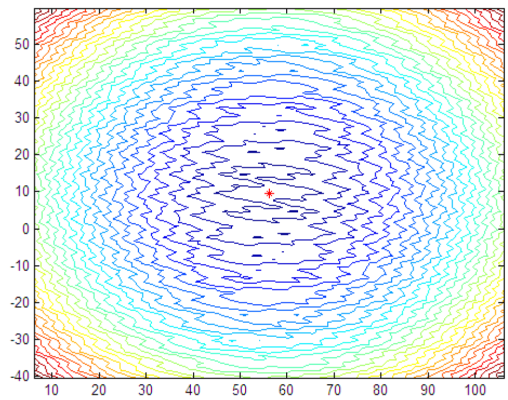

Figure 7(b).Contour map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Non-separable

8) Shifted Rastrigin's Function

$$F_8(\boldsymbol{x}) = f_8\left(\frac{5.12(\boldsymbol{x}-\boldsymbol{o}_8)}{100}\right) + F_7 * \tag{22}$$

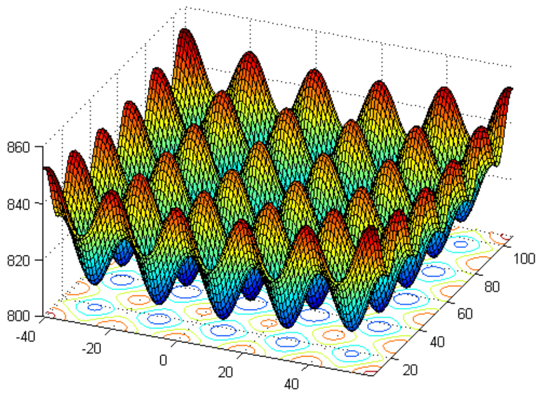

Figure 8. 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Separable
- Local optima's number is huge

9) Shifted and Rotated Rastrigin's Function

$$F_9(\boldsymbol{x}) = f_8\left(\mathbf{M}\left(\frac{5.12(\boldsymbol{x}-\boldsymbol{o}_9)}{100}\right)\right) + F_9 * \tag{23}$$

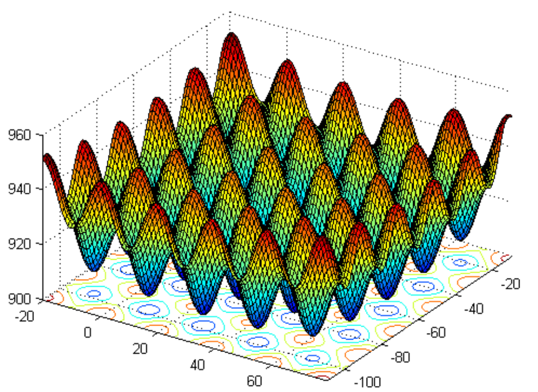

Figure 9. 3-*D* map for 2-*D* function

Properties:

- Multi-modal

- Non-separable
- Local optima's number is huge

10) Shifted Schwefel's Function

$$F_{10}(\boldsymbol{x}) = f_9\left(\frac{1000(\boldsymbol{x}-\boldsymbol{o}_{10})}{100}\right) + F_{10}{}^* \tag{24}$$

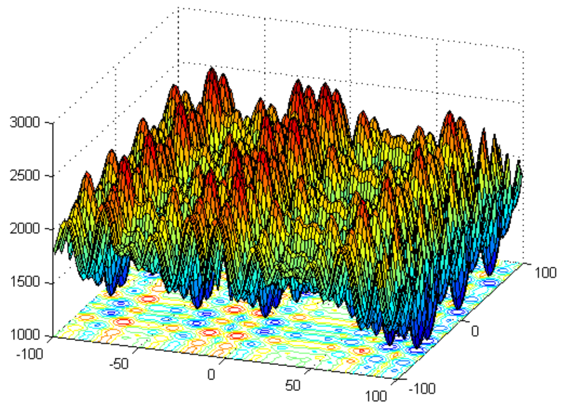

Figure 10(a). 3-*D* map for 2-*D* function

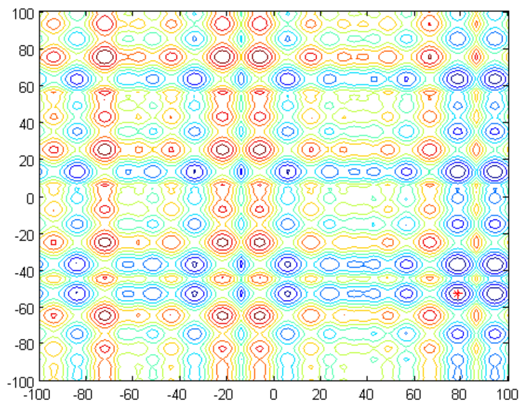

Figure 10(b).Contour map for 2-*D* function

Properties:

- Multi-modal
- Separable
- Local optima's number is huge and second better local optimum is far from the global optimum.

