JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 30, 749-763 (2014)

DOI:10.1688/JISE.2014.30.3.13

Adaptive Search Range and Multi-Mutation Strategies for Differential Evolution*

SHENG TA-HSIEH[1], SHIH-YUAN CHIU[2] AND SHI-JIM YEN[3]
[1]*Department of Communication Engineering*
Oriental Institute of Technology
New Taipei City, 220 Taiwan
[2]*Systems Development Center*
Chung-Shan Institute of Science and Technology
Taoyuan, 325 Taiwan
E-mail: fo013@mail.oit.edu.tw
[3]*Department of Computer Science and Information Engineering*
National Dong Hwa University
Hualien, 974 Taiwan

In this paper, an improved DE is proposed to improve optimization performance by involving four searching strategies: current-to-better mutation, real-random-mutation, sharing mutation, and focused search. When evolution speed is standstill, sharing mutation can increase the search depth; in addition, real-random mutation can disturb individuals and can help individuals diverge to local optimum, focused search can do large-scale searches around the best particle. When the evolution progresses well, current-to-better mutation will drive individuals to the correct evolution direction. Experiments were conducted on all of CEC 2005 test functions, include unimodal, multimodal and hybrid composition functions, to present performance of the proposed method and to compare with 5 variants of DE includes JADE, jDE, SaDE, DEGL and MDE_pBX. The proposed method exhibits better performance than other five related works in solving most the test functions.

Keywords: differential evolution, sharing mutation, optimization, real random mutation, focused search

1. INTRODUCTION

In last four decades, various heuristic-based algorithms were proposed for solving numerical optimization and real-world applications, such as genetic algorithm (GA) [1] and particle swarm optimizer (PSO) [2], *etc*. In 1995, the concept of original differential evolution (DE) was proposed by Storn and Price [3, 4]. It's a vector-based evolutionary algorithm with simple concept and high efficient. In recent years, more and more DE variants were proposed and have been applied for solving many real-world applications.

In 2006, Brest *et al.* proposed a self-adapting method for DE [5] to adjust control parameters Fi and Cri correspond to each individual. Each individual in the population is extended with parameter values. In 2009, Qin *et al.* proposed a self-adaptive DE named SaDE [6]. It combined two mutation strategies "DE/rand/1" and "DE/current-to-best/1" and self-adjusted control parameter according to their previous experiences to generate

Received February 28, 2013; accepted June 15, 2013.
Communicated by Hung-Yu Kao, Tzung-Pei Hong, Takahira Yamaguchi, Yau-Hwang Kuo, and Vincent Shin-Mu Tseng.
* This work was supported in part by National Science Council of Taiwan, Taiwan, under Grant NSC 101-2221-E-161-011.

potential solutions. Later, Das *et al.* proposed a neighborhood concept for population member of DE, called DEGL [7]. It's a similar idea from the community of the PSO algorithms. The small neighborhoods defined over the index-graph of parameter vectors. Later, Zhang and Sanderson proposed a new mutation strategy "DE/current-to-pbest" with optional external archive named JADE [8]. The optional archive operation adopted historical data to provide evolutionary information of direction. The adoptive parameters concept was also involved to JADE. In 2012, Islam *et al*. proposed an interesting DE named MDE_pBX [9], which adopted better group (q% of the population size) of randomly selected solutions from current generation to perturb the parent (target) vector.

In this paper, an improved DE is proposed for solving global numerical optimization. The proposed method is incorporated with focused search and three mutation schemes include current-to-better-mutation, real-random-mutation and sharing mutation for wide and deep searching in solution space. The proposed method can prevent the solutions from falling into the local minimum and enhance searching ability.

2. RELATED WORKS

2.1 Original DE

Differential evolution (DE) is arguably one of the most powerful stochastic real-parameter optimization algorithms in the mainstream. DE is a population-based optimization algorithm. The members of population in DE are called parameter vectors. There are four common mutation strategies of DE were developed [10, 11] and shown as follows.

1. DE/best/1

$$v_n^{child_i} = x_n^{best} + F(x_n^{r_1} - x_n^{r_2}) \tag{1}$$

2. DE/rand/2

$$v_n^{child_i} = x_n^{r_1} + F_1(x_n^{r_2} - x_n^{r_3}) + F_2(x_n^{r_4} - x_n^{r_5}) \tag{2}$$

3. DE/target/1

$$v_n^{child_i} = x_n^{i} + F(x_n^{r_1} - x_n^{r_2}) \tag{3}$$

4. DE/target to best/1

$$v_n^{child_i} = x_n^{i} + F(x_n^{best} - x_n^{r_1}) \tag{4}$$

where n and r_1, r_2, r_3, r_4, $r_5 \in (1, 2, \ldots, P)$ denote the iteration index and the index of randomly selected particles respectively, P denote the population sizes. The scaling factor $F \in (0, 1)$ adjusts the ratio of the particles' vector.

In DE algorithm, crossover will be is a performed after mutation. It will be active by crossover rate, it can prevent particle vector not to be convergent prematurely or too quickly, resulting in reducing capability for exploration. Regardless of change of the mutation mechanism, DE algorithm uses methods of uniform crossover. After mutation mechanism, DE algorithm randomly selected particles vector between current vector and donor vector for crossover and produce trial vector as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } (rand_{j,i}[0, 1] \le C_r \text{ or } j = j_{rand}) \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{5}$$

where $C_r \in [0, 1]$ is a predefined value for crossover rate. If the random value is less than C_r, the trial vector will inherit mutation. Otherwise, trial vector will adopt current vector directly. If the trial vector obtained by crossover-stage is better than current vector. It will be adopted for next iteration. Otherwise, the trail vector will be updated and replaced in next iteration. