Novel Multimodal Problems and Differential Evolution with Ensemble of Restricted Tournament Selection

Bo-Yang Qu, Ponnuthurai Nagaratnam Suganthan, Snr Member, IEEE

Abstract—Multi-modal optimization refers to locating not only one optimum but a set of locally optimal solutions. Niching is an important technique to solve multi-modal optimization problems. The ability of discover and maintain multiple niches is the key capability of these algorithms. In this paper, differential evolution with an ensemble of restricted tournament selection (ERTS-DE) algorithm is introduced to perform multimodal optimization. The algorithms is tested on 15 newly designed scalable benchmark multi-modal optimization problems and compared with the crowding differential evolution (Crowding-DE) in the literature. As shown by the experimental results, the proposed algorithm outperforms the Crowding-DE on the novel scalable benchmark problems.

I. INTRODUCTION

In real world optimization, many engineering problems can be classified as multi-modal problems, such as classification problems in machine learning [1] and inversion of teleseismic waves [2]. The aim is to locate several globally or locally optimal solutions and then to choose the most appropriate solution considering practical issues. In recent years, evolutionary algorithms (EA) with various niching techniques have been successfully applied to solve multi-modal optimization problems. The earliest niching approach was proposed by Cavicchio [3]. Subsequently, many other niching methods, such as crowding [4] and clearing [5], have also been proposed.

Differential evolution is a very powerful optimization technique compared with other EAs such as genetic algorithms and evolutionary programming. Like other EAs, DE is also a population-based algorithm. Although DE has been proven to be effective in locating one globally optimal solution [6], the basic DE is not efficient for solving multi-modal optimization problems [7]. Some work has been done to extend the DE to solve multi-modal problems [8]-[9]. Thomsen proposed a Crowding-DE [7] and showed that Crowding-DE outperformed a DE based fitness sharing algorithm. In this paper, DE with an ensemble of crowding and restricted tournament selection (ECRTS-DE) is proposed and compared with the Crowding-DE on a set of newly designed scalable multi-modal optimization problems.

The remainder of this paper is structured as follows. Section II provides a brief overview of differential evolution, crowding and restricted tournament selection as well as the Crowding-DE algorithm. In Section III, the proposed ERTS-DE algorithm is introduced. The definition of newly developed problems and the results of the experiments are presented in Sections IV and V, respectively. Finally, the paper is concluded in Section VI.

II. CROWDING DIFFERENTIAL EVOLUTION

This section introduces the differential evolution algorithm, crowding and restricted tournament selection based niching algorithms and crowding differential evolution algorithm which is a DE and crowding based multimodal optimization algorithm.

A. Differential Evolution

The differential evolution (DE) algorithm was first introduced by Storn and Price [10] and widely used in different areas [11]-[13]. The four major steps involved in DE are known as, initialization, mutation, recombination and selection. In the mutation operation, one of the following strategies is used [14]:

DE/rand/1:
$$v_p = x_{r1} + F \cdot (x_{r2} - x_{r3})$$

DE/best/1: $v_p = x_{best} + F \cdot (x_{r1} - x_{r2})$
DE/current-to-best/2:
 $v_p = x_p + F \cdot (x_{best} - x_p + x_{r1} - x_{r2})$
DE/best/2: $v_p = x_{best} + F \cdot (x_{r1} - x_{r2} + x_{r3} + x_{r4})$
DE/rand/2: $v_p = x_{r1} + F \cdot (x_{r2} - x_{r3} + x_{r4} - x_{r5})$

where r1, r2, r3, r4, r5 are mutually different integers randomly generated in the range [1, NP (population size)], F is the scale factor used to scale differential vectors. x_{best} is the solution with the best fitness value in the current population.

The crossover operation is applied to each pair of the generated mutant vector and its corresponding parent vector using the following equations:

$$u_{p,i} = \begin{cases} v_{p,i} & \text{if } rand_i \leq CR \\ x_{p,i} & \text{otherwise} \end{cases}$$

where u_p is the offspring vector. *CR* is the crossover rate which is a user-specified constant.

B. Crowding and Restricted Tournament Selection

Crowding [4] was introduced by De Jong in 1975 and extended to restricted tournament selection by Harik [15]. It differs from a simple evolutionary algorithm in the way of replacing individuals in the current population by offspring.

Manuscript received February 7, 2010. This work was supported by the Nanyang Technological University.

Authors are with Nanyang Technological University, School of Electrical and Electronic Engineering, Singapore, 639798 (phone: 65-67905404; fax: 65-67933318; emails: E070088@ntu.edu.sg, epnsugan@ntu.edu.sg).

For crowding and restricted tournament selection, in order to compare the offspring with the current population, a random set of w (window size) individuals are selected from the current population and the nearest to the offspring is determined by Euclidean distance measure. Finally, this nearest individual is replaced by the offspring if its fitness value is worse than the offspring's fitness value. This process is repeated for all the offspring in each generation. Crowding and restricted tournament selection methods are effective in maintaining the diversity of the population, which is important in multi-modal optimization.

C. Crowding DE

Crowding DE was first introduced by Thomsen to solve multi-modal optimization problems [7]. In Crowding DE, the fitness value of an offspring is compared with that of the nearest individual in the current population (*w* is same as the population size). The steps of Crowding DE are shown in Table I.

	Table I. Crowding DE algorithm
Step 1	Use the basic DE to produce NP (population size)
	offspring.
	For $i=1:NP$
Step 2	Calculate the Euclidean distance values of the
	offspring(<i>i</i>) to the other individuals in the DE
	Population.
Step 3	Compare the fitness value of offspring(<i>i</i>) and
	the fitness value of the individual that has the
	smallest Euclidean distance. The offspring will
	replace the individual if it is fitter than the
	individual.
	Endfor
Step 4	Stop if the termination criterion is met, otherwise go
_	to step 1.

III. ERTS-DE

As we know, there is one key parameter w that controls the performance of restricted tournament selection (Crowding DE is a special case with w=NP). According to the "No free lunch" theorem [16], it is impossible to find one parameter value that can be better than all other parameter values for all problems. Motivated by this observation, an ensemble of restricted tournament selection DE is proposed using parallel populations with different window sizes. In other words, different populations are used. In this paper, two populations can be used, if additional different parameters or different niching algorithms are used. Each populations need not only compete with their own offspring, but also the offspring generated by the other population. In this way, the algorithm

will always keep the offspring that was generated by the more suitable parameter leading to a better performance. The flowchart of the ERTS-DE algorithm is shown in Fig. 1.



Fig. 1. Flowchart of the ERTS-DE algorithm

IV. PROBLEM DEFINITIONS

There are several multi-modal benchmark problems available in the literature. However, these problems are relatively easy and many algorithms can solve them perfectly. There is also a lack of scalable multi-modal problems. Therefore, it is difficult to differentiate the performance of advanced algorithms. To overcome these problems, a new set of scalable multi-modal problems is designed in this article by making use of composition functions in [17]. All the test functions are maximization problems with equal globally optimal fitness value of 0. The composition functions are defined as follows:

F(x): new composition function

- $f_i(x)$: ith basic function used to construct the composition function.
- *n:* number of basic functions (number of optima)
- *D*: dimensions (can be chosen from 1-100)
- M_i : linear transformation matrix for each $f_i(x)$
- o_i : new shifted optima position for each $f_i(x)$

$$F(x) = \sum_{i=1}^{n} \left\{ w_i * [f'_i((x - o_i) / \lambda_i * M_i)] \right\}$$

 w_i : weight value for each $f_i(x)$, calculated as follow:

$$w_{i} = \exp(-\frac{\sum_{k=1}^{D} (x_{k} - o_{ik})}{2D\sigma_{i}^{2}})$$

$$w_{i} = \begin{cases} w_{i} & w_{i} == \max(w_{i}) \\ w_{i} * (1 - \max(w_{i}) .^{1}0) & w_{i} \neq \max(w_{i}) \end{cases}$$
then normalize the weight $w_{i} = w_{i} / \sum_{k=1}^{n} w_{i}$

 σ_i : used to control each $f_i(x)$'s coverage range. λ_i : used to stretch compress the function. $f'_i(x) = C * f_i(x) / |f_{maxi}|$, *C* is a predefined constant.