11) Shifted and Rotated Schwefel's Function

$$F_{11}(\boldsymbol{x}) = f_9\left(\mathbf{M}\left(\frac{1000(\boldsymbol{x}-\boldsymbol{o}_{11})}{100}\right)\right) + F_{11}{}^* \tag{25}$$

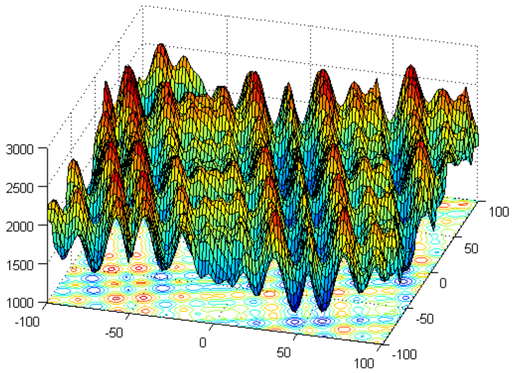

Figure 11(a). 3-*D* map for 2-*D* function

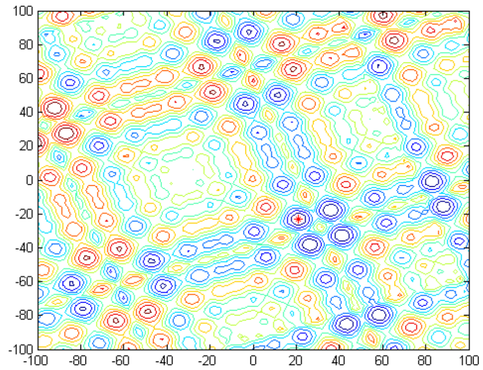

Figure 11(b).Contour map for 2-*D* function

Properties:

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

12) Shifted and Rotated Katsuura Function

$$F_{12}(\boldsymbol{x}) = f_{10}(\mathbf{M}(\frac{5(\boldsymbol{x}-\boldsymbol{o}_{12})}{100})) + F_{12}* \tag{26}$$

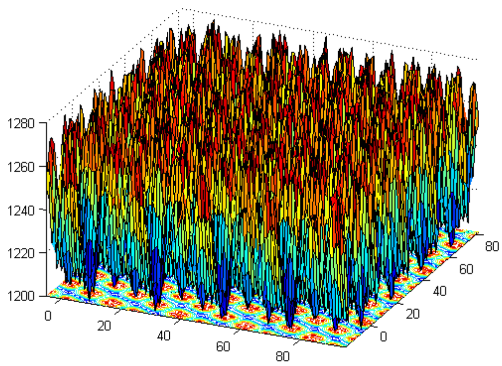

Figure 12(a). 3-*D* map for 2-*D* function

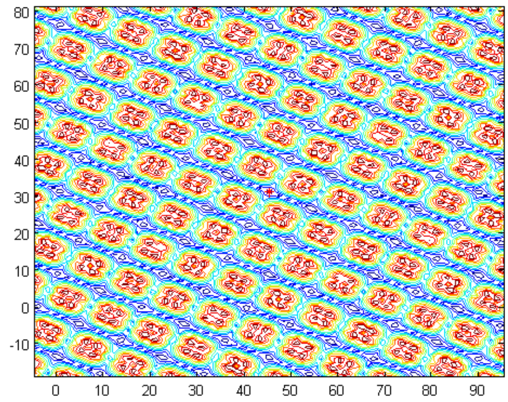

Figure 12(b).Contour map for 2-*D* function

Properties:

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

13) Shifted and Rotated HappyCat Function

$$F_{13}(\boldsymbol{x}) = f_{11}(\mathbf{M}(\frac{5(\boldsymbol{x}-\boldsymbol{o}_{13})}{100})) + F_{13}^{*} \tag{27}$$

