Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization

Article *in* Natural Computing · January 2005

7 authors, including:


Ponnuthurai N. Suganthan
Qatar University


Nikolaus Hansen
Inria (National Institute for Research in Computer Science and Control)


Jing Liang
Zhengzhou University


Kalyan Deb
Michigan State University

Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization

P. N. Suganthan[1], N. Hansen[2], J. J. Liang[1], K. Deb[3], Y. -P. Chen[4], A. Auger[2], S. Tiwari[3]

[1]School of EEE, Nanyang Technological University, Singapore, 639798
[2](ETH) Z¨urich, Switzerland
[3]Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, PIN 208 016, India
[4]Natural Computing Laboratory, Department of Computer Science, National Chiao Tung University, Taiwan

epnsugan@ntu.edu.sg, Nikolaus.Hansen@inf.ethz.ch, liangjing@pmail.ntu.edu.sg, deb@iitk.ac.in, ypchen@csie.nctu.edu.tw, Anne.Auger@inf.ethz.ch, tiwaris@iitk.ac.in

Technical Report, Nanyang Technological University, Singapore

And

KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur)

May 2005

Acknowledgement: We also acknowledge the contributions by Drs / Professors Maurice Clerc (Maurice.Clerc@WriteMe.com), Bogdan Filipic (bogdan.filipic@ijs.si), William Hart (wehart@sandia.gov), Marc Schoenauer (Marc.Schoenauer@lri.fr), Hans-Paul Schwefel (hans-paul.schwefel@cs.uni-dortmund.de), Aristin Pedro Ballester (p.ballester@imperial.ac.uk) and Darrell Whitley (whitley@CS.ColoState.EDU) .

Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization

In the past two decades, different kinds of optimization algorithms have been designed and applied to solve real-parameter function optimization problems. Some of the popular approaches are real-parameter EAs, evolution strategies (ES), differential evolution (DE), particle swarm optimization (PSO), evolutionary programming (EP), classical methods such as quasi-Newton method (QN), hybrid evolutionary-classical methods, other non-evolutionary methods such as simulated annealing (SA), tabu search (TS) and others. Under each category, there exist many different methods varying in their operators and working principles, such as correlated ES and CMA-ES. In most such studies, a subset of the standard test problems (Sphere, Schwefel's, Rosenbrock's, Rastrigin's, etc.) is considered. Although some comparisons are made in some research studies, often they are confusing and limited to the test problems used in the study. In some occasions, the test problem and chosen algorithm are complementary to each other and the same algorithm may not work in other problems that well. There is definitely a need of evaluating these methods in a more systematic manner by specifying a common termination criterion, size of problems, initialization scheme, linkages/rotation, etc. There is also a need to perform a scalability study demonstrating how the running time/evaluations increase with an increase in the problem size. We would also like to include some real world problems in our standard test suite with codes/executables.

In this report, 25 benchmark functions are given and experiments are conducted on some real-parameter optimization algorithms. The codes in Matlab, C and Java for them could be found at http://www.ntu.edu.sg/home/EPNSugan/. The mathematical formulas and properties of these functions are described in Section 2. In Section 3, the evaluation criteria are given. Some notes are given in Section 4.

1. Summary of the 25 CEC'05 Test Functions

- **Unimodal Functions (5):**
  - F_1: Shifted Sphere Function
  - F_2: Shifted Schwefel's Problem 1.2
  - F_3: Shifted Rotated High Conditioned Elliptic Function
  - F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness
  - F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds

- **Multimodal Functions (20):**
  - **Basic Functions** (7):
    - F_6: Shifted Rosenbrock's Function
    - F_7: Shifted Rotated Griewank's Function without Bounds
    - F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds
    - F_9: Shifted Rastrigin's Function
    - F_{10}: Shifted Rotated Rastrigin's Function
    - F_{11}: Shifted Rotated Weierstrass Function
    - F_{12}: Schwefel's Problem 2.13
  - **Expanded Functions** (2):

- F_{13}: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)
- F_{14}: Shifted Rotated Expanded Scaffer's F6

- **Hybrid Composition Functions** (11):
  - F_{15}: Hybrid Composition Function
  - F_{16}: Rotated Hybrid Composition Function
  - F_{17}: Rotated Hybrid Composition Function with Noise in Fitness
  - F_{18}: Rotated Hybrid Composition Function
  - F_{19}: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum
  - F_{20}: Rotated Hybrid Composition Function with the Global Optimum on the Bounds
  - F_{21}: Rotated Hybrid Composition Function
  - F_{22}: Rotated Hybrid Composition Function with High Condition Number Matrix
  - F_{23}: Non-Continuous Rotated Hybrid Composition Function
  - F_{24}: Rotated Hybrid Composition Function
  - F_{25}: Rotated Hybrid Composition Function without Bounds

- **Pseudo-Real Problems:** Available from `http://www.cs.colostate.edu/~genitor/functions.html`. If you have any queries on these problems, please contact Professor Darrell Whitley. Email: whitley@CS.ColoState.EDU

2. Definitions of the 25 CEC'05 Test Functions

2.1 Unimodal Functions:

2.1.1. F_1: *Shifted Sphere Function*

$$F_1(\mathbf{x}) = \sum_{i=1}^{D} z_i^2 + f\_bias_1 \, , \mathbf{z} = \mathbf{x} - \mathbf{o} \, , \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions. $\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum.

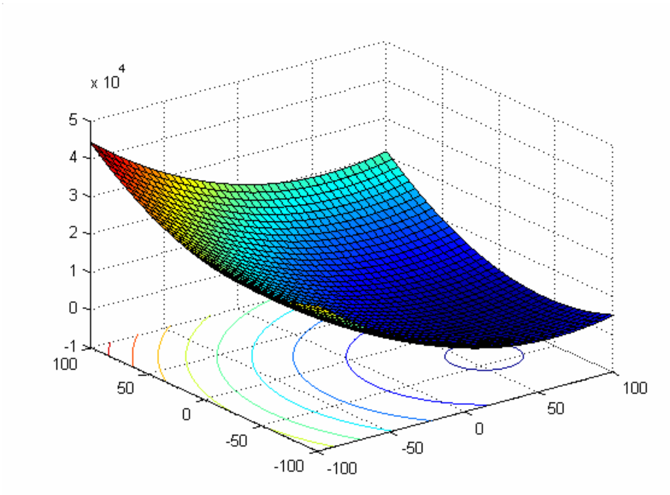

Figure 2-1 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Shifted
- Separable
- Scalable
- $\mathbf{x} \in [-100,100]^D$, Global optimum: $\mathbf{x}^* = \mathbf{o}$, $F_1(\mathbf{x}^*) = f\_bias_1 = -450$

Associated Data files:

Name:	sphere_func_data.mat sphere_func_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		
Name:	fbias_data.mat fbias_data.txt		
Variable:	**f_bias**	1*25 vector, record all the 25 function's f_bias_i	

2.1.2. F_2: *Shifted Schwefel's Problem 1.2*

$$F_2(\mathbf{x}) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 + f\_bias_2,\ \mathbf{z} = \mathbf{x} - \mathbf{o},\ \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

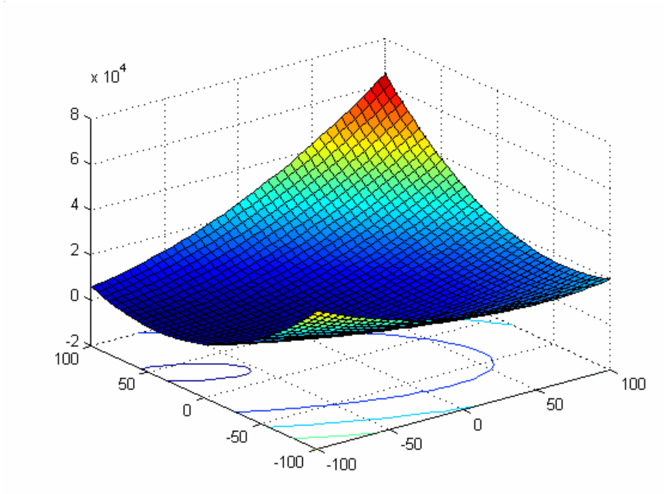

Figure 2-2 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Shifted
- Non-separable
- Scalable
- $\mathbf{x} \in [-100,100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_2(\mathbf{x}^*) = f\_bias_2 = -450$

Associated Data files:

Name:	schwefel_102_data.mat		
	schwefel_102_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		

2.1.3. F_3: *Shifted Rotated High Conditioned Elliptic Function*

$$F_3(\mathbf{x}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} z_i^2 + f\_bias_3, \ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}, \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

M: orthogonal matrix

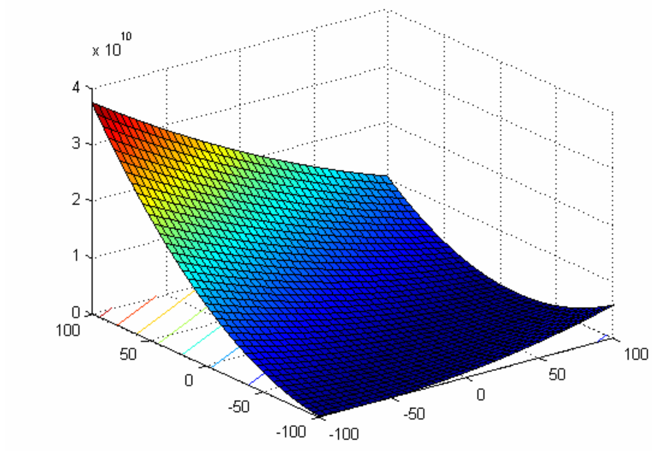

Figure 2-3 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Shifted
- Rotated
- Non-separable
- Scalable
- $\mathbf{x} \in [-100, 100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_3(\mathbf{x}^*) = f\_bias_3$ =- 450

Associated Data files:

Name:	high_cond_elliptic_rot_data.mat		
	high_cond_elliptic_rot_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		
Name:	elliptic_M_D10 .mat		elliptic_M_D10 .txt
Variable:	**M**	10*10 matrix	
Name:	elliptic_M_D30 .mat		elliptic_M_D30 .txt
Variable:	**M**	30*30 matrix	
Name:	elliptic_M_D50 .mat		elliptic_M_D50 .txt
Variable:	**M**	50*50 matrix	