 $|f_{\max i}|$ is estimated

using: $|f_{\text{max}i}| = f_i((x' / \lambda_i) * M_i), x' = [5, 5, ..., 5]$

Composition Function 1 (F1, *n*=8)

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

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

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

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

a=0.5, b=3, k_{max} =20 $f_{5-6}(x)$: Griewank's Function

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

$$f_{7-8}(x) : \text{Sphere Function}$$

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

$$\sigma_i = 1$$
 for all *i*

 $\lambda = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32]$

 M_i : are all identity matrices

These formulas are basic functions; shift and rotation should be added to these functions. Take f_1 as an example, the following function should be evaluated:

$$f_i(z) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10)$$

where $z = ((x - o_i) / \lambda_1) * M_1$.

Composition Function 2 (F2 *n*=6)

 $f_{1-2}(x) : \text{Griewank's Function} \\ f_{3-4}(x) : \text{Weierstrass Function} \\ f_{5-6}(x) : \text{Sphere Function} \\ \sigma_i = 1 \text{ for all } i \\ \lambda = [1, 1, 10, 10, 5/60, 5/60,] \\ M_i : \text{ are all identity matrices} \end{cases}$

Composition Function 3 (F3 *n*=6)

 $f_{1-2}(x) : \text{Rastrigin's Function} \\ f_{3-4}(x) : \text{Griewank's Function} \\ f_{5-6}(x) : \text{Sphere Function} \\ \sigma_i = 1 \text{ for all } i \\ \lambda = [1, 1, 10, 10, 5/60, 5/60,] \\ M_i : \text{are all identity matrices} \end{cases}$

Composition Function 4 (F4 *n*=6)

 $f_{1-2}(x)$: Rastrigin's Function $f_{3-4}(x)$: Weierstrass Function $f_{5-6}(x)$: Griewank's Function $\sigma_i = 1$ for all *i* $\lambda = [1, 1, 10, 10, 5/60, 5/60,]$ M_i : are all identity matrices

Composition Function 5 (F5 *n*=6)

 $f_{1-2}(x)$: Rastrigin's Function $f_{3-4}(x)$: Weierstrass Function

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

$$\sigma_i = 1$$
 for all *i*

$$\lambda = [1, 1, 10, 10, 5/60, 5/60,]$$

 M_i : are all identity matrices

Composition Function 6 (F6 *n*=6)

 $f_{1-2}(x)$: F8F2 Function

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

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

$$\begin{split} f_i(x) &= F8(F2(x_1,x_2)) + F8(F2(x_2,x_3)) + \ldots + \\ F8(F2(x_{D-1},x_D) + F8(F2(x_{D},x_1))) \\ f_{3-4}(x): & \text{Weierstrass Function} \\ f_{5-6}(x): & \text{Griewank's Function} \\ \sigma &= [1,1,1,1,2], \\ \lambda &= [5*5/100;5/100;5*1;1;5*1;1] \\ M_i: \text{ are all orthogonal matrix} \end{split}$$

Composition Function 7 (F7 *n*=6)

$$\begin{split} f_{1-2}(x) &: \text{Rotated Expanded Scaffer's F6 Function} \\ F(x,y) &= 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2} \\ f_i(x) &= F(x_1, x_2) + F(x_2, x_3) + ... + F(x_{D-1}, x_D) + F(x_D, x_1) \\ f_{3-4}(x) &: \text{F8F2 Function} \\ f_{5-6}(x) &: \text{Weierstrass Function} \\ \sigma &= [1, 1, 1, 1, 2], \\ \lambda &= [5; 10; 5; 1; 5*5/100; 5/100] \\ M_i &: \text{are all orthogonal matrix} \end{split}$$

Composition Function 8 (F8 *n*=6)

 $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function $f_{3-4}(x)$: F8F2 Function $f_{5-6}(x)$: Griewank's Function $\sigma = [1,1,1,1,2],$ $\lambda = [5*5/100;5/100;5*1;1;5*1;1]$ M_i : are all orthogonal matrix

Composition Function 9 (F9 *n*=6)

 $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function $f_{3-4}(x)$: Weierstrass Function $f_{5-6}(x)$: Griewank's Function $\sigma = [1,1,1,1,2],$ $\lambda = [5;10;5*5/100;5/100;5;1]$ M_i : are all orthogonal matrix

Composition Function 10 (F10 *n*=6)

 $f_{1-2}(x) : \text{Rastrigin's Function}$ $f_{3-4}(x) : \text{F8F2 Function}$ $f_{5-6}(x) : \text{Weierstrass Function}$ $\sigma = [1,1,1,1,2],$ $\lambda = [5;10;5*5/100;5/100;5;1]$ $M_i : \text{are all orthogonal matrix}$

Composition Function 11 (F11 *n*=8)

 $f_{1-2}(x) : \text{Rastrigin's Function}$ $f_{3-4}(x) : \text{F8F2 Function}$ $f_{5-6}(x) : \text{Weierstrass Function}$ $f_{7-8}(x) : \text{Griewank's Function}$ $\sigma = [1,1,1,1,1,2,2,2],$ $\lambda = [5;1;5;1;50;10;5*5/200;5/200]$ M_i : are all orthogonal matrix

Composition Function 12 (F12 *n*=8)

 $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function $f_{3-4}(x)$: F8F2 Function $f_{5-6}(x)$: Weierstrass Function $f_{7-8}(x)$: Griewank's Function $\sigma = [1,1,1,1,1,2,2,2],$ $\lambda = [5*5/100;5/100;5;1;5;1;50;10]$ M_i : are all orthogonal matrix

Composition Function 13 (F13 *n*=10)

 $f_{1-2}(x) : \text{Rotated Expanded Scaffer's F6 Function}$ $f_{3-4}(x) : \text{Rastrigin's Function}$ $f_{5-6}(x) : \text{F8F2 Function}$ $f_{7-8}(x) : \text{Weierstrass Function}$ $f_{9-10}(x) : \text{Griewank's Function}$ $\sigma = [1,1,1,1,1,2,2,2,2,2],$ $\lambda = [5*5/100;5/100;5;1;5;1;50;10;5*5/200;5/200]$ $M_i : \text{ are all orthogonal matrix}$

Composition Function 14 (F14 *n*=10)

All settings are the same as F13, except M_i 's condition numbers are [10 20 50 100 200 1000 2000 3000 4000 5000]

Composition Function 15 (F15 *n*=10)

- $f_1(x)$: Weierstrass Function
- $f_2(x)$: Rotated Expanded Scaffer's F6 Function
- $f_3(x)$: F8F2 Function
- $f_4(x)$: Ackley's Function

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

- $f_5(x)$: Rastrigin's Function
- $f_6(x)$: Griewank's Function
- $f_7(x)$: Non-Continuous Expanded Scaffer's F6 Function

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

$$f_i(x) = F(y_1, y_2) + F(y_2, y_3) + \dots + F(y_{D-1}, y_D) + F(y_D, y_1)$$

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

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

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

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

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

 $f_9(x)$: High Conditioned Elliptic Function

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

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

]	Table II. Test fu	nction propertie	S
Test Eurotion	No. of equal	No. of basic	Rotation used
Test Function	global optima	function used	(Y/N)
F1	8	4	Ν
F2	6	3	Ν
F3	6	3	Ν
F4	6	3	Ν
F5	6	3	Ν
F6	6	3	Y
F7	6	3	Y
F8	6	3	Y
F9	6	3	Y
F10	6	3	Y
F11	8	4	Y
F12	8	4	Y
F13	10	5	Y
F14	10	5	Y
F15	10	10	Y

V. EXPERIMENTS AND RESULTS

F=0.9, CR=0.1,
Two experiments are conducted as follows:
1. D=10, Test Functions: F1-F15
2. D=30, Test Functions: F1-F5
For comparison, the following two criteria are used:
1. Number of optima found [18]
2. The best value found
An optimum is considered to be found if there exists a
solution in the population within the tolerated Euclidean
distance to that optimum. The tolerance for all problems is set

shown in Tables III-V. Since for D=30, both algorithms are not able to locate any global optimum, the number of optima found for these problems will be zero. As can been seen from the results, the proposed algorithm outperforms the Crowding-DE on all benchmark problems.

Table III Comparison of number of optima found (D=10)

$f(x) = \sum_{i=1}^{n} (10^{\circ})^{i}$	$() X_i$			Test Evention		Crowdin o DE	
$f_{i}(x)$: Sphe	ere Function with	Noise in Fitness	3	Test Function		Crowding DE	
J10(0) ~ ~ P					Best	1	2
$f_{x}(x) = (\sum_{i=1}^{n} f_{x}(x))$	$x^{2}(1+0.1)N(0.1)$	1))		F1	Worst	0	0
J_{i}		· ·			Mean	0.1	1.2
<i>n</i> =10					Best	3	3
$\sigma = 2$ for all	l <i>i</i>				Worst	1	3
1 [10.5/00		5 / 50 1 5 / 100 A	(100]	F2	Mean	2	3
$\lambda = [10; 5/20]$;1;5/32;1;5/100	;5/50;1;5/100;3	5/100]		Std	0.6667	0
M_i are all rota	ation matrices, co	ondition number	are [100 50 30		Best	1	3
10 5 5 4 3 2 2	[];			F3	Worst	0	1
	Table II. Test fu	nction propertie	S		Mean	0.1	2.1 0.5676
Test Eurotion	No. of equal	No. of basic	Rotation used		Best	0.3102	0.5070
rest runction	global optima	function used	(Y/N)		Worst	0	1
F1	8	4	Ν	F4	Mean	0.7	1.5
F2	6	3	N		Std	0.4831	0.5270
F3	6	3	N		Best	2	4
F4	6	3	N	F5	Worst	0	2
FS	6	3	N		Mean	1.4	2.9
F0 E7	0	3	Y V		Best	0.0992	0.3077
Г7 F8	0	3	I V	74	Worst	0	1
F9	6	3	I V	F6	Mean	0	2.4
F10	6	3	Ŷ		Std	0	0.6992
F11	8	4	Ŷ		Best	0	0
F12	8	4	Ŷ	F7	Worst	0	0
F13	10	5	Y		Mean	0	0
F14	10	5	Y		Best	0	0
F15	10	10	Y	70	Worst	0	0
				F8	Mean	0	0.2
	V. EXPERIMEN	TS AND RESULTS			Std	0	0.4216
					Best	1	2
E the the		7 1 : 1 41	· · · ·	F9	Worst	0	0
For the sim	ulations, Matlab	7.1 is used as the	programming		Mean	0.1	I.I 0.8756
language. Th	e configurations	s of the comp	uter are Intel		Best	0.5102	0.8750
Pentium® 4 CPU 3.00 GHZ, 2 GB of memory. As the test				F10	Worst	0	0
problems are relatively complex and the number of optima is					Mean	0	0
large, a large	population size s	should be used.	The population		Std	0	0
size is set 60	0 for $D=10$ and 1	1200 for D=30.	The maximum		Best	1	1
number of ge	eneration is 500	for <i>D</i> =10 and 10	000 for <i>D</i> =30.	F11	Worst	0	1
Therefore, the	e maximum numb	per of function e	valuations will		Std	0.4	1
be the popula	ation size multipl	lied by number	of generations		Best	0	1
for both aloo	rithms The nara	meters used in	the algorithms	E10	Worst	0	0
are list as held	numis. The pure	inclus used in	the argorithms	F12	Mean	0	0.2
are list as bei	<i>F</i> -0.0	CP = 0.1			Std	0	0.4216
$\Gamma = 0.9, CK = 0.1,$					Best	0	1
Two experiments are conducted as follows:			F13	Worst	0	0	
1. $D=10$, Test Functions: F1-F15					Std	0	0.4216
2. $D=30$, Test Functions: F1-F5					Best	0	0.4210
For comparison, the following two criteria are used:			F14	Worst	0	0	
1. Numb	er of optima four	nd [18]		F14	Mean	0	0
2. The b	est value found				Std	0	0
An optimum	is considered t	to be found if	there exists a		Best	0	2
solution in t	he population w	vithin the tolera	ted Euclidean	F15	worst	0	0
distance to that optimum. The tolerance for all problems is set					Std	0	0.7071
to 0.1. All p	roblems are run	for 25 times. T	he results are		514	v	0.7071

Table IV. Comparison of best value found (D=10)

$\begin{array}{c c c c c c c c c c c c c c c c c c c $	Test Function		Crowding DE	ERTS-DE
Best -0.3218 -0.1777 Mean -1.1618 -0.4367 Std 0.4773 0.1912 Best -0.0480 -0.0074 P Worst -0.1316 -0.0339 F2 Mean -0.0911 -0.0239 Best -0.0907 -0.0306 F3 Worst -0.3566 -0.0904 Best -0.64690 -11.5970 F4 Worst -39.4740 -27.5910 F4 Worst -0.2112 -0.0832 Best -0.0999 -0.0137 Std 0.0345 0.0215 Best -2.2706 -0.1270 F6 Worst -0.1222 -0.0337 Std 0.0345 0.0215 Best -2.2706 -0.1270 F6 Worst -1.14.7600 -18.0540 Mean -4.6309 -0.5758 Std 1.2340 0.2838 Best -7.2632 -1.8996				
F1 Worst Mean -1.9420 -1.1618 -0.8395 -0.4367 Std 0.4773 0.1912 Best -0.0480 -0.0074 P Worst -0.1316 -0.0339 Std 0.0272 0.0077 Best -0.0987 -0.0306 F3 Worst -0.3566 -0.0904 Mean -0.1954 -0.0615 Std 0.0834 0.0225 Best -26.4690 -11.5970 Mean -31.7634 -1.84755 Mean -31.7634 -1.84755 Std 0.0345 0.0215 Best -0.0999 -0.0137 Worst -0.2112 -0.0832 Mean -1.122 -0.0377 Std 0.0345 0.0215 Best -2.2706 -0.1270 F6 Mean -4.6309 -0.5758 Std 1.2340 0.2838 Best -7.2632 -1.18093 Std <td< td=""><td></td><td>Best</td><td>-0.3218</td><td>-0.1777</td></td<>		Best	-0.3218	-0.1777
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	F1	Worst	-1.9420	-0.8395
Std 0.4773 0.1912 Best -0.0480 -0.0074 Worst -0.1316 -0.0339 Mean -0.0911 -0.0239 Std 0.0272 0.0077 Best -0.03566 -0.0904 Mean -0.1954 -0.0615 Std 0.0887 -0.0306 F3 Best -26.4690 -11.5970 F4 Worst -39.4740 -27.5910 F4 Worst -39.4740 -27.5910 Best -0.0999 -0.0137 Std 4.5066 5.5951 Best -0.222 -0.0832 Mean -0.122 -0.0832 Best -2.2706 -0.1270 F6 Mean -4.6309 -0.5758 Std 1.2340 0.2838 Best -7.2063 -1.8996 F7 Mean -4.6309 -0.5758 Std 2.01200 -5.8679 Mean <	11	Mean	-1.1618	-0.4367
Best -0.0480 -0.0074 Worst -0.1316 -0.0339 Std 0.0272 0.0077 Best -0.0987 -0.0306 F3 Worst -0.3566 -0.0904 Mean -0.1954 -0.0615 Std 0.0834 0.0225 Best -264690 -11.5970 F4 Worst -39.4740 -27.5910 Mean -31.7634 -18.4755 Std 0.0215 -0.0999 Best -0.0999 -0.0137 Best -0.2112 -0.0832 Mean -0.1292 -0.0377 Std 0.0345 0.0215 Best -2.2706 -0.1270 F6 Worst -114.7600 -18.0540 Mean -46.309 -0.5758 Std 1.2340 0.2838 Best -7.2632 -1.8996 Worst -20.1200 -5.8679 F8 Std 2.02045 -3.6100 <t< td=""><td></td><td>Std</td><td>0.4773</td><td>0.1912</td></t<>		Std	0.4773	0.1912
F2 Worst Mean -0.0316 -0.0239 -0.0239 -0.0077 Best -0.0987 -0.0306 F3 Worst Mean -0.3566 -0.0904 Best -0.26,4690 -11.5970 F4 Worst -39.4740 -27.5910 Mean -31.7634 -18.4755 Std 4.5066 5.5951 Best -0.0999 -0.0137 Worst -0.2112 -0.0832 Mean -0.1292 -0.0377 Std 0.0345 0.0215 Best -0.2706 -0.1270 F6 Worst -6.5206 -0.9615 Mean -4.6309 -0.5758 Std 1.2340 0.22838 Best -7.2632 -1.8996 F7 Worst -20.1200 -5.8679 Mean -4.60644 -11.08933 Std 20.8064 4.8449 Best -7.2632 -1.8996 F8 Worst -20.1200		Best	-0.0480	-0.0074
Mean -0.0911 -0.0237 Best -0.0987 -0.0306 F3 Worst -0.3566 -0.0904 Mean -0.1954 -0.0615 Std 0.0225 Best -26.4690 F4 Worst -39.4740 -27.5910 F4 Worst -39.4740 -27.5910 Best -0.0999 -0.0137 Std 4.5066 5.5951 Best -0.2112 -0.0832 Mean -0.1220 -0.0376 Mean -0.1270 -0.0375 Std 0.0345 0.0215 Best -2.2706 -0.1270 F6 Worst -6.5206 -0.9615 Mean -4.6309 -0.5758 Std 1.2340 0.2838 Best -7.2632 -1.8996 Worst -20.1200 -5.8679 Mean -13.1706 -3.7509 Std 2.0342 0.6195 Best	F2	Worst	-0.1316	-0.0339
$F3 = \begin{bmatrix} 3 \text{ best} & -0.097 \\ \text{Worst} & -0.3566 & -0.0904 \\ \text{Mean} & 0.1954 & -0.0615 \\ \text{Std} & 0.0834 & 0.0225 \\ \text{Best} & -26.4690 & -11.5970 \\ \text{Mean} & -31.7634 & -18.4755 \\ \text{Std} & 4.5066 & 5.5951 \\ \text{Best} & -0.0999 & -0.0137 \\ \text{Std} & 0.0345 & 0.0215 \\ \text{Best} & -0.0999 & -0.0137 \\ \text{Std} & 0.0345 & 0.0215 \\ \text{Best} & -2.2706 & -0.1270 \\ \text{F6} & Worst & -6.5206 & -0.9615 \\ \text{Mean} & -4.6309 & -0.5758 \\ \text{Std} & 1.2340 & 0.2838 \\ \text{Best} & -43.1750 & -3.6100 \\ \text{F7} & Worst & -114.7600 & -18.0540 \\ \text{Mean} & -4.66064 & -11.0893 \\ \text{Std} & 20.8064 & 4.8449 \\ \text{Best} & -7.2632 & -1.8996 \\ \text{Worst} & -0.1202 & -5.8670 \\ \text{Worst} & -10.0240 & -2.6185 \\ \text{Worst} & -10.0240 & -2.6185 \\ \text{F8} & Worst & -10.0240 & -2.6185 \\ \text{F9} & Mean & -5.9759 & -1.7434 \\ \text{Std} & 2.0342 & 0.6195 \\ \text{Best} & -2.7016 & -0.7779 \\ \text{F9} & Morean & -5.9759 & -1.7434 \\ \text{Std} & 2.0342 & 0.6195 \\ \text{Best} & -12.12510 & -1.6850 \\ \text{F10} & Mean & -13.1706 & -3.37509 \\ \text{Std} & 3.7754 & 1.2317 \\ \text{Best} & -2.7016 & -0.7779 \\ \text{F9} & Worst & -10.0240 & -2.6185 \\ \text{F10} & Mean & -13.1706 & -3.7509 \\ \text{Std} & 3.7754 & 1.2317 \\ \text{Best} & -2.7016 & -0.7779 \\ \text{F9} & Worst & -10.0240 & -2.6185 \\ \text{F10} & Mean & -13.1706 & -3.37509 \\ \text{Std} & 3.7754 & 1.2317 \\ \text{Best} & -2.7016 & -0.7779 \\ \text{F9} & Worst & -10.0240 & -2.6185 \\ \text{F10} & Mean & -13.1706 & -3.7509 \\ \text{Std} & 3.7754 & 1.2317 \\ \text{Best} & -2.7016 & -0.7779 \\ \text{F10} & Mean & -12.0240 & -2.6185 \\ \text{F10} & Mean & -13.1706 & -3.7509 \\ \text{F10} & Mean & -13.1706 & -3.7509 \\ \text{Std} & 6.55433 & 0.5745 \\ \text{Best} & -1.1.2310 & -2.6533 \\ \text{F10} & Mean & -12.0240 & -2.6185 \\ \text{F10} & Mean & -12.0240 & -2.6185 \\ \text{Std} & 6.65463 & -0.5305 \\ \text{Std} & 5.6613 & 1.1223 \\ \text{F11} & Mean & -12.650 & -7.5710 \\ \text{Mean} & -12.650 & -7.5710 \\ \text{Mean} & -12.6305 & -2.0342 \\ \text{Std} & 5.6613 & 1.1223 \\ \text{F13} & Morst & -3.00480 & -12.7220 \\ \text{A6686} & 3.2660 \\ \text{Best} & -3.81120 & -3.2552 \\ \text{F14} & Mean & -21.6305 & -2.9632 \\ \text{Std} & 16.0574 & 56.4823 \\ \text{Std} & 16.0574 & 56.4823 \\ \text{Std} & 16.0574 & 56$		Mean	-0.0911	-0.0239
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		Std	0.0272	0.0077
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Best	-0.0987	-0.0306
$F4 = \begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3$	F3	Mean	-0.5500	-0.0904
$F4 = \begin{bmatrix} 3.43 \\ 0.0024 \\ 0.002$		Std	-0.1934	-0.0015
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		Best	-26 4690	-11 5970
F4 Mean Mean Std -31.7634 4.5066 -18.4755 5.5951 Best -0.0999 -0.0137 Best -0.2112 -0.0832 Mean -0.1292 -0.0377 Std 0.0345 0.0215 Best -2.2706 -0.1270 F6 Worst -6.5206 -0.9615 Mean -4.6309 -0.5758 Std 1.2340 0.2838 Best -43.1750 -3.6100 F7 Worst -114.7600 -18.0540 Mean -64.6664 -11.0893 Std Std 20.8064 4.8449 Best -7.2632 -1.8996 F8 Worst -20.1200 -5.8679 Mean -5.9759 -1.7434 Std 2.0342 0.6195 Best -21.2510 -1.6850 F10 Worst -40.4930 -3.6430 Mean -2.95759 -1.7434 Std 2.0210 -4.8216		Worst	-39 4740	-27 5910