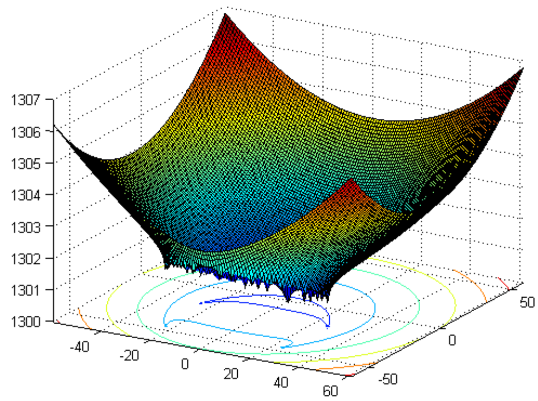

Figure 13(a). 3-*D* map for 2-*D* function

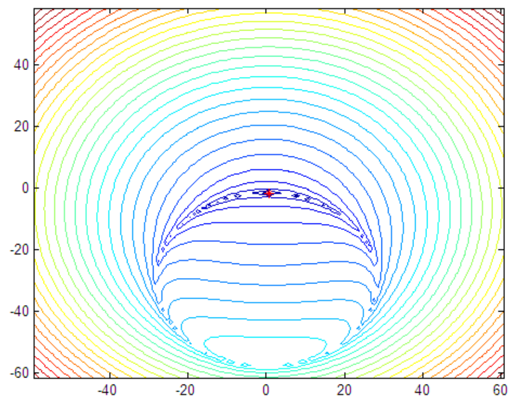

Figure 13(b).Contour map for 2-*D* function

Properties:

- Multi-modal
- Non-separable

14) Shifted and Rotated HGBat Function

$$F_{14}(\boldsymbol{x}) = f_{12}(\mathbf{M}(\frac{5(\boldsymbol{x}-\boldsymbol{o}_{14})}{100})) + F_{14}^* \quad (28)$$

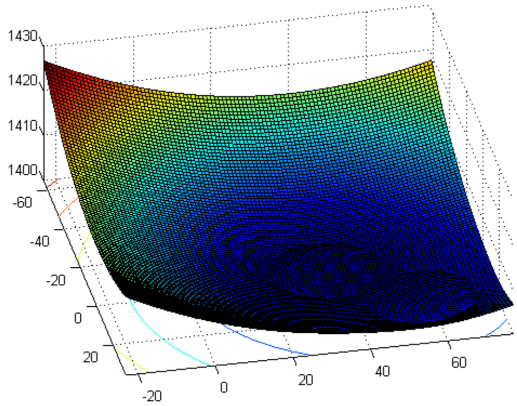

Figure 14(a). 3-*D* map for 2-*D* function

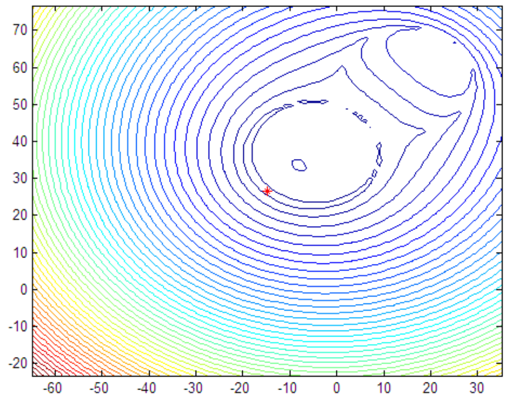

Figure 14(b).Contour map for 2-*D* function

Properties:

- Multi-modal
- Non-separable

15) Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function

$$F_{15}(\boldsymbol{x}) = f_{13}(\mathbf{M}(\frac{5(\boldsymbol{x}-\boldsymbol{o}_{15})}{100})+1) + F_{15}^* \quad (29)$$

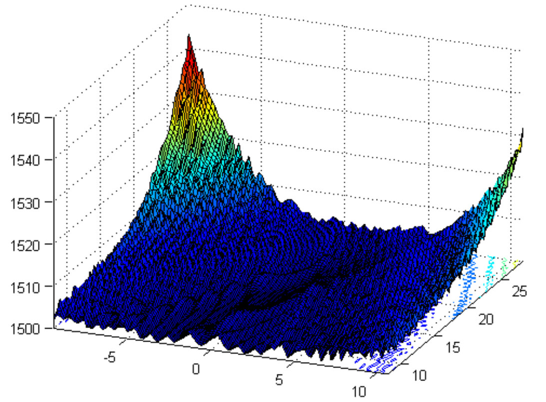

Figure 15(a). 3-*D* map for 2-*D* function

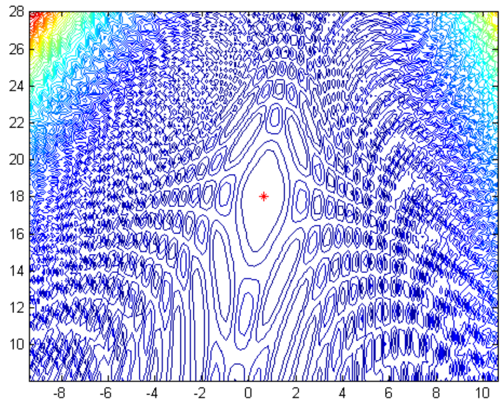

Figure 15(b).Contour map for 2-*D* function

Properties:

- Multi-modal
- Non-separable

16) Shifted and Rotated Expanded Scaffer's F6 Function

$$F_{16}(\boldsymbol{x}) = f_{14}(\mathbf{M}(\boldsymbol{x}-\boldsymbol{o}_{16})+1)+F_{16}^{*} \quad (30)$$

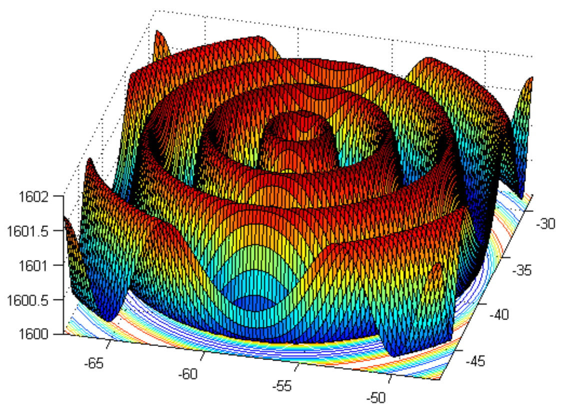

Figure 16. 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable

C. Hybrid Functions

Considering that in the real-world optimization problems, different subcomponents of the variables may have different properties[5]. In this set of hybrid functions, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents.

$$F(\boldsymbol{x}) = g_1(\mathbf{M}_1\boldsymbol{z}_1) + g_2(\mathbf{M}_2\boldsymbol{z}_2) + ... + g_N(\mathbf{M}_N\boldsymbol{z}_N) + F^*(\boldsymbol{x}) \tag{31}$$

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

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

N: number of basic functions

$$\boldsymbol{z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_N]$$

$$\boldsymbol{z}_1 = [\boldsymbol{y}_{S_1}, \boldsymbol{y}_{S_2}, ..., \boldsymbol{y}_{S_{n_1}}], \boldsymbol{z}_2 = [\boldsymbol{y}_{S_{n_1+1}}, \boldsymbol{y}_{S_{n_1+2}}, ..., \boldsymbol{y}_{S_{n_1+n_2}}], ..., \boldsymbol{z}_N = [\boldsymbol{y}_{S_{\sum_{i=1}^{N-1} n_i+1}}, \boldsymbol{y}_{S_{\sum_{k=1}^{N-1} n_i+2}}, ..., \boldsymbol{y}_{S_D}]$$

$$\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{o}_i, S = randperm(1:D)$$

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

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

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

Properties:

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

17) Hybrid Function 1

$N = 3$

$p = [0.3,0.3,0.4]$

g_1: Modified Schwefel's Function f_9

g_2: Rastrigin's Function f_8

g_3: High Conditioned Elliptic Function f_1

18) Hybrid Function 2

$N = 3$

p = [0.3, 0.3, 0.4]
g_1 : Bent Cigar Function f_2
g_2 : HGBat Function f_{12}
g_3: Rastrigin's Function f_8

19) Hybrid Function 3

N = 4
p = [0.2, 0.2, 0.3, 0.3]
g_1 : Griewank's Function f_7
g_2 : Weierstrass Function f_6
g_3: Rosenbrock's Function f_4
g_4: Scaffer's F6 Function:f_{14}

20) Hybrid Function 4

N = 4
p = [0.2, 0.2, 0.3, 0.3]
g_1 : HGBat Function f_{12}
g_2 : Discus Function f_3
g_3: Expanded Griewank's plus Rosenbrock's Function f_{13}
g_4: Rastrigin's Function f_8

21) Hybrid Function 5

N = 5
p = [0.1, 0.2, 0.2, 0.2, 0.3]
g_1 : Scaffer's F6 Function:f_{14}
g_2 : HGBat Function f_{12}
g_3: Rosenbrock's Function f_4
g_4: Modified Schwefel's Function f_9
g_5: High Conditioned Elliptic Function f_1

22) Hybrid Function 6

N = 5
p = [0.1, 0.2, 0.2, 0.2, 0.3]
g_1 : Katsuura Function f_{10}
g_2 : HappyCat Function f_{11}
g_3: Expanded Griewank's plus Rosenbrock's Function f_{13}
g_4: Modified Schwefel's Function f_9
g_5: Ackley's Function f_5

D. Composition Functions

$$F(\boldsymbol{x}) = \sum_{i=1}^{N} \{\omega_i * [\lambda_i g_i(\boldsymbol{x}) + bias_i]\} + F^* \tag{32}$$

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

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

N: number of basic functions

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

$bias_i$: defines which optimum is global optimum

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

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

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

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^{D}(x_j - o_{ij})^2}} \exp\left(-\frac{\sum_{j=1}^{D}(x_j - o_{ij})^2}{2D\sigma_i^2}\right) \tag{32}$$

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

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

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 $F_i' = F_i - F_i^*$ 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, 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: In order to test the algorithms' tendency to converge to the search centre, a local optimum is set to the origin as a trap for each composition functions included in this benchmark suite.

23) Composition Function 1

N= 5， σ = [10, 20, 30, 40, 50]

λ = [1, 1e-6, 1e-26, 1e-6, 1e-6]

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

g_1: Rotated Rosenbrock's Function F_4'

g_2: High Conditioned Elliptic Function F_1'

g_3 Rotated Bent Cigar Function F_2'

g_4: Rotated Discus Function F_3'

g_5: High Conditioned Elliptic Function F_1'

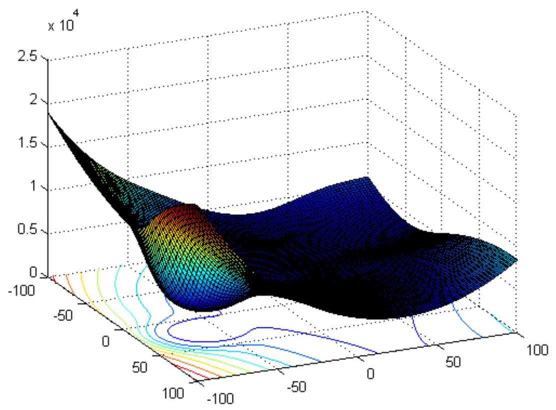

Figure 17(a). 3-*D* map for 2-*D* function

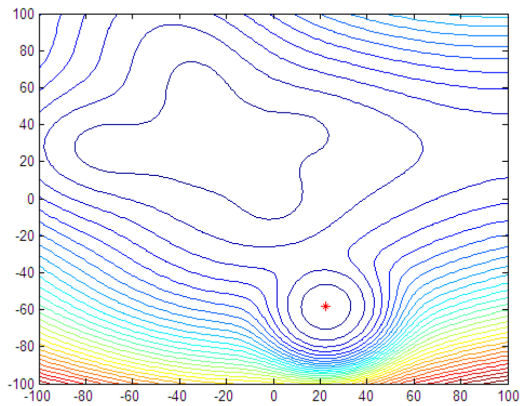

Figure 17 (b).Contour map for 2-*D* function

Properties:

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

24) Composition Function 2

N = 3

σ = [20, 20, 20]

λ = [1, 1, 1]

bias = [0, 100, 200]

$g_{1:}$ Schwefel's Function F_{10}'

$g_{2:}$ Rotated Rastrigin's Function F_9'

g_3 Rotated HGBat Function F_{14}'

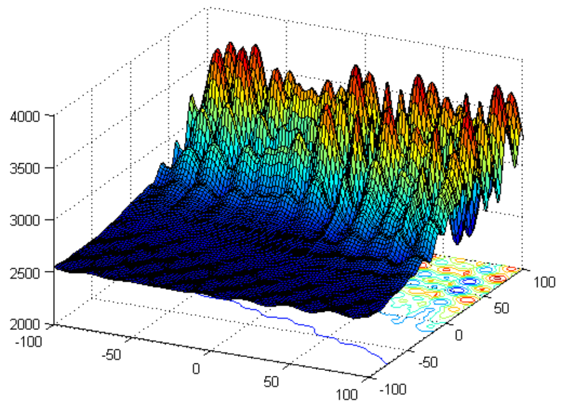

Figure 18(a). 3-*D* map for 2-*D* function

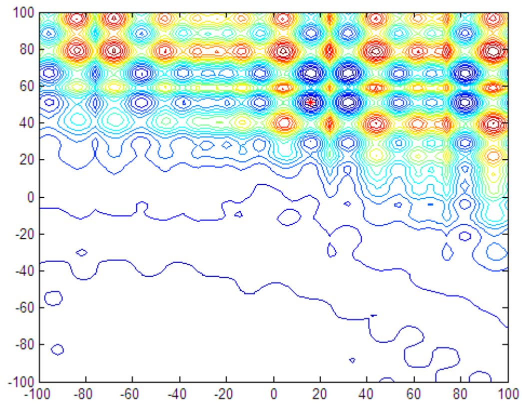

Figure 18(b).Contour map for 2-*D* function

Properties:

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

25) Composition Function 3

$N = 3$

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

$\lambda = [0.25, 1, 1e\text{-}7]$

$bias = [0, 100, 200]$

$g_{1:}$ Rotated Schwefel's Function F_{11}'

$g_{2:}$ Rotated Rastrigin's Function F_9'

$g_{3:}$ Rotated High Conditioned Elliptic Function F_1'

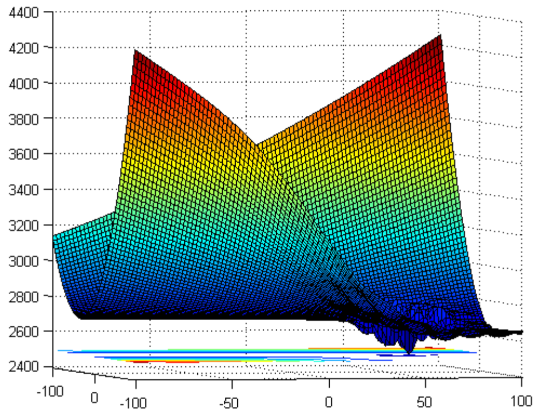

Figure 19(a). 3-*D* map for 2-*D* function

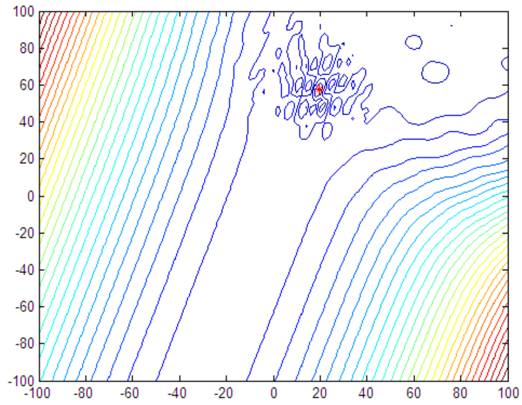

Figure 19(b).Contour map for 2-*D* function

Properties:

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

26) Composition Function 4

$N = 5$

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

$\lambda = [\ 0.25, 1, 1e\text{-}7, 2.5, 10]$

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

$g_{1:}$ Rotated Schwefel's Function F_{11}'

$g_{2:}$ Rotated HappyCat Function F_{13}'

$g_{3:}$ Rotated High Conditioned Elliptic Function F_1'

$g_{4:}$ Rotated Weierstrass Function F_6'

$g_{5:}$ Rotated Griewank's Function F_7'

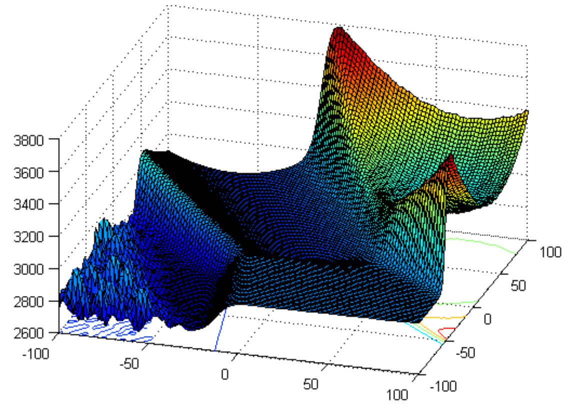

Figure 20(a). 3-*D* map for 2-*D* function

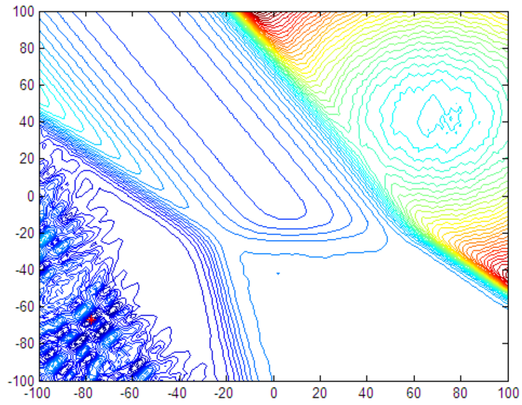

Figure 20(b).Contour map for 2-*D* function

Properties:

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

27) Composition Function 5

$N = 5$

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

$\lambda = [10, 10, 2.5, 25, 1e\text{-}6]$

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

$g_{1:}$ Rotated HGBat Function F_{14}’

$g_{2:}$ Rotated Rastrigin’s Function F_9’

$g_{3:}$ Rotated Schwefel's Function F_{11}’

$g_{4:}$ Rotated Weierstrass Function F_6’

$g_{5:}$ Rotated High Conditioned Elliptic Function F_1’

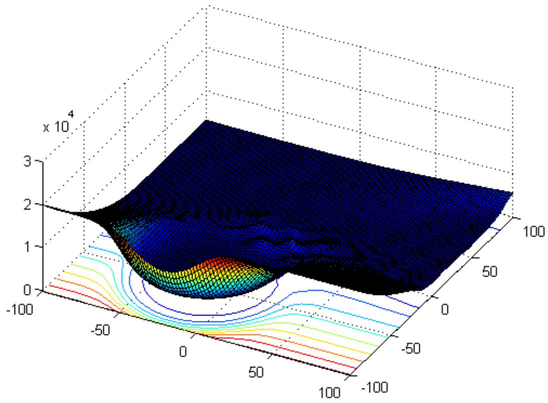

Figure 21(a). 3-*D* map for 2-*D* function

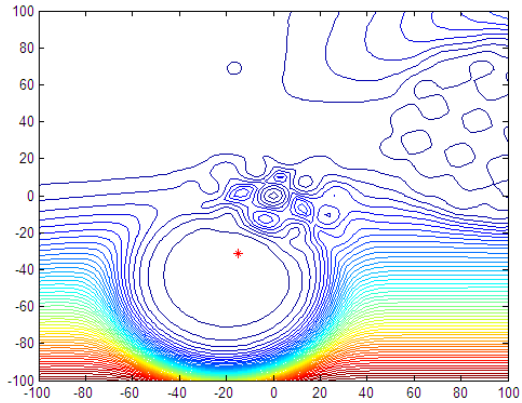

Figure 21(b).Contour map for 2-*D* function

Properties:

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

28) Composition Function 6

$N = 5$

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

$\lambda = [\ 2.5, 10, 2.5, 5e\text{-}4, 1e\text{-}6]$

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

g_1: Rotated Expanded Griewank's plus Rosenbrock's Function F_{15}'

g_2: Rotated HappyCat Function F_{13}'

g_3: Rotated Schwefel's Function F_{11}'

g_4: Rotated Expanded Scaffer's F6 Function F_{16}'

g_5: Rotated High Conditioned Elliptic Function F_1'