2.1.4. F_4: *Shifted Schwefel's Problem 1.2 with Noise in Fitness*

$$F_4(\mathbf{x}) = (\sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2) * (1 + 0.4|N(0,1)|) + f\_bias_4,\ \mathbf{z} = \mathbf{x} - \mathbf{o},\ \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

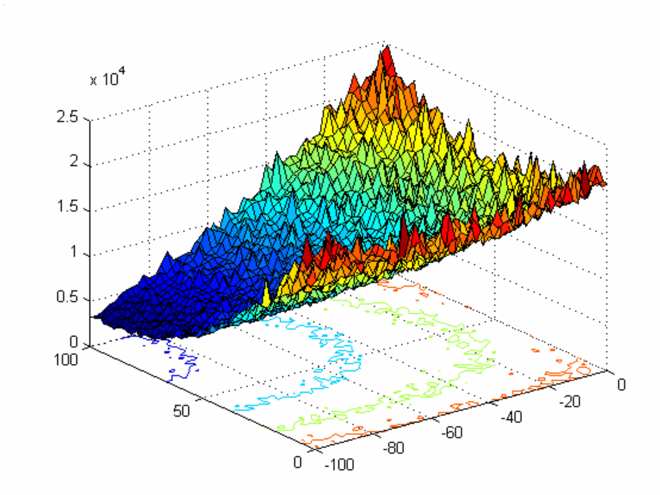

Figure 2-4 3-*D* map for 2-*D* function

Properties:

- Unimodal
- Shifted
- Non-separable
- Scalable
- Noise in fitness
- $\mathbf{x} \in [-100,100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_4(\mathbf{x}^*) = f\_bias_4 = -450$

Associated Data file:

Name:	schwefel_102_data.mat		
	schwefel_102_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		

2.1.5. F_5: *Schwefel's Problem 2.6 with Global Optimum on Bounds*

$f(\mathbf{x}) = \max\{|x_1 + 2x_2 - 7|, |2x_1 + x_2 - 5|\}, i = 1,...,n$, $\mathbf{x}^* = [1,3]$, $f(\mathbf{x}^*) = 0$

Extend to D dimensions:

$F_5(\mathbf{x}) = \max\{|\mathbf{A}_i\mathbf{x} - \mathbf{B}_i|\} + f\_bias_5, i = 1,...,D$, $\mathbf{x} = [x_1, x_2,..., x_D]$

D: dimensions

A is a D*D matrix, a_{ij} are integer random numbers in the range [*-500, 500*], $\det(\mathbf{A}) \neq 0$, $\mathbf{A}_i$ is the i^{th} row of **A**.

$\mathbf{B}_i = \mathbf{A}_i * \mathbf{o}$, **o** is a D*1 vector, o_i are random number in the range [-100,100]

After load the data file, set $o_i = -100$, for $i = 1,2,...,\lceil D/4 \rceil$, $o_i = 100$, for $i = \lfloor 3D/4 \rfloor,...,D$

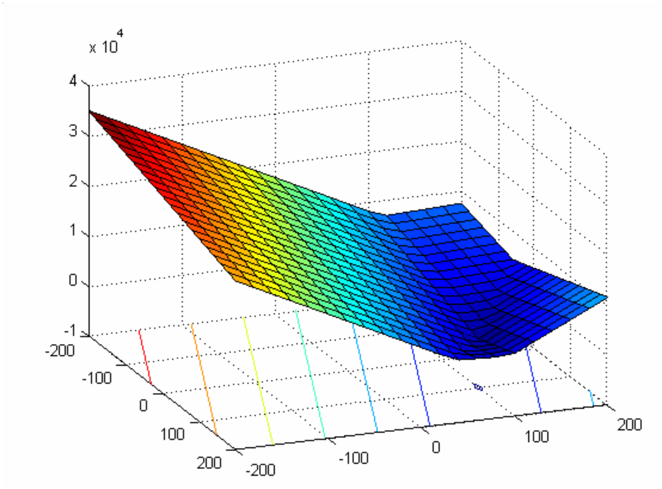

Figure 2-5 3-D map for 2-D function

Properties:

- Unimodal
- Non-separable
- Scalable
- If the initialization procedure initializes the population at the bounds, this problem will be solved easily.
- $\mathbf{x} \in [-100,100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_5(\mathbf{x}^*) = f\_bias_5 = -310$

Associated Data file:

Name:	schwefel_206_data.mat		
	schwefel_206_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	A	100*100 matrix	

When using, cut **o**=**o**(1:D) **A**=**A**(1:D,1:D)

In schwefel_206_data.txt ,the first line is **o** (1*100 vector),and line2-line101 is **A**(100*100 matrix)

2.2 Basic Multimodal Functions

2.2.1. F_6: *Shifted Rosenbrock's Function*

$$F_6(\mathbf{x}) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f\_bias_6 , \ \mathbf{z} = \mathbf{x} - \mathbf{o} + 1, \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

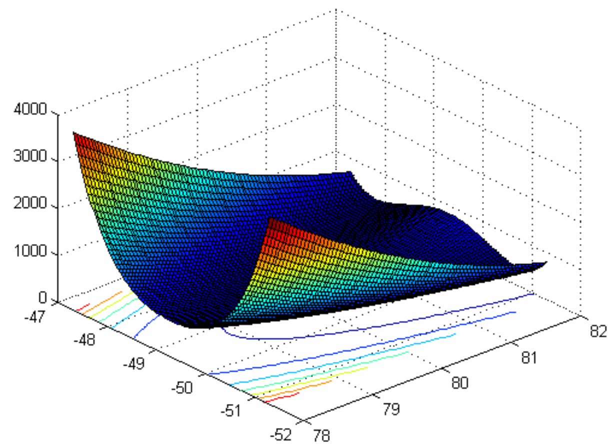

Figure 2-6 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Non-separable
- Scalable
- Having a very narrow valley from local optimum to global optimum
- $\mathbf{x} \in [-100,100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_6(\mathbf{x}^*) = f\_bias_6 = 390$

Associated Data file:

Name:	rosenbrock_func_data.mat		
	rosenbrock_func_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		

2.2.2. F_7: *Shifted Rotated Griewank's Function without Bounds*

$$F_7(\mathbf{x}) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias_7 \ , \ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M} , \ \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

M': linear transformation matrix, condition number=3

M =**M'**(1+0.3|N(0,1)|)

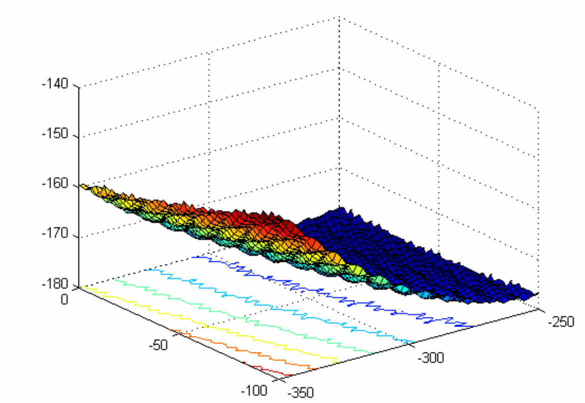

Figure 2-7 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Shifted
- Non-separable
- Scalable
- No bounds for variables *x*
- Initialize population in $[0, 600]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$ is outside of the initialization range, $F_7(\mathbf{x}^*) = f\_bias_7 = -180$

Associated Data file:

Name:	griewank_func_data.mat	griewank_func_data.txt
Variable:	**o** 1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)	
Name:	griewank_M_D10 .mat	griewank_M_D10 .txt
Variable:	**M** 10*10 matrix	
Name:	griewank_M_D30 .mat	griewank_M_D30 .txt
Variable:	**M** 30*30 matrix	
Name:	griewank_M_D50 .mat	griewank_M_D50 .txt
Variable:	**M** 50*50 matrix	

2.2.3. F_8: *Shifted Rotated Ackley's Function with Global Optimum on Bounds*

$$F_8(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e + f\_bias_8, \quad \mathbf{z} = (\mathbf{x}-\mathbf{o})*\mathbf{M},$$

$\mathbf{x} = [x_1, x_2, ..., x_D]$, *D:* dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum;

After load the data file, set $o_{2j-1} = -32$ o_{2j} are randomly distributed in the search range, for $j = 1, 2, ..., \lfloor D/2 \rfloor$

M: linear transformation matrix, condition number=100

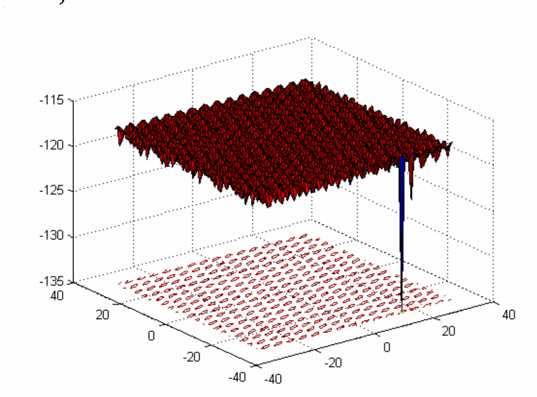

Figure 2-8 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Shifted
- Non-separable
- Scalable
- **A**'s condition number Cond(**A**) increases with the number of variables as $O(D^2)$
- Global optimum on the bound
- If the initialization procedure initializes the population at the bounds, this problem will be solved easily.
- $\mathbf{x} \in [-32, 32]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_8(\mathbf{x}^*) = f\_bias_8 = -140$

Associated Data file:

Name:	ackley_func_data.mat		ackley_func_data.txt
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		

Name:	ackley_M_D10 .mat	ackley_M_D10 .txt
Variable:	**M** 10*10 matrix	
Name:	ackley_M_D30 .mat	ackley_M_D30 .txt
Variable:	**M** 30*30 matrix	
Name:	ackley_M_D50 .mat	ackley_M_D50 .txt
Variable:	**M** 50*50 matrix	

2.2.4. F_9: *Shifted Rastrigin's Function*

$$F_9(\mathbf{x}) = \sum_{i=1}^{D}(z_i^{\,2} - 10\cos(2\pi z_i) + 10) + f\_bias_9, \ \mathbf{z} = \mathbf{x} - \mathbf{o}, \ \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\boldsymbol{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

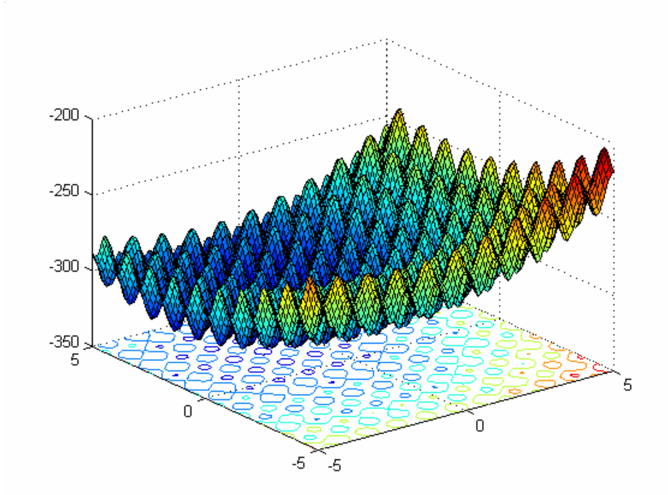

Figure 2-9 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Separable
- Scalable
- Local optima's number is huge
- $\mathbf{x} \in [-5, 5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_9(\boldsymbol{x}^*) = f\_bias_9 = -330$

Associated Data file:

Name:	rastrigin_func_data.mat rastrigin_func_data.txt		
Variable:	**o**	1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)		

2.2.5. F_{10}: *Shifted Rotated Rastrigin's Function*

$$F_{10}(\mathbf{x}) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias_{10}, \ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}, \mathbf{x} = [x_1, x_2, ..., x_D]$$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

M: linear transformation matrix, condition number=2

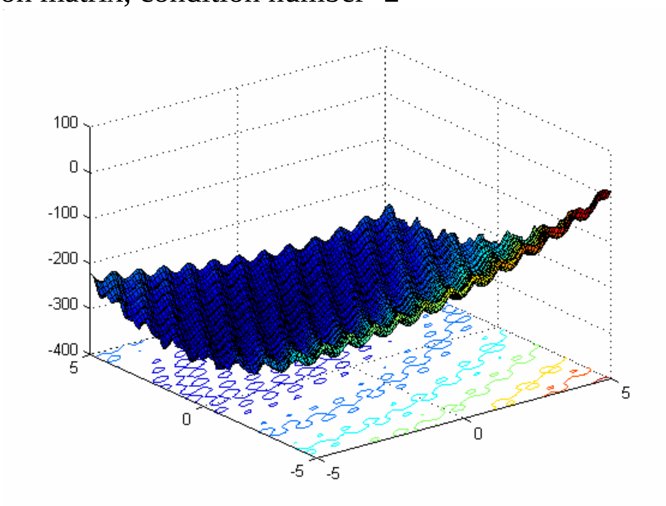

Figure 2-10 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Rotated
- Non-separable
- Scalable
- Local optima's number is huge
- $\mathbf{x} \in [-5, 5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_{10}(\mathbf{x}^*) = f\_bias_{10} = -330$

Associated Data file:

Name:	rastrigin_func_data.mat rastrigin_func_data.txt		
Variable:	**o**	1*100 vector When using, cut **o**=**o**(1:*D*)	the shifted global optimum
Name:	rastrigin_M_D10 .mat		rastrigin_M_D10 .txt
Variable:	**M**	10*10 matrix	
Name:	rastrigin_M_D30 .mat		rastrigin_M_D30 .txt
Variable:	**M**	30*30 matrix	
Name:	rastrigin_M_D50 .mat		rastrigin_M_D50 .txt
Variable:	**M**	50*50 matrix	

2.2.6. F_{11}: *Shifted Rotated Weierstrass Function*

$$F_{11}(\mathbf{x}) = \sum_{i=1}^{D} \left(\sum_{k=0}^{k\max} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - D \sum_{k=0}^{k\max} [a^k \cos(2\pi b^k \cdot 0.5)] + f\_bias_{11},$$

a=0.5, b=3, k_{max}=20, $\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}$, $\mathbf{x} = [x_1, x_2, ..., x_D]$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

M: linear transformation matrix, condition number=5

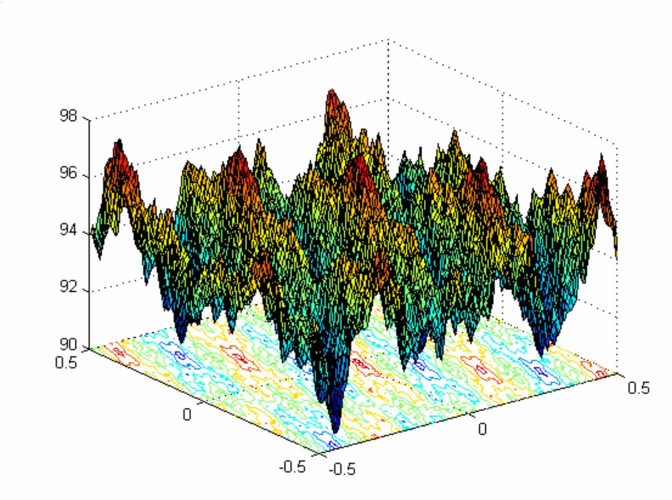

Figure 2-11 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Rotated
- Non-separable
- Scalable
- Continuous but differentiable only on a set of points
- $\mathbf{x} \in [-0.5, 0.5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_{11}(\mathbf{x}^*) = f\_bias_{11} = 90$

Associated Data file:

Name:	weierstrass_data.mat	weierstrass_data.txt
Variable:	**o** 1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)	
Name:	weierstrass_M_D10 .mat	weierstrass_M_D10 .txt
Variable:	**M** 10*10 matrix	
Name:	weierstrass_M_D30 .mat	weierstrass_M_D30 .txt
Variable:	**M** 30*30 matrix	
Name:	weierstrass_M_D50 .mat	weierstrass_M_D50 .txt
Variable:	**M** 50*50 matrix	

2.2.7. F_{12}: *Schwefel's Problem 2.13*

$$F_{12}(\mathbf{x}) = \sum_{i=1}^{D} (\mathbf{A}_i - \mathbf{B}_i(x))^2 + f\_bias_{12}, \mathbf{x} = [x_1, x_2, ..., x_D]$$

$$\mathbf{A}_i = \sum_{j=1}^{D} (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j), \mathbf{B}_i(x) = \sum_{j=1}^{D} (a_{ij} \sin x_j + b_{ij} \cos x_j), \text{ for } i = 1, ..., D$$

D: dimensions

A, **B** are two *D*D* matrix, a_{ij}, b_{ij} are integer random numbers in the range [-100,100],

$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_D], \alpha_j$ are random numbers in the range $[-\pi, \pi]$.

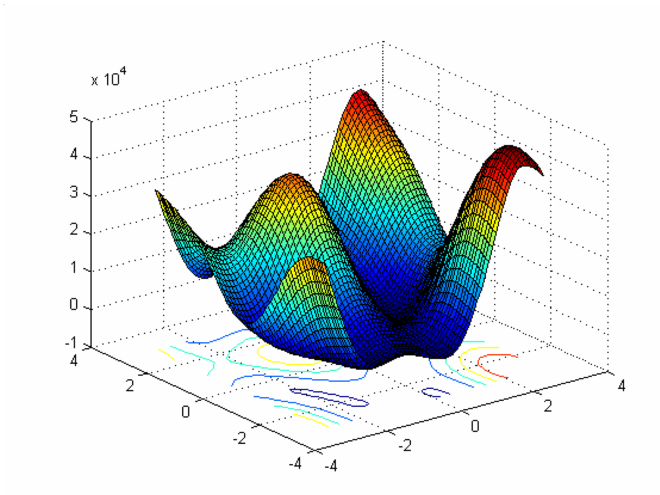

Figure 2-12 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Non-separable
- Scalable
- $\mathbf{x} \in [-\pi, \pi]^D$, Global optimum $\mathbf{x}^* = \boldsymbol{\alpha}$, $F_{12}(\mathbf{x}^*) = f\_bias_{12} = -460$

Associated Data file:

Name: schwefel_213_data.mat
schwefel_213_data.txt

Variable: **alpha** 1*100 vector the shifted global optimum
a 100*100 matrix
b 100*100 matrix
When using, cut **alpha**=**alpha**(1:*D*) **a**=**a**(1:*D*,1:*D*) **b**=**b**(1:*D*,1:*D*)
In schwefel_213_data.txt, and line1-line100 is **a** (100*100 matrix),and line101-line200 is **b** (100*100 matrix), the last line is **alpha**(1*100 vector),

2.3 Expanded Functions

Using a 2-*D* function $F(x, y)$ as a starting function, corresponding expanded function is:

$$EF(x_1, x_2, ..., x_D) = F(x_1, x_2) + F(x_2, x_3) + ... + F(x_{D-1}, x_D) + F(x_D, x_1)$$

2.3.1. F_{13}: *Shifted Expanded Griewank's plus Rosenbrock's Function (F8F2)*

F8: Griewank's Function: $F8(\boldsymbol{x}) = \sum_{i=1}^{D} \frac{x_i^{\,2}}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$

F2: Rosenbrock's Function: $F2(\boldsymbol{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$

$$F8F2(x_1, x_2, ..., x_D) = F8(F2(x_1, x_2)) + F8(F2(x_2, x_3)) + ... + F8(F2(x_{D-1}, x_D)) + F8(F2(x_D, x_1))$$

Shift to

$$F_{13}(\mathbf{x}) = F8(F2(z_1, z_2)) + F8(F2(z_2, z_3)) + ... + F8(F2(z_{D-1}, z_D)) + F8(F2(z_D, z_1)) + f\_bias_{13}$$

$\mathbf{z} = \mathbf{x} - \mathbf{o} + 1$ **,** $\mathbf{x} = [x_1, x_2, ..., x_D]$

D: dimensions $\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

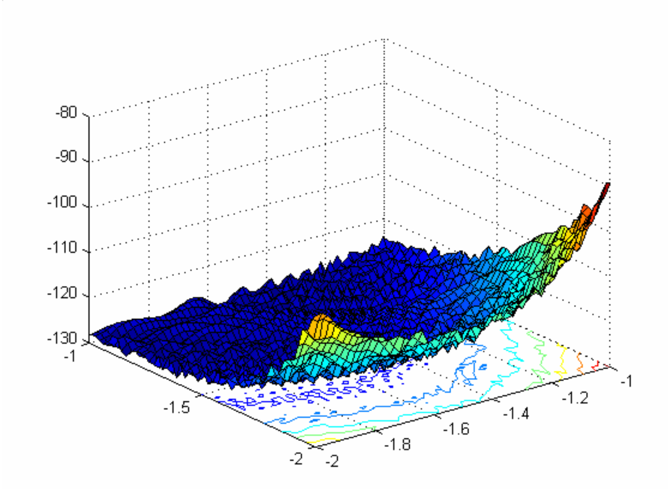

Figure 2-13 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Non-separable
- Scalable
- $\mathbf{x} \in [-3,1]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_{13}(\mathbf{x}^*) = f\_bias_{13}(13) = -130$

Associated Data file:

Name: EF8F2_func_data.mat
EF8F2_func_data.txt

Variable: **o** 1*100 vector the shifted global optimum
When using, cut **o**=**o**(1:*D*)

2.3.2. F_{14}: *Shifted Rotated Expanded Scaffer's F6 Function*

$$F(x,y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$$

Expanded to

$$F_{14}(\mathbf{x}) = EF(z_1, z_2, ..., z_D) = F(z_1, z_2) + F(z_2, z_3) + ... + F(z_{D-1}, z_D) + F(z_D, z_1) + f\_bias_{14},$$

$\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}$, $\mathbf{x} = [x_1, x_2, ..., x_D]$

D: dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$: the shifted global optimum

M: linear transformation matrix, condition number=3

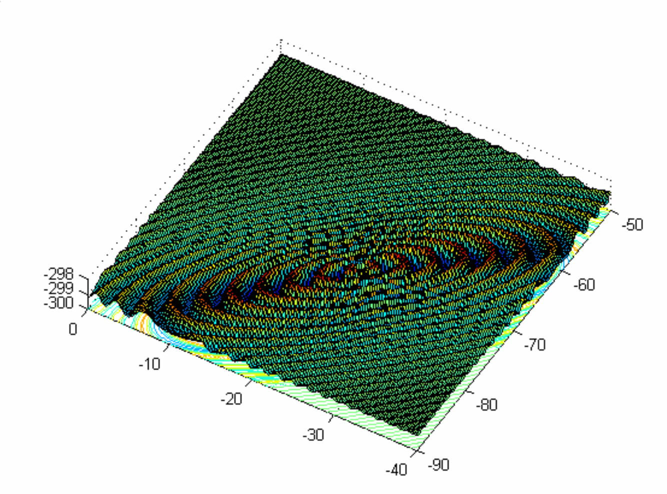

Figure 2-14 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Shifted
- Non-separable
- Scalable
- $\mathbf{x} \in [-100,100]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}$, $F_{14}(\mathbf{x}^*) = f\_bias_{14}$ (14)= -300

Associated Data file:

Name:	E_ScafferF6_func_data.mat	E_ScafferF6_func_data.txt
Variable:	**o** 1*100 vector	the shifted global optimum
	When using, cut **o**=**o**(1:*D*)	
Name:	E_ScafferF6_M_D10 .mat	E_ScafferF6_M_D10 .txt
Variable:	**M** 10*10 matrix	
Name:	E_ScafferF6_M_D30 .mat	E_ScafferF6_M_D30 .txt
Variable:	**M** 30*30 matrix	
Name:	E_ScafferF6_M_D50 .mat	E_ScafferF6_M_D50 .txt
Variable:	**M** 50*50 matrix	

2.4 Composition functions

$F(\mathbf{x})$: new composition function
$f_i(\mathbf{x})$: i^th^ basic function used to construct the composition function
n : number of basic functions
D : dimensions
$\mathbf{M}_i$: linear transformation matrix for each $f_i(\boldsymbol{x})$
$\mathbf{o}_i$: new shifted optimum position for each $f_i(\boldsymbol{x})$

$$F(\mathbf{x}) = \sum_{i=1}^{n} \{ w_i * [f_i '((\mathbf{x} - \mathbf{o}_i) / \lambda_i * \mathbf{M}_i) + bias_i] \} + f\_bias$$

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

$$w_i = \exp\left(-\frac{\sum_{k=1}^{D} (x_k - o_{ik})^2}{2D\sigma_i^{\ 2}}\right),$$

$$w_i = \begin{cases} w_i & w_i == \max(w_i) \\ w_i *(1\text{-}\max(w_i).\text{^}10) & w_i \neq \max(w_i) \end{cases}$$

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

σ_i : used to control each $f_i(\mathbf{x})$'s coverage range, a small σ_i give a narrow range for that $f_i(\mathbf{x})$
λ_i : used to stretch compress the function, λ_i >1 means stretch, λ_i <1 means compress
$\mathbf{o}_i$ define the global and local optima's position, $bias_i$ define which optimum is global optimum. Using $\mathbf{o}_i$, $bias_i$, a global optimum can be placed anywhere.

If $f_i(\mathbf{x})$ are different functions, different functions have different properties and height, in order to get a better mixture, estimate a biggest function value $f_{\max i}$ for 10 functions $f_i(\mathbf{x})$, then normalize each basic functions to similar heights as below:
$f_i'(\mathbf{x}) = C * f_i(\mathbf{x}) / |f_{\max i}|$, C is a predefined constant.
$|f_{\max i}|$ is estimated using $|f_{\max i}| = f_i((\mathbf{x}'/\lambda_i) * \mathbf{M}_i)$, $\mathbf{x}'$=[5,5…,5].

In the following composition functions,
Number of basic functions n=10.
D: dimensions
o: n*D matrix, defines $f_i(\mathbf{x})$'s global optimal positions
bias=[0, 100, 200, 300, 400, 500, 600, 700, 800, 900]. Hence, the first function $f_1(\mathbf{x})$ always the function with the global optimum.
C=2000

Pseudo Code:

Define f1-f10, σ , λ , bias, C, load data file o and rotated linear transformation matrix **M1-M10** $\mathbf{y}$ =[5,5…,5].

For i=1:10

$$w_i = \exp(-\frac{\sum_{k=1}^{D}(x_k - o_{ik})^2}{2D\sigma_i^{\ 2}}),$$

$$fit_i = f_i(((\mathbf{x}-\mathbf{o}_i)/\lambda_i)*\mathbf{M}_i)$$

$$f\max_i = f_i((\mathbf{y}/\lambda_i)*\mathbf{M}_i),$$

$$fit_i = C * fit_i \,/\, f\max_i$$

EndFor

$$SW = \sum_{i=1}^{n} w_i$$

$$MaxW = \max(w_i)$$

For i=1:10

$$w_i = \begin{cases} w_i & if \quad w_i == MaxW \\ w_i*(1\text{-}MaxW.\hat{}10) & if \quad w_i \neq MaxW \end{cases}$$

$$w_i = w_i / SW$$

EndFor

$$F(\mathbf{x}) = \sum_{i=1}^{n} \{w_i * [fit_i + bias_i]\}$$

$$F(\mathbf{x}) = F(x) + f\_bias$$

2.4.1. F_{15}: *Hybrid Composition Function*

$f_{1-2}(\mathbf{x})$: Rastrigin's Function

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

$f_{3-4}(\mathbf{x})$: Weierstrass Function

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

a=0.5, b=3, k_{max}=20

$f_{5-6}(\mathbf{x})$: Griewank's Function

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

$f_{7-8}(\mathbf{x})$: Ackley's Function

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

$f_{9-10}(\mathbf{x})$: Sphere Function

$$f_i(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$$

$\sigma_i = 1$ for $i = 1, 2, ..., D$

λ = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]

$\mathbf{M}_i$ are all identity matrices

Please notice that these formulas are just for the basic functions, no shift or rotation is included in these expressions. x here is just a variable in a function.

Take f_1 as an example, when we calculate $f_1(((\mathbf{x}-\mathbf{o}_1)/\lambda_1)*\mathbf{M}_1)$, we need calculate $f_1(\mathbf{z}) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10)$, $\mathbf{z} = ((\mathbf{x}-\mathbf{o}_1)/\lambda_1)*\mathbf{M}_1$.

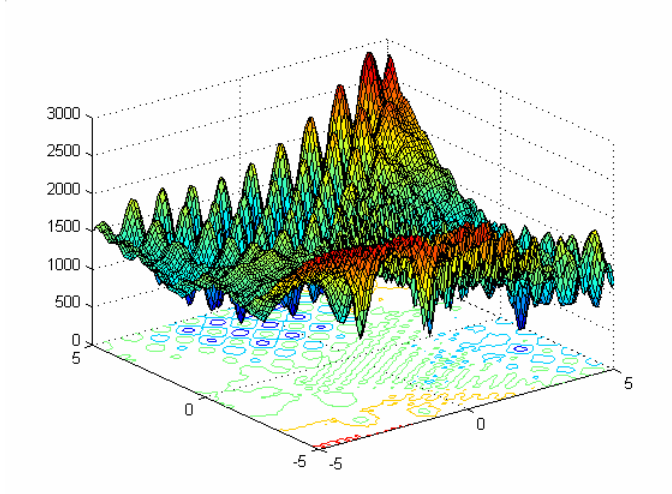

Figure 2-15 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Separable near the global optimum (Rastrigin)
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Sphere Functions give two flat areas for the function
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{15}(\mathbf{x}^*) = f\_bias_{15} = 120$

Associated Data file:

Name: hybrid_func1_data.mat
hybrid_func1_data.txt

Variable: **o** 10*100 vector the shifted optimum for 10 functions
When using, cut **o**=**o**(:,1:*D*)

2.4.2. F_{16}: *Rotated Version of Hybrid Composition Function* F_{15}

Except $\mathbf{M}_i$ are different linear transformation matrixes with condition number of 2, all other settings are the same as F_{15}.