$F_{1} = \begin{bmatrix} Std & 4.5066 & 5.5951 \\ Best & -0.0999 & -0.0137 \\ Worst & -0.2112 & -0.0832 \\ Mean & -0.1292 & -0.0377 \\ Std & 0.0345 & 0.0215 \\ Best & -2.2706 & -0.1270 \\ \hline & Worst & -6.5206 & -0.9615 \\ Mean & -4.6309 & -0.5758 \\ Std & 1.2340 & 0.2838 \\ Best & -43.1750 & -3.6100 \\ \hline & Worst & -114.7600 & -18.0540 \\ Mean & -64.6664 & -11.0893 \\ Std & 20.8064 & 4.8449 \\ Best & -7.2632 & -1.8996 \\ \hline & Worst & -20.1200 & -5.8679 \\ Mean & -13.1706 & -3.7509 \\ Std & 3.7754 & 1.2317 \\ Best & -2.7016 & -0.7779 \\ \hline & Worst & -10.0240 & -2.6185 \\ \hline & Mean & -5.9759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ \hline & Best & -2.12510 & -1.6850 \\ \hline & Worst & -40.4930 & -3.6430 \\ \hline & Mean & -29.6469 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ \hline & F10 & Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ \hline & F11 & Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ \hline & Worst & -20.9290 & -11.0240 \\ \hline & Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -3.9807 & -1.1163 \\ \hline & Worst & -20.2200 & -4.8248 \\ \hline & Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ \hline & F12 & Morst & -30.0480 & -12.7220 \\ \hline & Mean & -23.6060 & -6.2305 \\ \hline & Std & 6.6086 & 3.2660 \\ \hline & Best & -38.1120 & -3.2552 \\ \hline & F14 & Mean & -56.5465 & -30.5935 \\ \hline & Std & 16.0574 & 56.4823 \\ \hline & Best & -9.4756 & -1.2842 \\ \hline & F15 & Worst & -46.7710 & -5.0021 \\ \hline & Mean & -21.6305 & -2.9632 \\ \hline & Std & 11.6107 & 1.0729 \\ \hline \end{bmatrix}$	F4	Mean	-31.7634	-18.4755
$F5 = \begin{array}{c cccc} Best & -0.0999 & -0.0137 \\ Worst & -0.2112 & -0.0832 \\ Mean & -0.1292 & -0.0377 \\ Std & 0.0345 & 0.0215 \\ Best & -2.2706 & -0.1270 \\ \hline \\ P6 & Worst & -6.5206 & -0.9615 \\ Mean & -4.6309 & -0.5758 \\ Std & 1.2340 & 0.2838 \\ Best & -43.1750 & -3.6100 \\ Worst & -114.7600 & -18.0540 \\ \hline \\ F7 & Mean & -64.6664 & -11.0893 \\ Std & 20.8064 & 4.8449 \\ Best & -7.2632 & -1.8996 \\ \hline \\ Worst & -20.1200 & -5.8679 \\ Mean & -13.1706 & -3.7509 \\ Std & 3.7754 & 1.2317 \\ Best & -2.7016 & -0.7779 \\ F8 & Mean & -59759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ F8 & Mean & -5.9759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ Best & -21.2510 & -1.6850 \\ \hline \\ F9 & Morst & -40.4930 & -3.6430 \\ Mean & -59746 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ F10 & Worst & -20.9890 & -11.0240 \\ Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ \hline \\ F12 & Worst & -20.9890 & -11.0240 \\ Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ \hline \\ F12 & Worst & -20.2220 & -4.8248 \\ Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ \hline \\ F13 & Morst & -30.0480 & -12.7220 \\ \hline \\ F14 & Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ \hline \\ F14 & Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ \hline \end{array}$		Std	4.5066	5.5951
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Best	-0.0999	-0.0137
$\begin{array}{c ccccc} F5 & Mean & -0.1292 & -0.0377 \\ Std & 0.0345 & 0.0215 \\ Best & -2.2706 & -0.1270 \\ Worst & -6.5206 & -0.9615 \\ Mean & -4.6309 & -0.5788 \\ Std & 1.2340 & 0.2838 \\ Best & -43.1750 & -3.6100 \\ F7 & Worst & -114.7600 & -18.0540 \\ Mean & -64.6664 & -11.0893 \\ Std & 20.8064 & 4.8449 \\ Best & -7.2632 & -1.8996 \\ Worst & -20.1200 & -5.8679 \\ Mean & -13.1706 & -3.7509 \\ Std & 3.7754 & 1.2317 \\ Best & -2.7016 & -0.7779 \\ F8 & Mean & -5.9759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ Best & -21.2510 & -1.6850 \\ Worst & -40.4930 & -3.6430 \\ F10 & Mean & -29.6469 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ F10 & Mean & -17.7898 & -5.7436 \\ Std & 2.0380 & -11.0240 \\ Mean & -17.7898 & -5.7436 \\ Std & 5.6613 & 1.1223 \\ Best & -3.9807 & -1.1163 \\ F12 & Worst & -20.2220 & -4.8248 \\ Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -3.9807 & -1.1163 \\ F12 & Worst & -30.0480 & -12.7220 \\ Hean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ F13 & Worst & -30.0480 & -12.7220 \\ F13 & Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 & Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array}$	5.5	Worst	-0.2112	-0.0832
$F10 = \begin{array}{ccccccccccccccccccccccccccccccccccc$	F5	Mean	-0.1292	-0.0377
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		Std	0.0345	0.0215
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		Best	-2.2706	-0.1270
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	F6	Worst	-6.5206	-0.9615
$F7 = \begin{array}{ccccccccccccccccccccccccccccccccccc$	10	Mean	-4.6309	-0.5758
$F7 \qquad \begin{array}{c ccccccccccccccccccccccccccccccccccc$		Std	1.2340	0.2838
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Best	-43.1750	-3.6100
$F10 = Mean64.060411.0893 \\ Std 20.8064 - 4.8449 \\ Best -7.2632 - 1.8996 \\ Worst -20.1200 -5.8679 \\ Mean -13.1706 -3.7509 \\ Std 3.7754 - 1.2317 \\ Best -2.7016 -0.7779 \\ F9 = Morst -10.0240 -2.6185 \\ Mean -5.9759 - 1.7434 \\ Std 2.0342 - 0.6195 \\ Best -21.2510 - 1.6850 \\ Mean -29.6469 -2.5746 \\ Std 6.5543 - 0.5745 \\ Best -14.2310 - 2.6533 \\ F10 = Morst -40.4930 - 3.6430 \\ Mean -29.6469 -2.5746 \\ Std 6.5543 - 0.5745 \\ Best -14.2310 - 2.6533 \\ F11 = Morst -20.9890 - 11.0240 \\ Mean -17.7898 -5.7436 \\ Std 2.2209 - 2.3804 \\ Best -3.9807 - 1.1163 \\ F12 = Morst -20.2220 - 4.8248 \\ Mean -14.1562 - 2.0342 \\ Std 5.6613 - 1.1223 \\ Best -9.2783 - 3.0312 \\ F13 = Morst -30.0480 - 12.7220 \\ F13 = Morst -30.0480 - 12.7220 \\ Mean -23.6060 -6.2305 \\ Std 6.6086 - 3.2660 \\ Best -38.1120 - 3.2552 \\ F14 = Morst -81.2650 - 75.5100 \\ Mean -21.6305 -2.9632 \\ Std 10.60574 -56.4823 \\ Best -9.4756 - 1.2842 \\ F15 = Morst -46.7710 -5.0021 \\ Mean -21.6305 -2.9632 \\ Std -11.6107 - 1.0729 \\ $	F7	Worst	-114.7600	-18.0540
$F8 = \begin{bmatrix} 3.1d & 20.8064 & 4.8449 \\ Best & -7.2632 & -1.8996 \\ Worst & -20.1200 & -5.8679 \\ Mean & -13.1706 & -3.7599 \\ Std & 3.7754 & 1.2317 \\ Best & -2.7016 & -0.7779 \\ F9 & Worst & -10.0240 & -2.6185 \\ Mean & -5.9759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ Best & -21.2510 & -1.6850 \\ Worst & -40.4930 & -3.6430 \\ Mean & -29.6469 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ F10 & Morst & -20.9890 & -11.0240 \\ Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ F12 & Worst & -20.2220 & -4.8248 \\ Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ F13 & Morst & -30.0480 & -12.7220 \\ F13 & Morst & -30.0480 & -12.7220 \\ F14 & Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 & Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{bmatrix}$		Mean	-64.6664	-11.0893
$F8 = \begin{bmatrix} -7.2632 & -1.8996 \\ Worst & -20.1200 & -5.8679 \\ Mean & -13.1706 & -3.7599 \\ Std & 3.7754 & 1.2317 \\ Best & -2.7016 & -0.7779 \\ F9 & Worst & -10.0240 & -2.6185 \\ Mean & -5.9759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ Best & -21.2510 & -1.6850 \\ Worst & -40.4930 & -3.6430 \\ Mean & -29.6469 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ F10 & Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ F11 & Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ F12 & Worst & -20.2220 & -4.8248 \\ F12 & Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ F13 & Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -3.81.120 & -3.2552 \\ F14 & Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{bmatrix}$		Std	20.8064	4.8449
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Worst	-7.2032	-1.8990
$F9 \qquad \begin{array}{c ccccc} & & & & & & & & & & & & & & & & &$	F8	Mean	-20.1200	-3.8079
$F9 \qquad \begin{array}{c cccc} Best & -2.7016 & -0.7779 \\ Worst & -10.0240 & -2.6185 \\ Mean & -5.9759 & -1.7434 \\ Std & 2.0342 & 0.6195 \\ Best & -21.2510 & -1.6850 \\ Worst & -40.4930 & -3.6430 \\ Mean & -29.6469 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ F11 & Worst & -20.9890 & -11.0240 \\ Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ Best & -3.9807 & -1.1163 \\ F12 & Worst & -20.2220 & -4.8248 \\ Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ F13 & Worst & -30.0480 & -12.7220 \\ F13 & Worst & -30.0480 & -12.7220 \\ F14 & Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 & Worst & -81.2650 & -75.5100 \\ Mean & -23.6065 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array}$		Std	3 7754	1 2317
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		Best	-2.7016	-0.7779
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		Worst	-10.0240	-2.6185
$F10 \qquad \begin{array}{ccccccccccccccccccccccccccccccccccc$	F9	Mean	-5.9759	-1.7434
$F10 \qquad \begin{array}{c cccc} Best & -21.2510 & -1.6850 \\ Worst & -40.4930 & -3.6430 \\ Mean & -29.6469 & -2.5746 \\ Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ F11 & Worst & -20.9890 & -11.0240 \\ Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ Worst & -20.2220 & -4.8248 \\ Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ F12 & Worst & -30.0480 & -12.7220 \\ Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 & Worst & -81.2650 & -75.5100 \\ Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array}$		Std	2.0342	0.6195
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Best	-21.2510	-1.6850
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	E10	Worst	-40.4930	-3.6430
$F11 \qquad \begin{array}{cccccc} Std & 6.5543 & 0.5745 \\ Best & -14.2310 & -2.6533 \\ \hline Worst & -20.9890 & -11.0240 \\ \hline Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ \hline Worst & -20.2220 & -4.8248 \\ \hline Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ \hline Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ \hline F13 & Worst & -30.0480 & -12.7220 \\ \hline Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ \hline F14 & Worst & -81.2650 & -75.5100 \\ \hline Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ \hline F15 & Worst & -46.7710 & -5.0021 \\ \hline Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array}$	F10	Mean	-29.6469	-2.5746
$F11 \qquad \begin{array}{c cccc} Best & -14.2310 & -2.6533 \\ \hline Worst & -20.9890 & -11.0240 \\ \hline Mean & -17.7898 & -5.7436 \\ Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ \hline Worst & -20.2220 & -4.8248 \\ \hline Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ \hline Std & 5.6613 & 1.1223 \\ \hline Best & -9.2783 & -3.0312 \\ \hline F13 & Worst & -30.0480 & -12.7220 \\ \hline Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ \hline F14 & Worst & -81.2650 & -75.5100 \\ \hline Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ \hline F15 & Worst & -46.7710 & -5.0021 \\ \hline Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \hline \end{array}$		Std	6.5543	0.5745
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Best	-14.2310	-2.6533
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	F11	Worst	-20.9890	-11.0240
$F12 \qquad \begin{array}{cccccccc} Std & 2.2209 & 2.3804 \\ Best & -3.9807 & -1.1163 \\ Worst & -20.2220 & -4.8248 \\ Mean & -14.1562 & -2.0342 \\ Std & 5.6613 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ Worst & -30.0480 & -12.7220 \\ Mean & -23.6060 & -6.086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 & Worst & -81.2650 & -75.5100 \\ Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array}$		Mean	-17.7898	-5.7436
$F12 \qquad \begin{array}{c c c c c c c c c c c c c c c c c c c $		Std	2.2209	2.3804
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Best	-3.9807	-1.1163
$F13 \qquad \begin{array}{ccccccccccccccccccccccccccccccccccc$	F12	Worst	-20.2220	-4.8248
$F13 \begin{array}{ccccc} Std & 5.6013 & 1.1223 \\ Best & -9.2783 & -3.0312 \\ Worst & -30.0480 & -12.7220 \\ Mean & -23.6060 & 6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 \begin{array}{cccccc} Worst & -81.2650 & -75.5100 \\ Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 \begin{array}{cccccccccccccccccccccccccccccccccccc$		Mean	-14.1562	-2.0342
$ \begin{array}{c ccccc} F13 & Worst & -30.0480 & -12.7220 \\ Worst & -30.0480 & -12.7220 \\ Mean & -23.6060 & -6.2305 \\ Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ F14 & Worst & -81.2650 & -75.5100 \\ Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array} $		Baat	3.0015	2 0212
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		Worst	-9.2783	-3.0312
$F14 \begin{array}{c ccccc} Std & 6.6086 & 3.2660 \\ Best & -38.1120 & -3.2552 \\ Worst & -81.2650 & -75.5100 \\ Mean & -56.5465 & -30.5935 \\ Std & 16.0574 & 56.4823 \\ Best & -9.4756 & -1.2842 \\ F15 & Worst & -46.7710 & -5.0021 \\ Mean & -21.6305 & -2.9632 \\ Std & 11.6107 & 1.0729 \\ \end{array}$	F13	Mean	-33 6060	-6 2305
F14 Best -38.1120 -3.2552 Worst -81.2650 -75.5100 Mean -56.5465 -30.5935 Std 16.0574 56.4823 Best -9.4756 -1.2842 F15 Worst -46.7710 -5.0021 Mean -21.6305 -2.9632 Std 11.6107 1.0729		Std	6 6086	3 2660
F14 Worst -81.2650 -75.5100 Mean -56.5465 -30.5935 Std 16.0574 56.4823 Best -9.4756 -1.2842 F15 Worst -46.7710 -5.0021 Mean -21.6305 -2.9632 Std 11.6107 1.0729		Best	-38.1120	-3.2552
F14 Mean -56.5465 -30.5935 Std 16.0574 56.4823 Best -9.4756 -1.2842 F15 Worst -46.7710 -5.0021 Mean -21.6305 -2.9632 Std 11.6107 1.0729	T 4 4	Worst	-81.2650	-75.5100
F15 Std 16.0574 56.4823 Best -9.4756 -1.2842 Worst -46.7710 -5.0021 Mean -21.6305 -2.9632 Std 11.6107 1.0729	F14	Mean	-56.5465	-30.5935
Best -9.4756 -1.2842 Worst -46.7710 -5.0021 Mean -21.6305 -2.9632 Std 11.6107 1.0729		Std	16.0574	56.4823
F15 Worst -46.7710 -5.0021 Mean -21.6305 -2.9632 Std 11.6107 1.0729		Best	-9.4756	-1.2842
F1.3 Mean -21.6305 -2.9632 Std 11.6107 1.0729	F15	Worst	-46.7710	-5.0021
Std 11.6107 1.0729	F15	Mean	-21.6305	-2.9632
		Std	11.6107	1.0729

