2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Memetic Differential Evolution Based on Fitness Euclidean-Distance Ratio

B. Y. Qu[1], J. J. Liang[2]
[1]School of Electric and Information Engineering
Zhongyuan University of Technology
Zhengzhou, China
qby1984@hotmail.com, xiaojunming528@163.com

J. M. Xiao[1], Z. G. Shang[2]
[2]School of Electrical Engineering
Zhengzhou University
Zhengzhou, China
liangjing@zzu.edu.cn, zhigang_shang@zzu.edu.cn

***Abstract*—In this paper, a differential evolution algorithm based on fitness Euclidean-distance ratio which was proposed to maintain multiple peaks in the multimodal optimization problems was modified to solve the complex single objective real parameter optimization problems. With the fitness Euclidean-distance ratio technique, the diversity of the population was kept to enhance the exploration ability. And in order to improve the exploitation ability, the Quasi-Newton method was combined. The performance of the proposed method on the set of benchmark functions provided by CEC2014 competition on single objective real-parameter numerical optimization was reported.**

Keywords—real-parameter optimization; differential evolution, fitness Euclidean-distance ratio, memetic optimization

I. INTRODUCTION

Optimization problems exist in our normal life and engineering areas. These optimization problems can be classified into various categories. According the number of the objectives, there are single objective optimization problems and multi-objective optimization problems [1], [2]. For multi-objective optimization problems, usually we have two or more conflict objectives to be optimized at the same time. According to the existence of equality or inequality constraints, we can classify them into optimization problems with or without constraints [3], [4]. According to the state of the environments, there are stationary optimization problems and dynamic optimization problems [2], [5]. According to the form the required optimal solutions, we can divide them into real parameter optimization problems and discrete optimization problems [6]. Focusing on solving these different types of optimization problems, various optimization algorithms were proposed. While among so many categories of optimization algorithms, research on the single objective optimization algorithms influences the development of these optimization branches mentioned above. In recent years, various kinds of novel optimization algorithms have been proposed to solve real-parameter optimization problems. For the sake of making the comparison different optimization function become easier and fairly. A novel CEC'14 test suite which includes 30 black-box benchmark functions is provided [7].

Differential evolution (DE) is a simple yet effective global optimization technique which was proposed by Storn and Price in 1995 [8]. As the other evolutionary algorithms, differential evolution searches good solutions in the provided search ranges based on the historical search experiences without knowing the explicit equations of the objective optimization. DEs present good performance in solving global optimization problems in terms of convergence speed, accuracy, and robustness. Various modified DEs were proposed to further improve the performance of DEs on complex optimization problems [9][10]. In [11], a differential evolution algorithm based on fitness Euclidean-distance ratio (FERDE) was proposed to solve multimodal optimization problems of which the prime target is to find multiple global and local optima of a problem in one single run. Through comparing with some state-of-the-art multimodal optimization approaches, FERDE present(s) good performance on complex multimodal benchmark functions. Considering the good diversity of the FERDE, it is a promising method for normal complex single objective function, but the local search ability of it needs to be improved. Memetic method is a good choice [12][13]. In order to improve exploitation ability of the original FERDE, the Quasi-Newton method is employed periodically with the currently found good solutions as the start points. We call this improved FERDE as Memetic FERDE (MFERDE). The MERDE was tested on the 10-*D* and 30-*D* test functions provided in CEC2014 competition on single objective real-parameter numerical optimization.

The rest of this paper is organized as follows. Section II gives a brief overview of the original FERDE. Section III introduces the proposed MFERDE in detail. The experimental setup and experimental results are presented and discussed in Section IV. Finally, the paper is concluded in Section V.

II. DIFFERENTIAL EVOLUTION BASED ON FITNESS EUCLIDEAN-DISTANCE RATIO

A. Differential Evolution

As the other evolutionary algorithms, DE is also a population based searching method with a randomly

This work was supported partially by National Natural Science Foundation of China (61305080, U1304602), Postdoctoral Science Foundation of China (20100480859), Specialized Research Fund for the Doctoral Program of Higher Education (20114101110005), Scientific and Technological Project of Henan Province (132102210521, 122300410264), and Key Foundation of Henan Educational Committee (14A410001)

initialization. After the random initialization, the new offspring are constructed using mutation and recombination operators. Then with the selection operator, the next generations are selected. While different from traditional EAs, DE generates mutant vectors by using differences of randomly sampled pairs of individual vectors from the population. The following are five commonly used mutation strategies [14]:

- *"DE/best/1"*

$$v_p = x_{best} + F * rand(1, D) \cdot (x_{r1} - x_{r2}) \quad (1)$$

- *"DE/rand/1"*

$$v_p = x_{r1} + F * rand(1, D) \cdot (x_{r2} - x_{r3}) \quad (2)$$

- *"DE/current-to-best/2"*

$$v_p = x_p + F * rand(1, D) \cdot (x_{best} - x_p + x_{r1} - x_{r2}) \quad (3)$$

- *"DE/best/2"*

$$v_p = x_{best} + F * rand(1, D) \cdot (x_{r1} - x_{r2}) + F * rand(1, D) \cdot (x_{r3} - x_{r4}) \quad (4)$$

- *"DE/rand/2"*

$$v_p = x_{r1} + F * rand(1, D) \cdot (x_{r2} - x_{r3}) + F * rand(1, D) \cdot (x_{r4} - x_{r5}) \quad (5)$$

Here r_1 to r_5 are mutually different integers generated between 1 and *NP* (population size) based on some criteria. *F* is the scale factor used to scale differential vectors. x_{best} is the solution with the best fitness value in the current population.

Then the offspring u_p is generated by applying the crossover operation to the generated mutant vector and its corresponding parent vector:

$$u_{p,d} = \begin{cases} v_{p,d} & \text{if } rand_d \le CR \\ x_{p,d} & \text{otherwise} \end{cases} \quad (6)$$

CR is a user-specified constant in the range of [0, 1] and is used to control the crossover rate. A detailed study about the effects of *NP*, *CR* and *F* on the performance of DE algorithm is presented in [9].

B. Fitness Euclidean-distance Ratio

Fitness Euclidean-distance ratio (FER) was first introduced by Li [15] to solve multi-modal optimization problems. By observing the FER calculation method has some problems for the best individual in the neighborhood, a modified FER calculation equation was proposed in [11] :

$$FER_{(j,i)} = \frac{f(p_j) - f(p_w)}{\|p_j - p_i\|} \quad (7)$$

Here p_j and p_i are the personal best of the j^{th} and i^{th} individual respectively. $f(p_j)$ and $f(p_i)$ are the fitness values of p_j and p_i. Here fitness values are employed. If the problem is a minimization problem, the negative values of the cost function can be assigned as the fitness values. p_w is the worst-fit individual in the current population.

C. Differential Evolution Based on Fitness Euclidean-distance Ratio

In order to keep the diversity of the population, in FERDE x_{r1} to x_{r5} and x_{best} which are used to generate the mutant vector are selected randomly according to the FER values. The individuals with higher FER values will have a higher chance to be selected. The selection probability P_j is calculated as following:

$$P_j = FER_{(j,i)} / \sum_{j=1}^{NP} FER_{(j,i)} \quad , j = 1, 2, ..., N \quad (8)$$

And in order to avoid to chose the current individual x_i and the individuals are too far from the current individual, FER is recalculated as below:

$$FER_{(j,i)} = \begin{cases} 0 & \text{if } j=i \text{ or } \|p_j - p_i\| >= \frac{1}{NP}\sum_{j=1}^{NP}\|p_j - p_i\| \\ FER_{(j,i)} & \text{otherwise} \end{cases} \quad (9)$$

The flowchart of x_{best} and x_{ri} selection strategy is presented in Fig. 1.

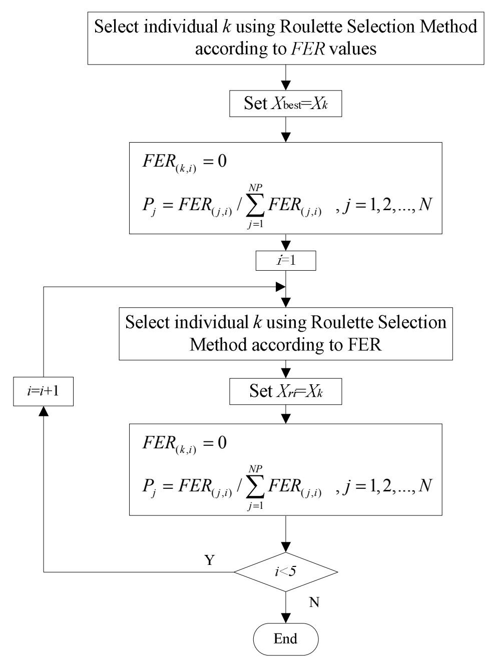

Fig. 1. x_{best} and x_{ri} selection strategy

III. MEMETIC FERDE

A. Modified Fitness Euclidean-distance Ratio

In the previous work, the individual with a high fitness value will be assigned a higher selection probability even it is far from the current individual p_j. For example, if we have four individuals of which the fitness values are 100, 300, 300, and 1000 respectively. The Euclidean distances between them and the current individual are 2, 2, 3, and 5 respectively. $f(p_w)$ is equal to 0. By using (7) and (8), we can get the selection probabilities P for these four individuals are 0.1, 0.2, 0.3, and 0.4 respectively. The individual which has the largest fitness value and the largest distance has the largest probability to be chosen. But our aim is to choose the best individual in the neighborhood. Thus in order to overcome this defect, a modified FER calculation method is proposed in this paper.

In the new method, the differences of fitness values of the individuals and the worst fitness values in the current population are normalized to the range [0.1, 1]. The same operation is executed on the Euclidean distances. We denote $N(x)$ as the normalization operator. Then (7) is modified to:

$$FER_{(j,i)} = \frac{N(f(p_j) - f(p_w))}{N(\|p_j - p_i\|)} \tag{10}$$

Then for same example, the selection probabilities P for these four individuals become 0.1739, 0.5217, 0.1304, and 0.1739 respectively. The third individual which has a good fitness value and is near to the current individual will have a higher chance to be selected. And at the same time, even the bad solutions also have a chance to be selected. In this way, the diversity of is increased.

Different from the multimodal optimization problems, only one global best solution needs to be obtained in one run for the global optimization problems. Thus (9) is modified as following in the new proposed MFERDE:

$$FER_{(j,i)} = \begin{cases} 0 & \text{if } j=i \\ FER_{(j,i)} & \text{otherwise} \end{cases} \tag{11}$$

B. Local Search Strategy

In the original FERDE, in order to maintain all the optima in one run, the fitness value of an offspring is compared with that of the nearest individual in the current population and the fitter one will be kept for following generation. In this way the population is discouraged to converge to a single optimum. While for the global optimization problems, in order to speed up the convergence, a local search strategy which was used in [16] is incorporated in MFERDE:

1) Every L generations, the neighborhood best individuals based on FER values of five randomly chosen individuals are refined using the Quasi-Newton method. Here L is a user defined constant value.

2) In the end of the search, the best solution achieved so far is refined using Quasi-Newton method.

C. Re-initialization Strategy

Since in the MERDE, the nearest individual other than the current individual is replaced if the new generated solution is better, some bad solutions which is far from the population is difficult to be replaced and become dead points. Though the diversity is kept in this way, but dead points far from the global region do not help the search much. In MERDE, the individuals who do not improve for R generations will be re-initialized using "DE/rand/2" strategy with r_1 to r_5 randomly generated between 1 and NP.

D. Bounds handling mechanism

To restrain the search in the predefined search range [xl, xu], the new generated solutions are modified as following before their fitness values are calculated:

$$u_{p,d} = \begin{cases} xl_d + rem(xl_d - u_{p,d}, xu_d - xl_d) & \text{if } u_{p,d} < xl_d \\ u_{p,d} & \text{if } xl_d \le u_{p,d} \le xu_d \\ xu_d - rem(u_{p,d} - xu_d, xu_d - xl_d) & \text{if } u_{p,d} > xu_d \end{cases} \tag{12}$$

E. Memetic FERDE

The flowchart of MFERDE is given in Fig. 2.

IV. EXPERIMENTAL RESULTS

A. Benchmark Function

CEC2014 test suite includes thirty functions with different properties. We can classify them into four groups:

- *Group A: Unimodal functions (F1-F3)*
- *Group B: Simple multi-modal functions (F4-F16)*
- *Group C: Hybrid functions (F17-F22)*
- *Group D: Composition functions (F23-F30)*

For the unimodal function, there is only one optimum for each function. While for the simple multi-modal function, there is one global optimum and many local optima. For the hybrid function, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents. In this way, the different subcomponents of the variables may have different properties. For the composition function, each is composed using several different basic sub-functions which make the composition function have different properties at the different ranges. Especially for F29 and F30, since the basic sub-functions are hybrid functions, except that they have different properties for different areas, the different subcomponents of the variables also have different properties.

The global optima of all thirty functions are shifted to a predefined location. Except F8 and F10, all functions are rotated. Different rotation matrixes are employed for different subcomponents in hybrid functions. And each basic sub-function in the composition functions is also has a separate rotation matrix.

B. Experimental Settings

The performance of MFERDE on 10-D and 30-D benchmark functions is tested. Strategy "DE/rand/1" is adopted. According to [7], the max fitness evaluation times (Max_FES) is set at 100,000 for 10-D, 300,000 for 30-D and 500,000 for 50-D. For each function, the MFERDE is run 51 times. The error value which is smaller than 10^{-8} is treated as 0. The PC configuration and parameters setting are listed below:

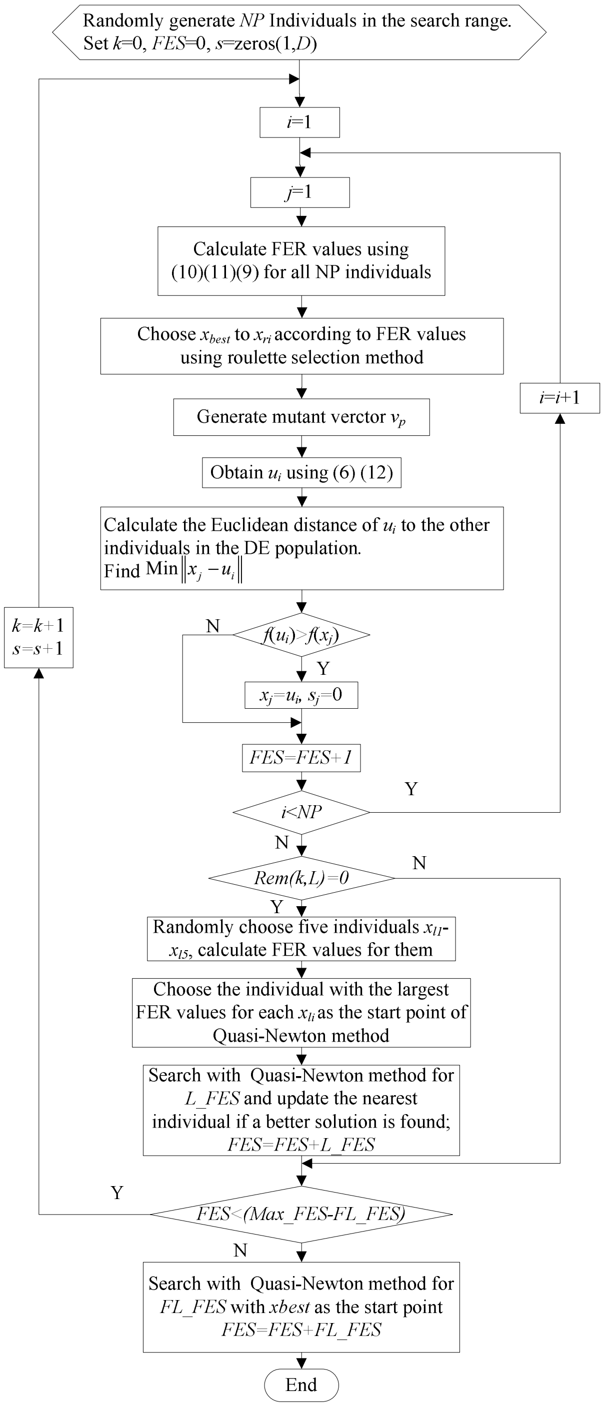

Fig. 2. *Flowchart of MFERDE*

- *PC Configuration:*

System: Windows XP SP3; CPU: 2.93GHz

RAM: 3.46 GB; Language: Matlab R2008a

Algorithm: MERDE

- *Parameters Setting:*

1) Parameters to be adjusted:

F, *CR*, *NP*, *L*, *L_FES*, *FL_FES*

2) Actual parameter values used:

The population size *NP* is set 60, 100 and 150 for 10-*D*, 30-*D* and 50-*D* respectively;

F=0.9, *CR*=0.1, *L*=100, *L_FES*=50**D*, *FL_FES*=100**D*.

C. Experimental Results

The best, worst, median, mean, and standard variance values of function error values achieved within the Max_FEs for the 51 runs are presented in Table I, II and III. The convergence curve of the median run for all the thirty functions for 10-*D*, 30-*D* and 50-*D* are given in Fig. 3-Fig. 17.

TABLE I. RESULTS FOR 10D

F	Best	Worst	Median	Mean	Std
1	3.46e-04	5.36e+01	2.03e-02	1.58	7.61
2	2.57e-07	6.57e-04	1.97e-05	6.31e-05	1.12e-04
3	4.04e-07	5.85e-03	9.41e-04	1.35e-03	1.22e-03
4	0	0	0	0	0
5	7.72	2.00e+01	2.00e+01	1.90e+01	2.71
6	2.74e-01	1.71	8.91e-01	8.93e-01	2.81e-01
7	0	8.37e-02	1.72e-02	1.83e-02	1.51e-02
8	0	0	0	0	0
9	2.98	8.95	5.97	5.58	1.74
10	0	1.25e-01	6.25e-02	3.67e-02	3.98e-02
11	3.75e-01	2.77e+02	3.52e+01	7.55e+01	7.63e+01
12	1.86e-02	3.11e-01	1.02e-01	1.17e-01	6.93e-02
13	2.62e-02	2.03e-01	1.16e-01	1.17e-01	4.43e-02
14	4.33e-02	1.66e-01	9.18e-02	9.37e-02	2.73e-02
15	3.45e-01	1.22	6.43e-01	6.72e-01	2.18e-01
16	3.87e-01	2.57	1.54	1.53	4.63e-01
17	4.44e-01	4.22e+01	3.41	7.93	9.59
18	1.06	6.34	2.40	2.72	1.29
19	1.85e-01	1.02	4.72e-01	5.10e-01	1.76e-01
20	2.39e-01	3.25	1.72	1.70	7.50e-01
21	7.86e-03	1.43e+02	6.61e-01	8.54	2.66e+01
22	3.58e-02	2.16e+01	2.57	3.24	3.96
23	3.29e+02	3.29e+02	3.29e+02	3.29e+02	2.68e-11
24	1.10e+02	1.20e+02	1.14e+02	1.15e+02	2.45
25	1.15e+02	1.55e+02	1.36e+02	1.36e+02	8.34
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02	4.18e-02
27	3.21	3.28e+02	7.86	2.87e+01	7.60e+01
28	3.46e+02	3.80e+02	3.69e+02	3.66e+02	7.37
29	2.38e+02	4.62e+02	3.07e+02	3.17e+02	5.48e+01
30	2.84e+02	6.46e+02	5.31e+02	5.34e+02	6.06e+01

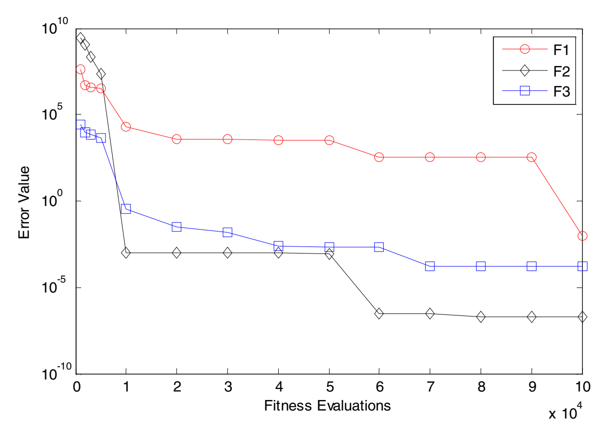

Fig. 3. *Convergence curves for F1-F3 of 10-D*

TABLE II. RESULTS FOR 30D

F	Best	Worst	Median	Mean	Std
1	2.82e-02	2.48e+03	2.69e+02	5.41e+02	6.40e+02
2	1.09e-06	1.33e-02	1.23e-03	2.39e-03	3.24e-03
3	1.04e-05	3.43e-03	1.02e-03	1.13e-03	7.36e-04
4	0	3.99	0	6.25e-01	1.46
5	2.00e+01	2.00e+01	2.00e+01	2.00e+01	7.17e-05
6	1.42e+01	2.21e+01	1.82e+01	1.82e+01	1.72
7	0	0	0	0	0
8	0	0	0	0	0
9	3.48e+01	8.16e+01	5.57e+01	5.50e+01	1.04e+01
10	8.33e-02	7.50	3.33e-01	1.29	1.61
11	1.41e+03	3.74e+03	2.75e+03	2.72e+03	4.69e+02
12	2.14e-01	9.38e-01	5.54e-01	5.64e-01	1.74e-01
13	1.91e-01	3.81e-01	2.85e-01	2.84e-01	4.41e-02
14	1.07e-01	2.69e-01	2.15e-01	2.14e-01	2.69e-02
15	2.25	6.07	3.96	4.14	7.77e-01
16	1.00e+01	1.20e+01	1.13e+01	1.12e+01	4.58e-01
17	4.60e+02	1.92e+03	1.13e+03	1.16e+03	3.73e+02
18	1.22e+01	3.64e+01	2.17e+01	2.24e+01	6.47
19	5.25	9.14	7.75	7.74	7.32e-01
20	7.72	5.22e+01	2.63e+01	2.80e+01	1.06e+01
21	5.60e+01	1.01e+03	5.72e+02	5.99e+02	2.15e+02
22	2.15e+01	2.89e+02	1.43e+02	1.15e+02	7.29e+01
23	3.14e+02	3.14e+02	3.14e+02	3.14e+02	3.74e-09
24	2.23e+02	2.26e+02	2.25e+02	2.25e+02	5.57e-01
25	2.00e+02	2.00e+02	2.00e+02	2.00e+02	2.83e-02
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02	7.61e-02
27	3.46e+02	4.48e+02	3.79e+02	3.84e+02	2.32e+01
28	7.53e+02	8.54e+02	8.03e+02	8.05e+02	2.49e+01
29	9.35e+02	1.42e+03	1.18e+03	1.19e+03	1.06e+02
30	6.27e+02	1.82e+03	1.02e+03	1.07e+03	2.90e+02

TABLE III. RESULTS FOR 50D

F	Best	Worst	Median	Mean	Std
1	6.13e-01	4.86e+03	1.98e+01	3.39e+02	8.57e+02
2	6.92e-05	4.34e-03	5.22e-04	8.39e-04	8.53e-04
3	4.76e-03	4.62e-02	9.20e-03	1.19e-02	8.48e-03
4	0	3.99	0	7.98e-02	5.58e-01
5	2.00e+01	2.00e+01	2.00e+01	2.00e+01	2.21e-05
6	4.00e+01	5.16e+01	4.62e+01	4.62e+01	2.67
7	0.00	0.00	0.00	0.00	0.00
8	1.59e+01	4.18e+01	2.79e+01	2.82e+01	5.77
9	9.35e+01	1.95e+02	1.45e+02	1.42e+02	2.48e+01
10	1.10e+01	8.83e+02	2.41e+02	3.25e+02	2.48e+02
11	4.54e+03	7.41e+03	5.79e+03	5.80e+03	7.11e+02
12	5.09e-01	1.64	8.78e-01	8.96e-01	2.56e-01
13	2.57e-01	5.15e-01	3.83e-01	3.80e-01	6.22e-02
14	1.59e-01	2.65e-01	2.34e-01	2.26e-01	2.14e-02
15	7.24	1.54e+01	9.97	1.06e+01	2.07
16	1.86e+01	2.19e+01	2.08e+01	2.09e+01	5.94e-01
17	1.28e+03	5.30e+03	2.47e+03	2.59e+03	7.99e+02
18	4.65e+01	1.32e+02	7.54e+01	7.92e+01	1.94e+01
19	2.68e+01	4.59e+01	3.34e+01	3.40e+01	4.56
20	4.03e+01	2.12e+02	1.12e+02	1.13e+02	3.51e+01
21	9.95e+02	1.03e+04	2.00e+03	2.60e+03	1.69e+03
22	2.63e+02	9.86e+02	6.60e+02	6.58e+02	1.64e+02
23	3.37e+02	3.37e+02	3.37e+02	3.37e+02	1.07e-07
24	2.56e+02	2.65e+02	2.64e+02	2.62e+02	3.37
25	2.00e+02	2.00e+02	2.00e+02	2.00e+02	4.49e-02
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02	1.04e-01
27	1.17e+03	1.48e+03	1.35e+03	1.35e+03	7.16e+01
28	1.00e+03	1.21e+03	1.12e+03	1.12e+03	4.67e+01
29	1.52e+03	2.55e+03	2.02e+03	2.00e+03	2.35e+02
30	1.41e+03	3.65e+03	2.00e+03	2.16e+03	5.24e+02

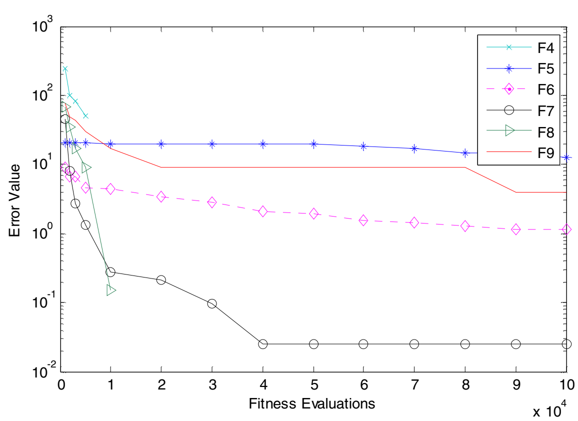

Fig. 4. *Convergence curves for F4-F9 of 10-D*

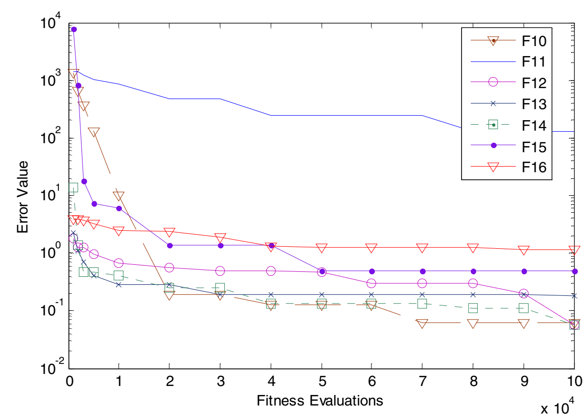

Fig. 5. *Convergence curves for F10-F16 of 10-D*

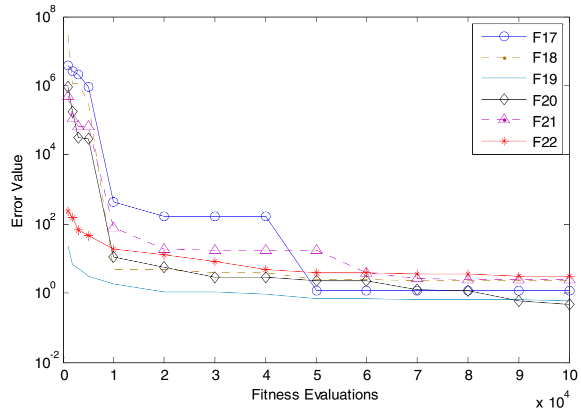

Fig. 6. *Convergence curves for F17-F22 of 10-D*

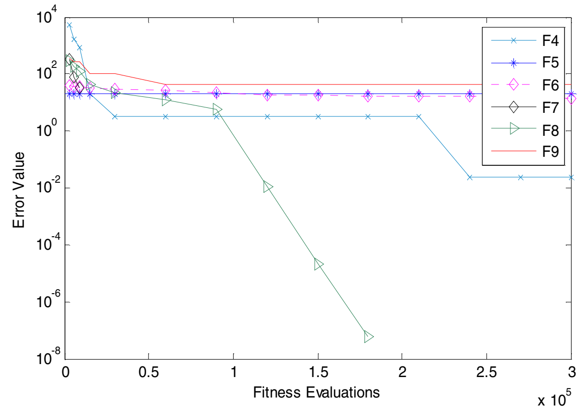

Fig. 9. *Convergence curves for F4-F9 of 30-D*

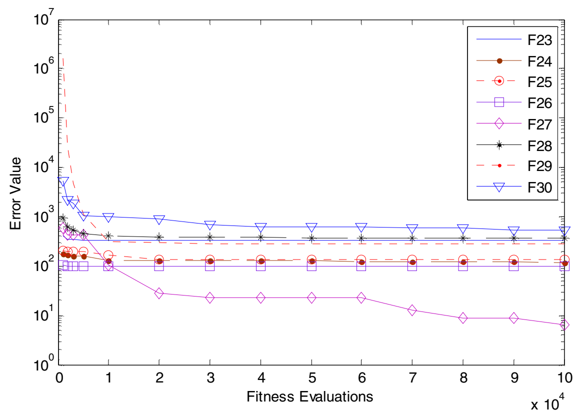

Fig. 7. *Convergence curves for F23-F30 of 10-D*

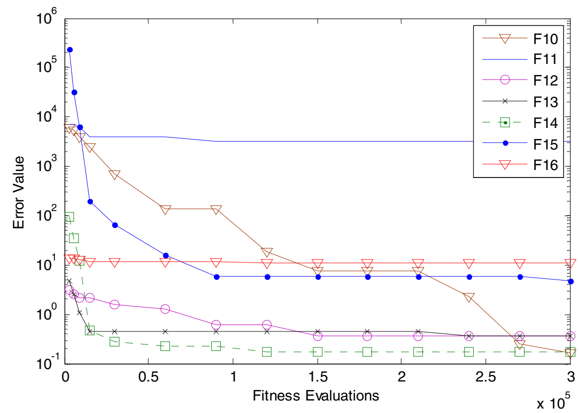

Fig. 10. *Convergence curves for F10-F16 of 30-D*

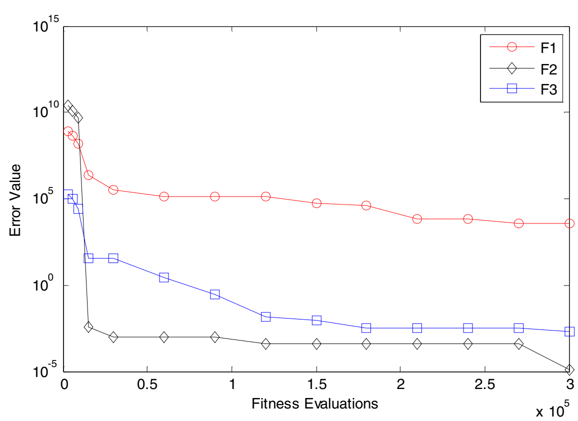

Fig. 8. *Convergence curves for F1-F3 of 30-D*

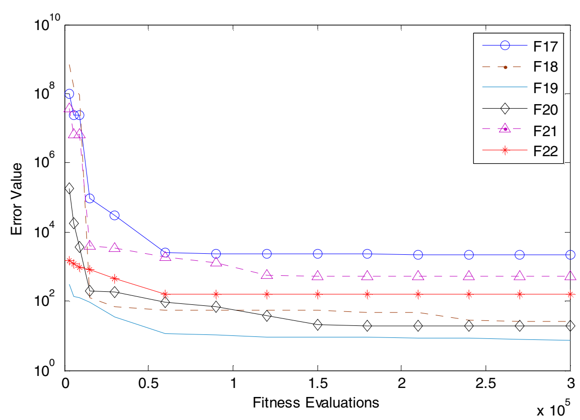

Fig. 11. *Convergence curves for F17-F22 of 30-D*

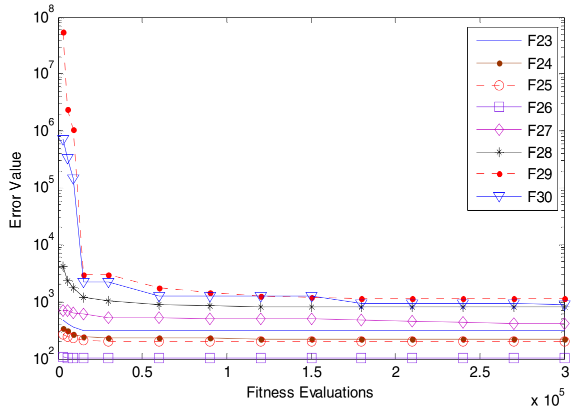

Fig. 12. *Convergence curves for F23-F30 of 30-D*

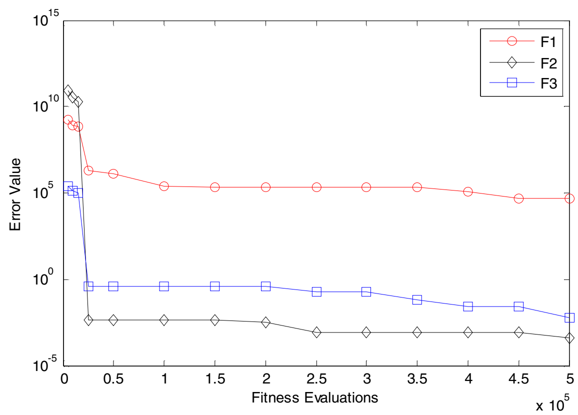

Fig. 13. *Convergence curves for F1-F3 of 50-D*

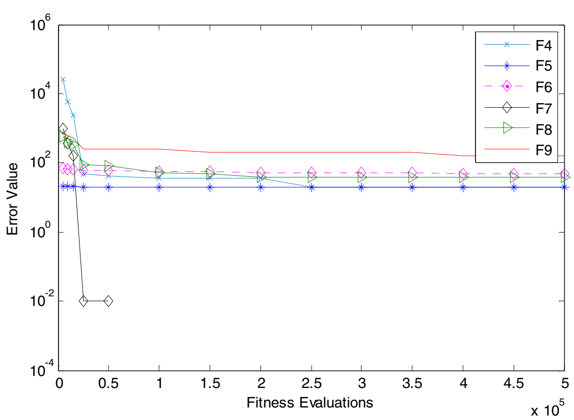

Fig. 14. *Convergence curves for F4-F9 of 50-D*

From the results, we can observe that for F4, Rosenbrock's function, which is a multimodal function but for which the global optimum can be found only when the algorithm has good local search ability because there is a narrow valley leading from the local optimum to the global optimum, MERDE presents a good performance for 10-*D*. While for 30-*D* and 50-*D*, the performance is not very stable. Sometimes the algorithm fails But for F1- F3, the unimodal functions, because of the diversity of the population, there is

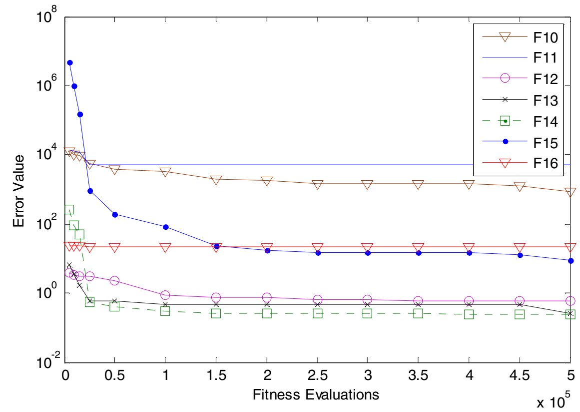

Fig. 15. *Convergence curves for F10-F16 of 50-D*

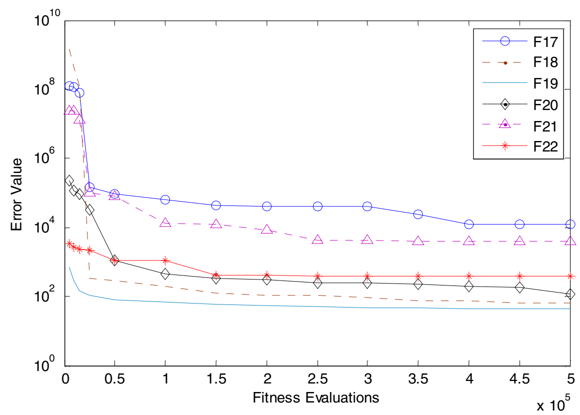

Fig. 16. *Convergence curves for F17-F22 of 50-D*

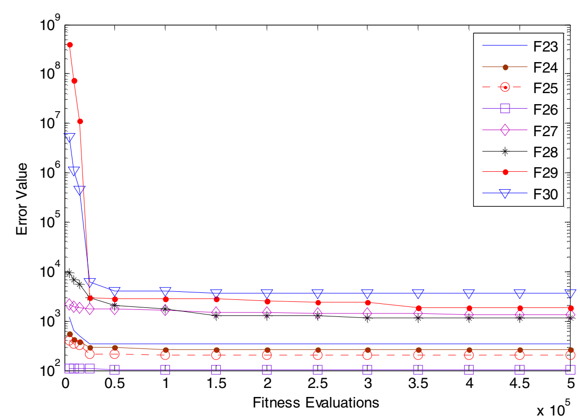

Fig. 17. *Convergence curves for F23-F30 of 50-D*

still improvement space for MFERDE. For F5, Ackley's function, MERDE's performance is not satisfactory because the basin of the global optimum is too small comparing the flat areas of the whole search space. The niching behavior of MERDE does not enhance the searching and slows down the convergence speed.

Comparing the results on F8 and F9, shifted Rastrigin's function and shifted and rotated Rastrigin's function, MERDE gives a better performance on un-rotated one. It is because we set a small *CR* value which makes the algorithm have a chance to locate the global optimum one dimension by one dimension. Comparing the results on F10 and F11, shifted Schwefel's function and shifted and rotated Schwefel's function, the same phenomenon is observed.

For the more complex functions such as hybrid functions and composition functions, it is observed the convergence speed is fast in the initial stage and the best solution obtained by MERDE stop improving at about 10% of the Max_FES for most functions of these two groups.

The algorithm's computational complexity is calculated as required in [7] on 10, 30 dimensions, to show the complexity's relationship with increasing dimensions. The results are presented in Table IV.

TABLE IV. COMPUTATIONAL COMPLEXITY

	T0	***T1***	$\hat{T}2$	**($\hat{T}2$ -*T1*)/*T0***
10-*D*	7.81e-02	1.81	406.12	5.18e+003
30-*D*		2.33	472.23	6.02e+003
50-*D*		3.30	635.01	8.09e+003

V. CONCLUSION

In this paper, a memetic FERDE was proposed to solve single objective global optimization problems. A modified fitness Euclidean-distance ratio was introduced based on the observation of the defects of the previous FER calculation. A local search method was combined with the algorithm to improve its local search ability. With the re-initialization strategy, the novel constructed algorithm, MFERDE, was tested on a set of benchmark functions proposed in CEC2014 and the results are analyzed. For the comparison of MFERDE with other algorithms, please refer to other articles in this competition.

REFERENCES

[1] C. Brizuela, "Applications of Multi-Objective Evolutionary Algorithms," *IEEE Computational Intelligence Magazine*, vol. 1(1), pp.43 - 44, 2006.

[2] A. Zhou, Yaochu Jin, and Qingfu Zhang, "A Population Prediction Strategy for Evolutionary Dynamic Multiobjective Optimization," *IEEE Transactions on Cybernetics*, vol.44(1), pp: 40-53, Jan. 2014

[3] Y. Wang and Z. Cai, "Combining Multiobjective Optimization With Differential Evolution to Solve Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 16(1), pp.117 - 134, 2012.

[4] J. J. Liang, Zhigang Shang, and Zhihui Li, "Coevolutionary Comprehensive Learning Particle Swarm Optimizer, " *IEEE Congress on Evolutionary Computation 2010*, pp. 1505-1512, July, 2010.

[5] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8(5), pp. 425 - 442, 2004.

[6] W. Chen, Jun Zhang, H.S.H. Chung, W. Zhong, W. Wu, and Y. Shi , "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol.14(2), pp. 278 - 300, 2010.

[7] J. J. Liang, B-Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization", *Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University*, Singapore, December 2013.

[8] R. Storn and K. V. Price, "Differential evolution- A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1995.

[9] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-art", *IEEE Trans. on Evolutionary Computation*, vol. 15(1), pp. 4-31, Feb. 2011..

[10] A. Basak, S. Das, K. C. Tan, "Multimodal Optimization Using a Biobjective Differential Evolution Algorithm Enhanced With Mean Distance-Based Selection," IEEE Trans. Evolutionary Computation, vol. 17(5), pp. 666-685, 2013.

[11] J. J. Liang, B. Y. Qu, X. B. Mao, B. Niu, and D.Y. Wang, " Differential Evolution Based on Fitness Euclidean-distance Ratio for Multimodal Optimization," accepted by *Neurocomputing*, 2014.

[12] X. Chen, Y. Ong, M. Lim, K. C. Tan, "A Multi-Facet Survey on Memetic Computation," *IEEE Trans. Evolutionary Computation*, vol. 15(5), pp.591-607, 2011

[13] J. Lin, Y. Chen,"Analysis on the Collaboration Between Global Search and Local Search in Memetic Computation," IEEE Trans. Evolutionary Computation, vol.15(5),pp. 608-623, 2011

[14] K. Price. "An introduction to differential evolution." *New Ideas in Optimization*, pages 79-108, 1999.

[15] X. Li, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ration," in *Proc. of Genetic and Evolutionary Computation Conference 2007 (GECCO 2007)*, 2007, pp.78-85.

[16] J. J. Liang & P. N. Suganthan, "Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constraint-Handling Mechanism," in *Proc. of IEEE Congress on Evolutionary Computation*, vol. 1-6, pp. 9-16, 2006.