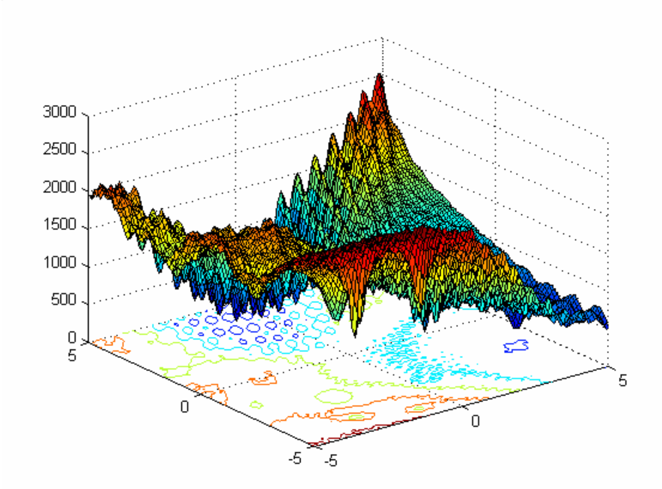

Figure 2-16 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Non-Separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Sphere Functions give two flat areas for the function.
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{16}(\mathbf{x}^*) = f\_bias_{16}$ =120

Associated Data file:

Name: hybrid_func1_data.mat
hybrid_func1_data.txt
Variable: **o** 10*100 vector the shifted optima for 10 functions
When using, cut **o**=**o**(:,1:*D*)

Name: hybrid_func1_M_D10 .mat
Variable: **M** an structure variable
Contains **M.M1 M.M2**, … , **M.M10** ten 10*10 matrixes
Name: hybrid_func1_M_D10 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 10*10 matrixes, 1-10 lines are **M1**, 11-20 lines are **M2**,....,91-100 lines are **M10**

Name: hybrid_func1_M_D30 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 30*30 matrix
Name: hybrid_func1_M_D30 .txt

Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 30*30 matrixes, 1-30 lines are **M1**, 31-60 lines are **M2**,....,271-300 lines are **M10**

Name: hybrid_func1_M_D50 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 50*50 matrix
Name: hybrid_func1_M_D50 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 50*50 matrixes, 1-50 lines are **M1**, 51-100 lines are **M2**,....,451-500 lines are **M10**

2.4.3. F_{17}: F_{16} *with Noise in Fitness*

Let (F_{16} - f_bias_{16}) be $G(x)$, then

$F_{17}(\mathbf{x}) = G(\mathbf{x})*(1+0.2|\mathrm{N}(0,1)|) + f\_bias_{17}$

All settings are the same as F_{16}.

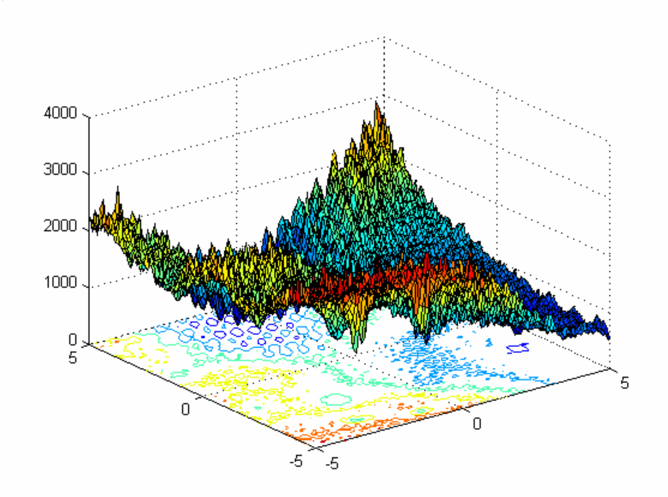

Figure 2-17 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Non-Separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Sphere Functions give two flat areas for the function.
- With Gaussian noise in fitness
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{17}(\mathbf{x}^*) = f\_bias_{17}$ =120

Associated Data file:

Same as F_{16}.

2.4.4. F_{18}: *Rotated Hybrid Composition Function*

$f_{1-2}(\mathbf{x})$: Ackley's Function

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

$f_{3-4}(\mathbf{x})$: Rastrigin's Function

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

$f_{5-6}(\mathbf{x})$: Sphere Function

$$f_i(\mathbf{x}) = \sum_{i=1}^{D}x_i^2$$

$f_{7-8}(\mathbf{x})$: Weierstrass Function

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

a=0.5, b=3, k_{max}=20

$f_{9-10}(\mathbf{x})$: Griewank's Function

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

σ=[1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2];

λ = [2*5/32; 5/32; 2*1; 1; 2*5/100; 5/100; 2*10; 10; 2*5/60; 5/60]

$\mathbf{M}_i$ are all rotation matrices. Condition numbers are [2 3 2 3 2 3 20 30 200 300]

$\mathbf{o}_{10} = [0, 0, ..., 0]$

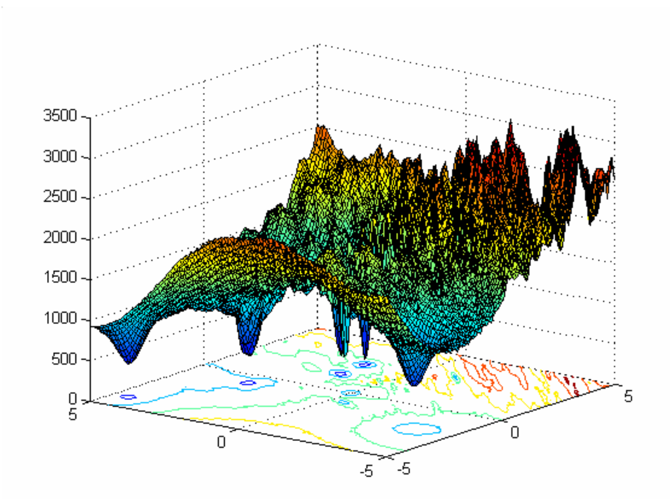

Figure 2-18 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Non-Separable
- Scalable

- A huge number of local optima
- Different function's properties are mixed together
- Sphere Functions give two flat areas for the function.
- A local optimum is set on the origin
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{18}(\boldsymbol{x}^*) = f\_bias_{18} = 10$

Associated Data file:

Name: hybrid_func2_data.mat
hybrid_func2_data.txt
Variable: **o** 10*100 vector the shifted optima for 10 functions
When using, cut **o**=**o**(:,1:*D*)

Name: hybrid_func2_M_D10 .mat
Variable: **M** an structure variable
Contains M**.M1 M.M2, … , M.M10** ten 10*10 matrixes
Name: hybrid_func2_M_D10 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 10*10 matrixes, 1-10 lines are **M1**, 11-20 lines are **M2**,....,91-100 lines are **M10**

Name: hybrid_func2_M_D30 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 30*30 matrix
Name: hybrid_func2_M_D30 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 30*30 matrixes, 1-30 lines are **M1**, 31-60 lines are **M2**,....,271-300 lines are M10

Name: hybrid_func2_M_D50 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 50*50 matrix
Name: hybrid_func2_M_D50 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 50*50 matrixes, 1-50 lines are **M1**, 51-100 lines are **M2**,....,451-500 lines are **M10**

2.4.5. F_{19}: *Rotated Hybrid Composition Function with narrow basin global optimum*

All settings are the same as F_{18} except

σ =[0.1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2];,

λ = [0.1*5/32; 5/32; 2*1; 1; 2*5/100; 5/100; 2*10; 10; 2*5/60; 5/60]

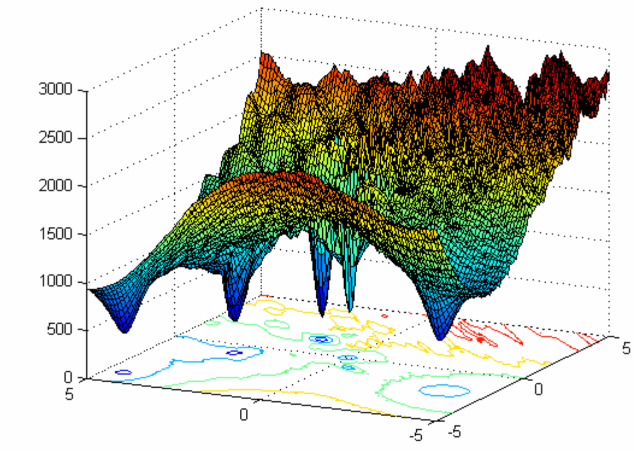

Figure 2-19 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Sphere Functions give two flat areas for the function.
- A local optimum is set on the origin
- A narrow basin for the global optimum
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{19}(\mathbf{x}^*) = f\_bias_{19}(19)=10$

Associated Data file:

Same as F_{18}.

2.4.6. F_{20}: *Rotated Hybrid Composition Function with Global Optimum on the Bounds*

All settings are the same as F_{18} except after load the data file, set $o_{1(2j)} = 5$, for $j = 1, 2, \ldots, \lfloor D/2 \rfloor$

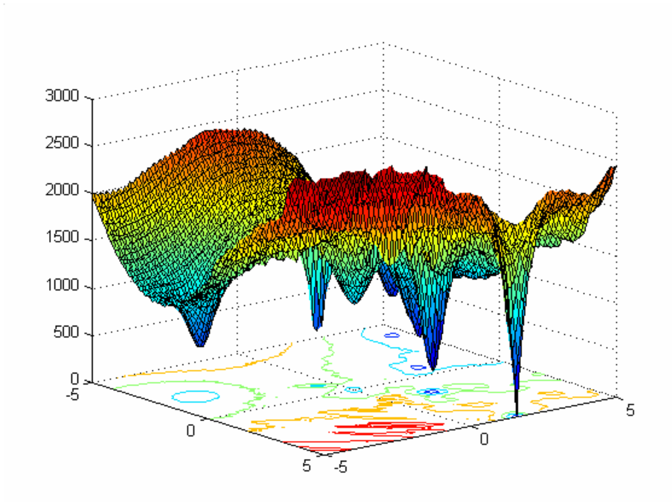

Figure 2-20 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Sphere Functions give two flat areas for the function.
- A local optimum is set on the origin
- Global optimum is on the bound
- If the initialization procedure initializes the population at the bounds, this problem will be solved easily.
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{20}(\mathbf{x}^*) = f\_bias_{20}$ =10

Associated Data file:

Same as F_{18}.

2.4.7. F_{21}: *Rotated Hybrid Composition Function*

$f_{1-2}(\mathbf{x})$: Rotated Expanded Scaffer's F6 Function

$$F(x,y)=0.5+\frac{(\sin^2(\sqrt{x^2+y^2})-0.5)}{(1+0.001(x^2+y^2))^2}$$

$$f_i(\mathbf{x})=F(x_1,x_2)+F(x_2,x_3)+...+F(x_{D-1},x_D)+F(x_D,x_1)$$

$f_{3-4}(\mathbf{x})$: Rastrigin's Function

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

$f_{5-6}(\mathbf{x})$: F8F2 Function

$$F8(\mathbf{x})=\sum_{i=1}^{D}\frac{x_i^{\,2}}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$$

$$F2(\mathbf{x})=\sum_{i=1}^{D-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$$

$$f_i(\mathbf{x})=F8(F2(x_1,x_2))+F8(F2(x_2,x_3))+...+F8(F2(x_{D-1},x_D))+F8(F2(x_D,x_1))$$

$f_{7-8}(\mathbf{x})$: Weierstrass Function

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

a=0.5, b=3, k_{max}=20

$f_{9-10}(\mathbf{x})$: Griewank's Function

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

$\boldsymbol{\sigma}=[1,1,1,1,1,2,2,2,2,2]$,

λ = [5*5/100; 5/100; 5*1; 1; 5*1; 1; 5*10; 10; 5*5/200; 5/200];

$\mathbf{M}_i$ are all orthogonal matrix

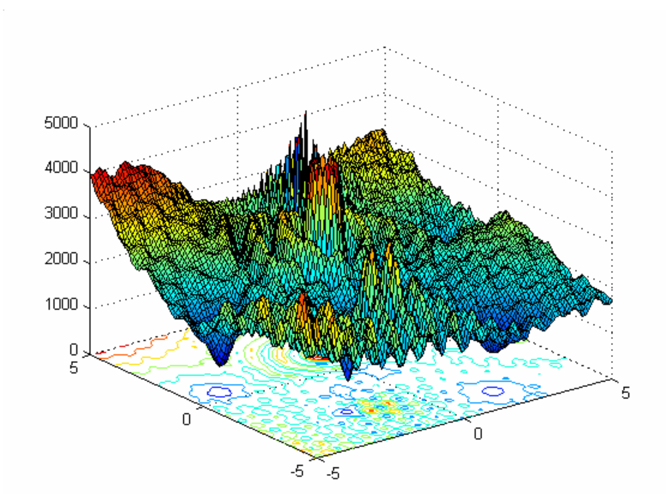

Figure 2-21 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Non-Separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{21}(\mathbf{x}^*) = f\_bias_{21}$ =360

Associated Data file:

Name: hybrid_func3_data.mat
hybrid_func3_data.txt
Variable: **o** 10*100 vector the shifted optima for 10 functions
When using, cut **o**=**o**(:,1:*D*)

Name: hybrid_func3_M_D10 .mat
Variable: **M** an structure variable
Contains **M.M1 M.M2, … , M.M10** ten 10*10 matrixes
Name: hybrid_func3_M_D10 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 10*10 matrixes, 1-10 lines are **M1**, 11-20 lines are **M2,**....,91-100 lines are **M10**

Name: hybrid_func3_M_D30 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 30*30 matrix
Name: hybrid_func3_M_D30 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 30*30 matrixes, 1-30 lines are **M1**, 31-60 lines are **M2**,....,271-300 lines are **M10**

Name: hybrid_func3_M_D50 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 50*50 matrix
Name: hybrid_func3_M_D50 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 50*50 matrixes, 1-50 lines are **M1**, 51-100 lines are **M2**,....,451-500 lines are **M10**

2.4.8. *F_{22}: Rotated Hybrid Composition Function with High Condition Number Matrix*

All settings are the same as F_{21} except $\mathbf{M}_i$'s condition numbers are [10 20 50 100 200 1000 2000 3000 4000 5000]

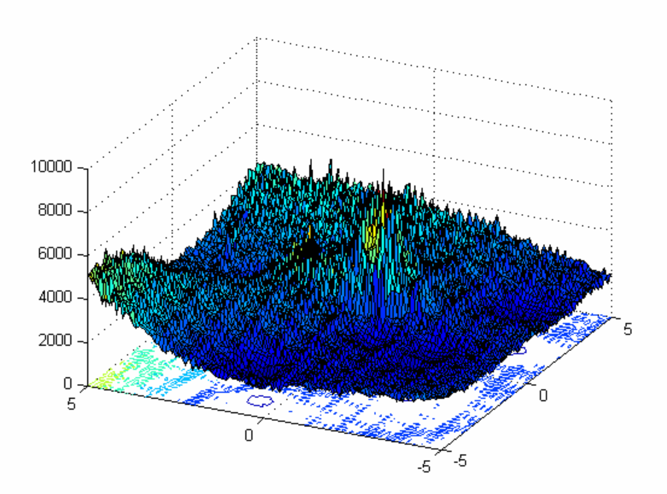

Figure 2-22 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Global optimum is on the bound
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{22}(\mathbf{x}^*) = f\_bias_{22}$ =360

Associated Data file:

Name: hybrid_func3_data.mat
hybrid_func3_data.txt
Variable: **o** 10*100 vector the shifted optima for 10 functions
When using, cut **o**=**o**(:,1:*D*)

Name: hybrid_func3_HM_D10 .mat
Variable: **M** an structure variable
Contains **M.M1 M.M2, … , M.M10** ten 10*10 matrixes
Name: hybrid_func3_HM_D10 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 10*10 matrixes, 1-10 lines are **M1**, 11-20 lines are **M2**,....,91-100 lines are **M10**

Name: hybrid_func3_HM_D30 .mat
Variable: **M** an structure variable contains M.**M1,…,M.M10** ten 30*30 matrix
Name: hybrid_func3_MH_D30 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 30*30 matrixes, 1-30 lines are **M1**, 31-60 lines are **M2**,....,271-300 lines are **M10**

Name: hybrid_func3_MH_D50 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 50*50 matrix
Name: hybrid_func3_HM_D50 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 50*50 matrixes, 1-50 lines are **M1**, 51-100 lines are **M2**,....,451-500 lines are **M10**

2.4.9. F_{23}: *Non-Continuous Rotated Hybrid Composition Function*

All settings are the same as F_{21}.

Except $x_j = \begin{cases} x_j & |x_j - o_{1j}| < 1/2 \\ round(2x_j)/2 & |x_j - o_{1j}| >= 1/2 \end{cases}$ for $j = 1, 2, .., D$

$$round(x) = \begin{cases} a-1 & if \quad x <= 0 \,\&\, b >= 0.5 \\ a & if \quad b < 0.5 \\ a+1 & if \quad x > 0 \,\&\, b >= 0.5 \end{cases},$$

where *a* is x's integral part and *b* is x's decimal part

All "round" operators in this document use the same schedule.

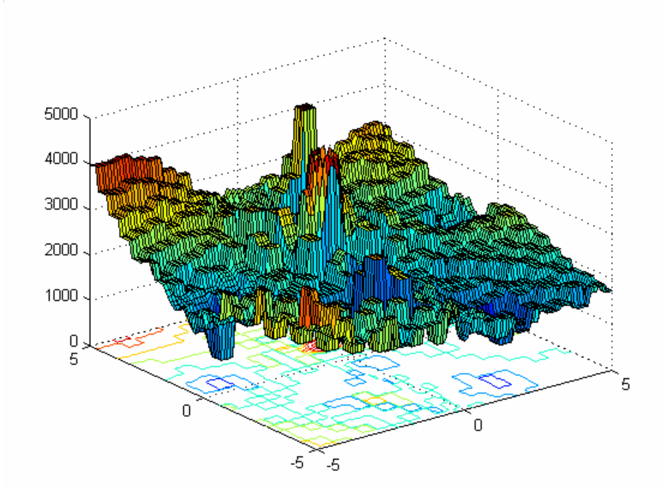

Figure 2-23 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Non-separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Non-continuous
- Global optimum is on the bound
- $\mathbf{x} \in [-5,5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $f(\mathbf{x}^*) \approx$ ***f_bias*** (23)=360

Associated Data file:

Same as F_{21}.

2.4.10. F_{24}: *Rotated Hybrid Composition Function*

$f_1(\mathbf{x})$: Weierstrass Function

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

a=0.5, b=3, k_{max}=20

$f_2(\mathbf{x})$: Rotated Expanded Scaffer's F6 Function

$$F(x,y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$$

$$f_i(\mathbf{x}) = F(x_1,x_2) + F(x_2,x_3) + ... + F(x_{D-1},x_D) + F(x_D,x_1)$$

$f_3(\mathbf{x})$: F8F2 Function

$$F8(\mathbf{x}) = \sum_{i=1}^{D}\frac{{x_i}^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$F2(\mathbf{x}) = \sum_{i=1}^{D-1}(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$f_i(\mathbf{x}) = F8(F2(x_1,x_2)) + F8(F2(x_2,x_3)) + ... + F8(F2(x_{D-1},x_D)) + F8(F2(x_D,x_1))$$

$f_4(\mathbf{x})$: Ackley's Function

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

$f_5(\mathbf{x})$: Rastrigin's Function

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

$f_6(\mathbf{x})$: Griewank's Function

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

$f_7(\mathbf{x})$: Non-Continuous Expanded Scaffer's F6 Function

$$F(x,y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$$

$$f(\mathbf{x}) = F(y_1,y_2) + F(y_2,y_3) + ... + F(y_{D-1},y_D) + F(y_D,y_1)$$

$$y_j = \begin{cases} x_j & |x_j| < 1/2 \\ round(2x_j)/2 & |x_j| >= 1/2 \end{cases} \quad \text{for } j = 1,2,..,D$$

$f_8(\mathbf{x})$: Non-Continuous Rastrigin's Function

$$f(\mathbf{x}) = \sum_{i=1}^{D}({y_i}^2 - 10\cos(2\pi y_i) + 10)$$

$$y_j = \begin{cases} x_j & |x_j| < 1/2 \\ round(2x_j)/2 & |x_j| >= 1/2 \end{cases} \quad \text{for } j = 1, 2, .., D$$

$f_9(\mathbf{x})$: High Conditioned Elliptic Function

$$f(\mathbf{x}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$$

$f_{10}(\mathbf{x})$: Sphere Function with Noise in Fitness

$$f_i(\mathbf{x}) = (\sum_{i=1}^{D} x_i^2)(1 + 0.1|N(0,1)|)$$

$\sigma_i = 2$,for $i = 1, 2..., D$

λ =[10; 5/20; 1; 5/32; 1; 5/100; 5/50; 1; 5/100; 5/100]

$\mathbf{M}_i$ are all rotation matrices, condition numbers are [100 50 30 10 5 5 4 3 2 2];