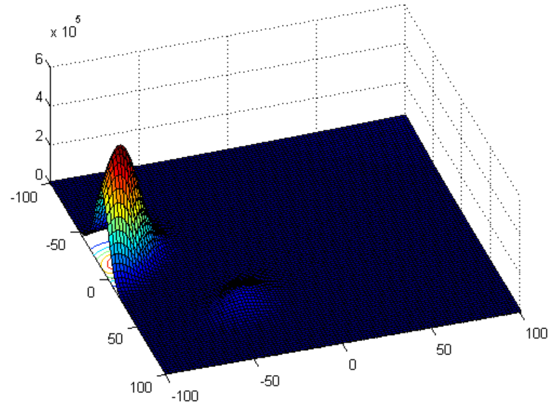

Figure 28(a). 3-*D* map for 2-*D* function

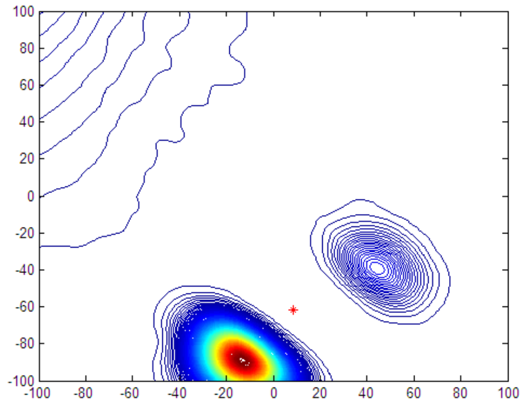

Figure 28(b).Contour map for 2-*D* function

Properties:

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

29) Composition Function 7

$N = 3$

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

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

$bias = [0, 100, 200]$

$g_{1:}$ Hybrid Function 1 F_{17}'

$g_{2:}$ Hybrid Function 2 F_{18}'

$g_{3:}$ Hybrid Function 3 F_{19}'

Properties:

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

30) Composition Function 8

$N = 3$

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

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

$bias = [0, 100, 200]$

$g_{1:}$ Hybrid Function 4 F_{20}'

$g_{2:}$ Hybrid Function 5 F_{21}'

$g_{3:}$ Hybrid Function 6 F_{22}'

Properties:

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

2. Evaluation Criteria

2.1 Experimental Setting

Problems: 30 minimization problems

Dimensions: D=10, 30, 50, 100 (Results only for 10D and 30D are acceptable for the initial submission; but 50D and 100D should be included in the final version)

Runs / problem: 51 (**Do not run many 51 runs to pick the best run**)

MaxFES: 10000*D (Max_FES for 10D = 100000; for 30D = 300000; for 50D = 500000; for 100D = 1000000)

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

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

2.1 Results Record

1) **Record function error value ($F_i(x)$-$F_i(x^*)$) after (0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0)*MaxFES for each run.**

   In this case, **14** error values are recorded for each function for each run. Sort the error values achieved after MaxFES in 51 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 51 runs.

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

2) **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 18**. 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 18** 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, 30 and 50 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, PSO results for test function 5 and D=30, the file name should be "PSO_5_30.txt".

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

Table II. Information Matrix for *D* Dimensional Function X

***.txt	Run 1	Run 2	…	Run 51
Function error values when FES=0.01*MaxFES				
Function error values when FES=0.02*MaxFES				
Function error values when FES=0.03*MaxFES				
Function error values when FES=0.05*MaxFES				
… …				
Function error values when FES=0.9*MaxFES				
Function error values when FES=MaxFES				

Thus **30*4** (10D, 30D, 50D and 100D) files should be zipped and sent to the organizers. Each file contains a **14*51** matrix.

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 Temple

Language: Matlab 2008a

Algorithm: Particle Swarm Optimizer (PSO)

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 (30*4 files described in section 2.2) to the organizers for a better comparison among the algorithms.

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					
25					
26					
27					
28					
29					
30					

Table IV. Results for 30D

…

Table V. Results for 50D

…

Table VI. Results for 100D

…

Algorithm Complexity

Table VII. Computational Complexity

	T0	*T1*	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
D=10				
D=30				
D=50				

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.

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", Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, January 2013.

[3] Joaquin Derrac, Salvador Garcia, Sheldon Hui, Francisco Herrera, Ponnuthurai N. Suganthan, "Statistical analysis of convergence performance throughout the search: A case study with SaDE-MMTS and Sa-EPSDE-MMTS," IEEE Symp. on Differential Evolution 2013, IEEE SSCI 2013, Singapore.

[4] Nikolaus Hansen, Steffen Finck, Raymond Ros and Anne Auger, "Real-Parameter Black-Box Optimization Benchmarking 2010: Noiseless Functions Definitions" INRIA research report RR-6829, March 24, 2012.

[5] 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.

[6] H.-G. Beyer and S. Finck, "HappyCat -- A Simple Function Class Where Well-Known Direct Search Algorithms Do Fail," In Proc. of Parallel Problem Solving from Nature~12, pp. 367-376, Ed by C. A. Coello Coello et al., Springer, Berlin, 2012.