Table V. Comparison of best value found (D=30)

Test Function		Crowding DE	ERTS-DE
	Best	-5.1271	-2.9273
E1	Worst	-7.8151	-4.5686
ГІ	Mean	-6.2561	-3.8060
	Std	0.8332	0.6433
	Best	-2.6091	-1.1209
ED	Worst	-3.7313	-1.6555
ΓZ	Mean	-3.1617	-1.3865
	Std	0.4195	0.1907
	Best	-2.4809	-0.7774
E2	Worst	-4.3563	-2.0588
F3	Mean	-3.7140	-1.6436
	Std	0.6220	0.3615
	Best	-72.2550	-59.2930
E4	Worst	-74.8780	-72.5570
Γ4	Mean	-73.9101	-64.6000
	Std	0.9569	4.5664
	Best	-2.2144	-1.0255
E5	Worst	-3.9658	-1.6979
1.2	Mean	-3.0425	-1.4246
	Std	0.5437	0.1853

VI. CONCLUSION

In this paper, differential evolution algorithm with an ensemble of restricted tournament selection-based niching algorithm is proposed to overcome the difficulty of choosing window size parameter when solving multi-modal optimization problems. The proposed algorithm is compared with the Crowding-DE on a set of newly designed scalable multi-modal problems. As we can see from the result, the proposed algorithm outperforms the Crowding-DE on all the test problems.

REFERENCES

- S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, Urbana, IL, USA, 1995. [Online]. Available: citeseer.ist.psu.edu/mahfoud95niching.html.
- [2] K. Koper and M. Wysession, "Multimodal function optimization with a niching genetic algorithm: A seis-mological example," *Bulletin of the Seismological Society of America*, vol. 89, pp. 978-988, 1999.
- [3] D. J. Cavicchio, "Adaptive search using simulated evolution," Ph.D. dissertation, University of Michigan, Ann Arbor, 1970.
- [4] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems, Ph.D. dissertation, University of Michigan, 1975.
- [5] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms", Proc. of the IEEE Int. Conf. on Evolutionary Computation, New York, USA, 1996, pp. 798–803.
- [6] K. Price. "An introduction to differential evolution." *New Ideas in Optimization*, pages 79-108, 1999.
 [7] R. Thomsen. "Multi-modal optimization using crowding-based
- [7] R. Thomsen. "Multi-modal optimization using crowding-based differential evolution. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volumn 2, pages 1382-1389, 2004.
- [8] D. Zaharie. "Extensions of differential evolution algorithms for multimodal optimization," In Proceedings of SYNASC'04, 6th International Symposium of Symbolic and Numeric Algorithms for Scientific Computing, pages 523 [534, 2004.

- [9] Z. Hendershot. "A differential evolution algorithm for automatically discovering multiple global optima in multidimensional, discontinues spaces. In Proceedings of MAICS 2004, Fifteenth Midwest Artificial Intelligence and Cognitive Sciences Conference, pages 92{97, 2004.
- [10] R. Storn and K. V. Price, "Differential evolution-Asimple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1995.
- [11] S. Das, A. Konar and U. K. Chakraborty, "Annealed differential evolution," In Proc. IEEE Congress on Evolutionary Computation, Singapore, 2007.
- [12] U. K. Chakraborty, S. Das and A. Konar, "Differential evolution with local neighborhood, In *Proc. IEEE Congress on Evolutionary Computation*, Canada, 2006.
- [13] V. L. Huang, A. K. Qin and P. N. Suganthan, "Self-adaptive Differential Evolution Algorithm for Constrained Real-Parameter Optimization", In Proc. IEEE Congress on Evolutionary Computation, Canada, 2006.
- [14] X. Q. Chen, Z. X. Hou, and J. X. Liu, "Multi-objective optimization with modified pareto differential evolution," in *International Conference on Intelligent Computation Technology and Automation*, Piscataway, NJ 08855-1331, United States, 2008.
- [15] G. Harik, "Finding multiple solutions in problems of bounded difficulty," University of Illinois at Urbana-Champaign, 1994, IIIiGAL Report No. 94002.
- [16] D H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1997) 67–82.
- [17] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. –P. Chen, A. Auger and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Technical Report*, Nanyang Technological University, May 2005.
- [18] J. Gan, K. Warwick, "A variable radius niching technique for speciation in genetic algorithms," In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2000)*, San Francisco, USA, 2000, pp. 96–103.

<u>Appendix</u> (not to publish, just to assist the review process)

Test Function Set 1

F1: Two-Peak Trap

$$f_1(x) = \begin{cases} \frac{160}{15}(15-x), & \text{for } 0 \le x \le 15\\ \frac{200}{5}(x-15), & \text{for } 15 \le x \le 20 \end{cases}$$

Range: $0 \le x \le 20$

F2: Central Two-Peak Trap

$$f_2(x) = \begin{cases} \frac{160}{10}x, & \text{for } 0 \le x \le 10\\ \frac{160}{5}(15-x) & \text{for } 10 \le x \le 15\\ \frac{200}{5}(x-15), & \text{for } 15 \le x \le 20 \end{cases}$$

Range: $0 \le x \le 20$

F3: Five-Uneven-Peak Trap

$$f_3(x) = \begin{cases} 80(2.5-x) & \text{for } 0 \le x < 2.5 \\ 64(x-2.5) & \text{for } 2.5 \le x < 5 \\ 64(7.5-x) & \text{for } 5 \le x < 7.5 \\ 28(x-7.5) & \text{for } 7.5 \le x < 12.5 \\ 28(17.5-x) & \text{for } 12.5 \le x < 17.5 \\ 32(x-17.5) & \text{for } 17.5 \le x < 22.5 \\ 32(27.5-x) & \text{for } 22.5 \le x < 27.5 \\ 80(x-27.5) & \text{for } 27.5 \le x \le 30 \end{cases}$$

Range: $0 \le x \le 20$

F4: Equal Maxima

 $f_4(x) = \sin^6(5\pi x)$

Range: $0 \le x \le 1$

F5: Decreasing Maxima

$$f_5(x) = \exp[-2\log(2) \cdot (\frac{x-0.1}{0.8})^2] \cdot \sin^6(5\pi x)$$

Range: $0 \le x \le 1$

F6: Uneven Maxima

$$f_6(x) = \sin^6(5\pi(x^{3/4} - 0.05))$$

Range: $0 \le x \le 1$

F7: Uneven Decreasing Maxima

$$f_7(x) = \exp[-2\log(2) \cdot (\frac{x - 0.08}{0.854})^2] \cdot \sin^6(5\pi (x^{3/4} - 0.05))$$

Range: $0 \le x \le 1$

F8: Himmelblau's function

$$f_8(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$$

Range:
$$-6 \le x, y \le 6$$

F9: Six-Hump Camel Back

$$f_9(x, y) = -4[(4-2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4+4y^2)y^2]$$

Range: $-1.9 \le x \le 1.9;$ $-1.1 \le y \le 1.1$

F10: Shekel's foxholes

$$f_{10}(x, y) = 500 - \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1 + i + (x - a(i))^6 + (y - b(i))^6}}$$

where $a(i) = 16(i \mod 5) - 2$, and $b(i) = 16(\lfloor (i/5) \rfloor - 2)$

Range: $-65.536 \le x, y \le 65.535$

F11: 2D Inverted Shubert function

$$f_{11}(\vec{x}) = -\prod_{i=1}^{2} \sum_{j=1}^{5} j \cos[(j+1)x_i + j]$$

Range: $-10 \le x_1, x_2 \le 10$

F12-14: Inverted Vincent function

$$f(\vec{x}) = \frac{1}{n} \sum_{i=1}^{n} \sin(10.\log(x_i))$$

where *n* is the dimesnion of the problem

Range: $0.25 \le x_i \le 10$

Test Function Set 2

The set 2 composition function are defined as follow:

- F(x): new composition function
- $f_i(x)$: ith basic function used to construct the composition

function.

- *n:* number of basic functions (number of optima)
- *D*: dimensions (can be chosen from 1-100)
- M_i : linear transformation matrix for each $f_i(x)$
- o_i : new shifted optima position for each $f_i(x)$

$$F(x) = \sum_{i=1}^{n} \left\{ w_i * [f_i'((x - o_i) / \lambda_i * M_i)] \right\}$$

 w_i : weight value for each $f_i(x)$, calculated as follow:

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

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

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

- σ_i : used to control each $f_i(x)$'s coverage range.
- λ_i : used to stretch compress the function.

 $f_i(x) = C * f_i(x) / |f_{\max i}|$, *C* is a predefined constant.

 $|f_{\max i}|$ is estimated using: $|f_{\max i}| = f_i((x' / \lambda_i) * M_i), x' = [5, 5, ..., 5]$

Composition Function 1 (F15, *n*=8)

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

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

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

$$f_i(x) = \sum_{i=1}^{D} \left(\sum_{k=0}^{k_{\text{max}}} D \sum_{k=0}^{k_{\text{max}}} [a^k \cos(2\pi b^k (x_i + 0.5))] - a = 0.5, b = 3, k_{\text{max}} = 20 \right)$$

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

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

 $f_{7-8}(x)$: Sphere Function

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

 $\sigma_i = 1$ for all *i*

 $\lambda = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32]$

 M_i : are all identity matrices

These formulas are basic functions; shift and rotation should be added to these functions. Take f_1 as an example, the following function should be evaluated:

$$f_i(z) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10)$$

where $z = ((x - o_i) / \lambda_1) * M_1$.

Composition Function 2 (F16 *n*=6)

 $f_{1-2}(x)$: Griewank's Function $f_{3-4}(x)$: Weierstrass Function $f_{5-6}(x)$: Sphere Function $\sigma_i = 1$ for all i

 $\lambda = [1, 1, 10, 10, 5/60, 5/60,]$

 M_i : are all identity matrices

Composition Function 3 (F17 *n*=6)

- $f_{1-2}(x)$: Rastrigin's Function
- $f_{3-4}(x)$: Griewank's Function
- $f_{5-6}(x)$: Sphere Function
- $\sigma_i = 1$ for all *i*
- $\lambda = [1, 1, 10, 10, 5/60, 5/60,]$
- M_i : are all identity matrices

Composition Function 4 (F18 *n*=6)

- $f_{1-2}(x)$: Rastrigin's Function
- $f_{3-4}(x)$: Weierstrass Function

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

 $\sigma_i = 1$ for all *i*

 $\lambda = [1, 1, 10, 10, 5/60, 5/60,]$

 M_i : are all identity matrices

Composition Function 5 (F19 *n*=6)

- $f_{1-2}(x)$: Rastrigin's Function
- $f_{3-4}(x)$: Weierstrass Function
- $f_{5-6}(x)$: Sphere Function
- $\sigma_i = 1$ for all *i*
- $\lambda = [1, 1, 10, 10, 5/60, 5/60,]$
- M_i : are all identity matrices

Composition Function 6 (F20 *n*=6)

$$f_{1-2}(x) : F8F2 \text{ Function}$$

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

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

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

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

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

$$\sigma = [1, 1, 1, 1, 1, 2],$$

 $\lambda = [5*5/100; 5/100; 5*1; 1; 5*1; 1]$

 M_i : are all orthogonal matrix

Composition Function 7 (F21 *n*=6)

 $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function $F(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$

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

 $f_{3-4}(x)$: F8F2 Function

- $f_{5-6}(x)$: Weierstrass Function
- $\sigma \!=\! [1,\!1,\!1,\!1,\!1,\!2]\,,$
- $\lambda = [5;10;5;1;5*5/100;5/100]$
- M_i : are all orthogonal matrix

Composition Function 8 (F22 *n*=6)

- $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function
- $f_{3-4}(x)$: F8F2 Function
- $f_{5-6}(x)$: Griewank's Function
- $\sigma \!=\! [1,\!1,\!1,\!1,\!1,\!2]\,,$
- $\lambda = [5*5/100; 5/100; 5*1; 1; 5*1; 1]$
- M_i : are all orthogonal matrix

Composition Function 9 (F23 *n*=6)

- $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function
- $f_{3-4}(x)$: Weierstrass Function
- $f_{5-6}(x)$: Griewank's Function

 $\sigma = [1, 1, 1, 1, 1, 2],$

 $\lambda = [5;10;5*5/100;5/100;5;1]$

 M_i : are all orthogonal matrix

Composition Function 10 (F24 *n*=6)

- $f_{1-2}(x)$: Rastrigin's Function
- $f_{3-4}(x)$: F8F2 Function
- $f_{5-6}(x)$: Weierstrass Function
- $\sigma = [1, 1, 1, 1, 1, 2],$
- $\lambda = [5;10;5*5/100;5/100;5;1]$
- M_i : are all orthogonal matrix

Composition Function 11 (F25 *n*=8)

- $f_{1-2}(x)$: Rastrigin's Function
- $f_{3-4}(x)$: F8F2 Function
- $f_{5-6}(x)$: Weierstrass Function
- $f_{7-8}(x)$: Griewank's Function
- $\sigma = [1, 1, 1, 1, 1, 2, 2, 2],$
- $\lambda = [5;1;5;1;50;10;5*5/200;5/200]$
- M_i : are all orthogonal matrix

Composition Function 12 (F26 *n*=8)

- $f_{1-2}(x)$: Rotated Expanded Scaffer's F6 Function
- $f_{3-4}(x)$: F8F2 Function
- $f_{5-6}(x)$: Weierstrass Function

 $f_{7-8}(x)$: Griewank's Function

 $\sigma = [1, 1, 1, 1, 1, 2, 2, 2],$

 $\lambda = [5*5/100; 5/100; 5; 1; 5; 1; 50; 10]$

 M_i : are all orthogonal matrix

Composition Function 13 (F27 *n*=10)

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

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

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

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

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

 $\sigma = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2],$

 $\lambda = [5*5/100; 5/100; 5; 1; 5; 1; 50; 10; 5*5/200; 5/200]$

 M_i : are all orthogonal matrix

Composition Function 14 (F28 *n*=10)

All settings are the same as F13, except M_i 's condition numbers are [10 20 50 100 200 1000 2000 3000 4000 5000]

Composition Function 15 (F29 *n*=10)

- $f_1(x)$: Weierstrass Function
- $f_2(x)$: Rotated Expanded Scaffer's F6 Function
- $f_3(x)$: F8F2 Function
- $f_4(x)$: Ackley's Function

$$f_i(x) = -20 \exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e \ f_5(x)$$
: Rastrigin's Function

 $f_6(x)$: Griewank's Function

 $f_7(x)$: Non-Continuous Expanded Scaffer's F6 Function

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

$$f_i(x) = F(y_1, y_2) + F(y_2, y_3) + \dots + F(y_{D-1}, y_D) + F(y_D, y_1)$$

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

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

$f_8(x)$: Non-Continuous Rastrigin's Function

$$f_i(x) = \sum_{i=1}^{D} (y_i^2 - 10\cos(2\pi y_i) + 10)$$
$$y_i = \begin{cases} x_j & |x_j| < 1/2\\ round(2x_j)/2 & |x_j| > 1/2 \end{cases} \text{ for } j = 1, 2, ..., D$$

 $f_9(x)$: High Conditioned Elliptic Function

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

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

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

n=10

 $\sigma_i = 2$ for all *i*

 $\lambda = [10; 5 / 20; 1; 5 / 32; 1; 5 / 100; 5 / 50; 1; 5 / 100; 5 / 100]$

M_i are all rotation matrices, condition number are [100 50 30 10 5 5 4 3 2 2];