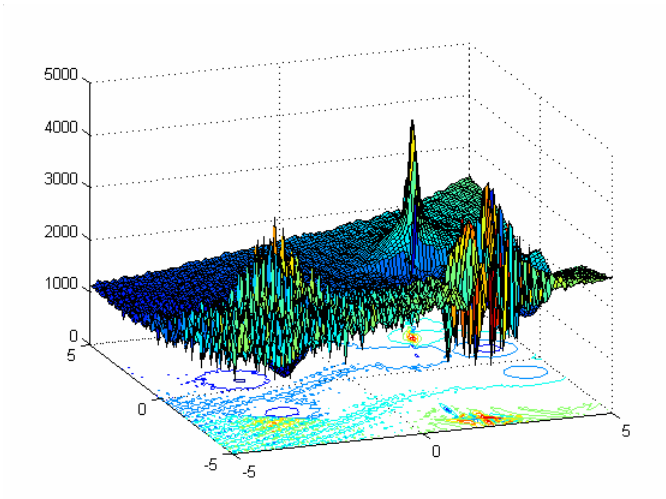

Figure 2-24 3-*D* map for 2-*D* function

Properties:

- Multi-modal
- Rotated
- Non-Separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Unimodal Functions give flat areas for the function.
- $\mathbf{x} \in [-5, 5]^D$, Global optimum $\mathbf{x}^* = \mathbf{o}_1$, $F_{24}(\mathbf{x}^*) = f\_bias_{24} = 260$

Associated Data file:

Name: hybrid_func4_data.mat
hybrid_func4_data.txt

Variable: **o** 10*100 vector the shifted optima for 10 functions
When using, cut **o**=**o**(:,1:*D*)

Name: hybrid_func4_M_D10 .mat
Variable: **M** an structure variable
Contains **M.M1 M.M2, … , M.M10** ten 10*10 matrixes
Name: hybrid_func4_M_D10 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 10*10 matrixes, 1-10 lines are **M1**, 11-20 lines are **M2**,....,91-100 lines are **M10**

Name: hybrid_func4_M_D30 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 30*30 matrix
Name: hybrid_func4_M_D30 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 30*30 matrixes, 1-30 lines are **M1**, 31-60 lines are **M2**,....,271-300 lines are **M10**

Name: hybrid_func4_M_D50 .mat
Variable: **M** an structure variable contains **M.M1,…,M.M10** ten 50*50 matrix
Name: hybrid_func4_M_D50 .txt
Variable: **M1 M2 M3 M4 M5 M6 M7 M8 M9 M10** are ten 50*50 matrixes, 1-50 lines are **M1**, 51-100 lines are **M2**,....,451-500 lines are **M10**

2.4.11. F_{25}: *Rotated Hybrid Composition Function without bounds*

All settings are the same as F_{24} except no exact search range set for this test function.

Properties:

- Multi-modal
- Non-separable
- Scalable
- A huge number of local optima
- Different function's properties are mixed together
- Unimodal Functions give flat areas for the function.
- Global optimum is on the bound
- No bounds
- Initialize population in $[2,5]^D$, Global optimum $\boldsymbol{x}^* = \boldsymbol{o}_1$ is outside of the initialization range, $F_{25}(\mathbf{x}^*) = f\_bias_{25}$ =260

Associated Data file:

Same as F_{24}

2.5 Comparisons Pairs

Different Condition Numbers:

- F_1. Shifted Rotated Sphere Function
- F_2. Shifted Schwefel's Problem 1.2
- F_3. Shifted Rotated High Conditioned Elliptic Function

Function With Noise Vs Without Noise

Pair 1:

- F_2. Shifted Schwefel's Problem 1.2
- F_4. Shifted Schwefel's Problem 1.2 with Noise in Fitness

Pair 2:

- F_{16}. Rotated Hybrid Composition Function
- F_{17}. F_{16}. with Noise in Fitness

Function without Rotation Vs With Rotation

Pair 1:

- F_9. Shifted Rastrigin's Function
- F_{10}. Shifted Rotated Rastrigin's Function

Pair 2:

- F_{15}. Hybrid Composition Function
- F_{16}. Rotated Hybrid Composition Function

Continuous Vs Non-continuous

- F_{21}. Rotated Hybrid Composition Function
- F_{23}. Non-Continuous Rotated Hybrid Composition Function

Global Optimum on Bounds Vs Global Optimum on Bounds

- F_{18}. Rotated Hybrid Composition Function
- F_{20}. Rotated Hybrid Composition Function with the Global Optimum on the Bounds

Wide Global Optimum Basin Vs Narrow Global Optimum Basin

- F_{18}. Rotated Hybrid Composition Function
- F_{19}. Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum

Orthogonal Matrix Vs High Condition Number Matrix

- F_{21}. Rotated Hybrid Composition Function
- F_{22}. Rotated Hybrid Composition Function with High Condition Number Matrix

Global Optimum in the Initialization Range Vs outside of the Initialization Range

- F_{24}. Rotated Hybrid Composition Function
- F_{25}. Rotated Hybrid Composition Function without Bounds

2.6 Similar Groups:

Unimodal Functions

Function 1-5

Multi-modal Functions

Function 6-25

- Single Function: Function 6-12
- Expanded Function: Function 13-14
- Hybrid Composition Function: Function 15-25

Functions with Global Optimum outside of the Initialization Range

- F_7. Shifted Rotated Griewank's Function without Bounds
- F_{25}. Rotated Hybrid Composition Function 4 without Bounds

Functions with Global Optimum on Bounds

- F_5. Schwefel's Problem 2.6 with Global Optimum on Bounds
- F_8. Shifted Rotated Ackley's Function with Global Optimum on Bounds
- F_{20}. Rotated Hybrid Composition Function 2 with the Global Optimum on the Bounds

3. Evaluation Criteria

3.1 Description of the Evaluation Criteria

Problems: 25 minimization problems

Dimensions: *D*=10, 30, 50

Runs / problem: 25 (**Do not run many 25 runs to pick the best run**)

Max_FES: 10000**D* (Max_FES_10D= 100000; for 30D=300000; for 50D=500000)

Initialization: Uniform random initialization within the search space, except for problems 7 and 25, for which initialization ranges are specified.

Please use the same initializations for the comparison pairs (problems 1, 2, 3 & 4, problems 9 & 10, problems 15, 16 & 17, problems 18, 19 & 20, problems 21, 22 & 23, problems 24 & 25). One way to achieve this would be to use a fixed seed for the random number generator.

Global Optimum: All problems, except 7 and 25, have the global optimum within the given bounds and there is no need to perform search outside of the given bounds for these problems. 7 & 25 are exceptions without a search range and with the global optimum outside of the specified initialization range.

Termination: Terminate before reaching Max_FES if the error in the function value is 10^{-8} or less.

Ter_Err: 10^{-8} (termination error value)

1) **Record function error value (f(*x*)-f(*x**)) after 1e3, 1e4, 1e5 FES and at termination (due to Ter_Err or Max_FES) for each run.**

   For each function, sort the error values in 25 runs from the smallest (best) to the largest (worst)

   Present the following: 1^{st} (best), 7^{th}, 13^{th} (median), 19^{th}, 25^{th} (worst) function values

   Mean and STD for the 25 runs

2) **Record the FES needed in each run to achieve the following fixed accuracy level. The Max_FES applies.**

Table 3-1 Fixed Accuracy Level for Each Function

Function	Accuracy	Function	Accuracy
1	-450 + 1e-6	14	-300 + 1e-2

2	-450 + 1e-6	15	120 + 1e-2
3	-450 + 1e-6	16	120 + 1e-2
4	-450 + 1e-6	17	120 + 1e-1
5	-310 + 1e-6	18	10+ 1e-1
6	390 + 1e-2	19	10 + 1e-1
7	-180 + 1e-2	20	10 + 1e-1
8	-140 + 1e-2	21	360 + 1e-1
9	-330 + 1e-2	22	360 + 1e-1
10	-330 + 1e-2	23	360 + 1e-1
11	90 + 1e-2	24	260 + 1e-1
12	-460 + 1e-2	25	260 + 1e-1
13	-130 + 1e-2		

Successful Run: A run during which the algorithm achieves the fixed accuracy level within the Max_FES for the particular dimension.

For each function/dimension, sort FES in 25 runs from the smallest (best) to the largest (worst**)**

Present the following: 1^{st} (best), 7^{th}, 13^{th} (median), 19^{th}, 25^{th} (worst) FES
Mean andSTD for the 25 runs

3) Success Rate & success Performance For Each Problem

Success Rate**= (#** of successful runs according to the table above) / total runs

Success Performance=mean (FEs for successful runs)*(# of total runs) / (# of successful runs)

The above two quantities are computed for each problem separately.

4) Convergence Graphs (or Run-length distribution graphs)

Convergence Graphs for each problem for ***D*=30**. The graph would show the median performance of the total runs with termination by either the Max_FES or the Ter_Err. The semi-log graphs should show log10(f($\boldsymbol{x}$)- f($\boldsymbol{x}$*)) vs FES for each problem.

5) Algorithm Complexity

a) Run the test program below:

```
for i=1:1000000
x= (double) 5.55;
x=x + x; x=x./2; x=x*x; x=sqrt(x); x=ln(x); x=exp(x); y=x/x;
end
```

Computing time for the above=*T0*;

b) evaluate the computing time just for Function 3. 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 benchmark function 3 is *T2*. Execute step c 5 times and get 5 *T2* values.

$\widehat{T}2$ =Mean(*T2*)

The complexity of the algorithm is reflected by: $\widehat{T}2$, *T1*, *T0*, and ($\widehat{T}2$ -*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 5 times to accommodate variations in execution time due adaptive nature of some algorithms.

6) Parameters

We discourage participants searching 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.

7) 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.

3.2 Example

System: Windows XP (SP1)

CPU: Pentium(R) 4 3.00GHz

RAM: 1 G

Language: Matlab 6.5

Algorithm: Particle Swarm Optimizer (PSO)

Results

***D*=10**
Max_FES=100000

Table 3-2 Error Values Achieved When FES=1e3, FES=1e4, FES=1e5 for Problems 1-8

FES \ Prob		1	2	3	4	5	6	7	8
1e3	1st(Best)	4.8672e+2	4.7296e+2	2.2037e+6	4.6617e+2	2.3522e+3			
	7th	8.0293e+2	9.8091e+2	8.5141e+6	1.2900e+3	4.0573e+3			
	13th(Median)	9.2384e+2	1.5293e+3	1.4311e+7	1.9769e+3	4.6308e+3			
	19th	1.3393e+3	1.7615e+3	1.9298e+7	2.9175e+3	4.8015e+3			
	25th (Worst)	1.9151e+3	3.2337e+3	4.4688e+7	6.5038e+3	5.6701e+3			
	Mean	1.0996e+3	1.5107e+3	1.5156e+7	2.3669e+3	4.4857e+3			
	Std	4.0575e+2	7.2503e+2	9.3002e+6	1.5082e+3	7.0081e+2			
1e4	1st(Best)	3.1984e-3	1.0413e+0	1.3491e+5	6.7175e+0	1.6584e+3			
	7th	2.6509e-2	1.3202e+1	4.4023e+5	3.8884e+1	2.3522e+3			
	13th(Median)	6.0665e-2	1.9981e+1	1.1727e+6	5.5027e+1	2.6335e+3			
	19th	1.0657e-1	3.5319e+1	2.0824e+6	7.1385e+1	2.8788e+3			
	25th (Worst)	4.3846e-1	1.0517e+2	2.9099e+6	1.7905e+2	3.6094e+3			
	Mean	8.6962e-2	2.7883e+1	1.3599e+6	5.9894e+1	2.6055e+3			
	Std	9.6616e-2	2.3526e+1	9.1421e+5	3.5988e+1	4.5167e+2			
1e5	1st(Best)	4.7434e-9T	5.1782e-9T	4.2175e+4	1.7070e-5	1.1864e+3			
	7th	7.9845e-9T	8.5278e-9T	1.2805e+5	1.2433e-3	1.4951e+3			
	13th(Median)	9.0901e-9T	9.7281e-9T	2.3534e+5	4.0361e-3	1.7380e+3			
	19th	9.6540e-9T	1.5249e-8	4.6436e+5	1.8283e-2	1.9846e+3			
	25th (Worst)	9.9506e-9T	2.3845e-7	2.2776e+6	3.9795e-1	2.3239e+3			
	Mean	8.5375e-9T	3.2227e-8	4.6185e+5	3.4388e-2	1.7517e+3			
	Std	1.4177e-9T	6.2340e-8	5.4685e+5	8.2733e-2	2.9707e+2			

*** xxx.e-9T means it get termination error before it gets the predefined record FES.**

Table 3-3 Error Values Achieved When FES=1e+3, FES=1e+4, FES=1e+5 for Problems 9-17

FES \ Prob		9	10	11	12	13	14	15	16	17
1e+3	1st(Best)									
	7th									
	13th(Median)									
	19th									
	25th (Worst)									
	Mean									
	Std									
1e+4	1st(Best)									
	7th									
	13th(Median)									
	19th									
	25th (Worst)									
	Mean									
	Std									
1e+5	1st(Best)									
	7th									
	13th(Median)									
	19th									
	25th (Worst)									
	Mean									
	Std									

Table 3-4 Error Values Achieved When FES=1e+3, FES=1e+4, FES=1e+5 for Problems 18-25

FES \ Prob		18	19	20	21	22	23	24	25
1e+3	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								
1e+4	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								
1e+5	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								

Table 3-5 Number of FES to achieve the fixed accuracy level

Prob	1^{st}(Best)	7^{th}	13^{th} (Median)	19^{th}	25^{th} (Worst)	Mean	Std	Success rate	Success Performance
1	11607	12133	12372	12704	13022	1.2373e+4	3.6607e+2	100%	1.2373e+4
2	17042	17608	18039	18753	19671	1.8163e+4	7.5123e+2	100%	1.8163e+4
3	-	-	-	-	-	-	-	0%	-
4	-	-	-	-	-	-	-	0%	-
5	-	-	-	-	-	-	-	0%	-
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									

***D*=30**
Max_FES=300000

Table 3-6 Error Values Achieved When FES=1e3, FES=1e4, FES=1e5 for Problems 1-8

FES	Prob	1	2	3	4	5	6	7	8
1e3	1^{st}(Best)								
	7^{th}								
	13^{th}(Median)								
	19^{th}								
	25^{th} (Worst)								
	Mean								
	Std								
1e4	1^{st}(Best)								
	7^{th}								
	13^{th}(Median)								
	19^{th}								
	25^{th} (Worst)								
	Mean								
	Std								
1e5	1^{st}(Best)								
	7^{th}								
	13^{th}(Median)								
	19^{th}								

	25th (Worst)								
	Mean								
	Std								
3e5	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								

……
……

D=50
Max_FES=500000

Table 3-7 Error Values Achieved When FES=1e3, FES=1e4, FES=1e5 for Problems 1-8

FES \ **Prob**		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1e3	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								
1e4	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								
1e5	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								
3e5	1st(Best)								
	7th								
	13th(Median)								
	19th								
	25th (Worst)								
	Mean								
	Std								

……
……

Convergence Graphs (30D)

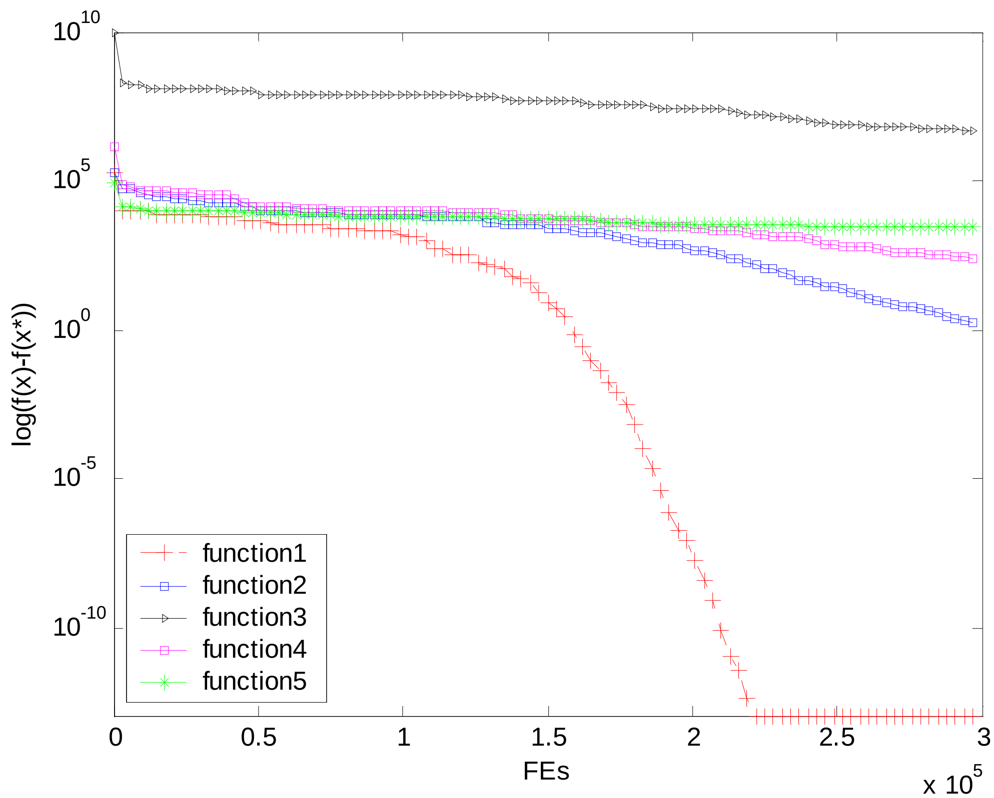

Figure 3-1 Convergence Graph for Functions 1-5

…

Figure 3-2 Convergence Graph for Function 6-10

…

Figure 3-3 Convergence Graph for Function 11-14

…

Figure 3-4 Convergence Graph for Function 15-20

…

Figure 3-5 Convergence Graph for Function 21-25

Algorithm Complexity

Table 3-8 Computational Complexity

	T0	*T1*	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
D=10	39.5470	31.1250	82.3906	1.2963
D=30		38.1250	90.8437	1.3331
D=50		46.0780	108.9094	1.5888

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.

4. Notes

Note 1: Linear Transformation Matrix

M=**P*****N*****Q**

P, **Q** are two orthogonal matrixes, generated using Classical Gram-Schmidt method
N is diagonal matrix

$$u = rand(1, D)\ ,\ d_{ii} = c^{\frac{u_i - \min(u)}{\max(u) - \min(u)}}$$

M's condition number Cond(**M**)=c

Note 2: On page 17, w_i values are sorted and raised to a higher power. The objective is to ensure that each optimum (local or global) is determined by only one function while allowing a higher degree of mixing of different functions just a very short distance away from each optimum.

Note 3: We assign different positive and negative objective function values, instead of zeros. This may influence some algorithms that make use of the objective values.

Note 4: We assign the same objective values to the comparison pairs in order to make the comparison easier.

Note 5: High condition number rotation may convert a multimodal problem into a unimodal problem. Hence, moderate condition numbers were used for multimodal.

Note 6: Additional data files are provided with some coordinate positions and the corresponding fitness values in order to help the verification process during the code translation.

Note 7: It is insufficient to make any statistically meaningful conclusions on the pairs of problems as each case has at most 2 pairs. We would probably require 5 or 10 or more pairs for each case. We would consider this extension for the edited volume.

Note 8: Pseudo-real world problems are available from the web link given below. If you have any queries on these problems, please contact Professor Darrell Whitley directly. Email: whitley@CS.ColoState.EDU
`Web-link: http://www.cs.colostate.edu/~genitor/functions.html.`

Note 9: We are recording the numbers such as 'the number of FES to reach the given fixed accuracy', 'the objective function value at different number of FES' **for each run of each problem and each dimension** in order to perform some statistical significance tests. The details of a statistical significance test would be made available a little later.

References:

[1] N. Hansen, S. D. Muller and P. Koumoutsakos, “Reducing the Time Complexity of the Derandomized evolution Strategy with Covariance Matrix Adaptation (CMA-ES).” *Evolutionary Computation*, 11(1), pp. 1-18, 2003

[2] A. Klimke, “Weierstrass function’s matlab code”, http://matlabdb.mathematik.uni-stuttgart.de/download.jsp?MC_ID=9&MP_ID=56

[3] H-P. Schwefel, “Evolution and Optimum Seeking”, http://ls11-www.cs.uni-dortmund.de/lehre/wiley/

[4] D. Whitley, K. Mathias, S. Rana and J. Dzubera, “Evaluating Evolutionary Algorithms” *Artificial Intelligence*, 85 (1-2): 245-276 AUG 1996.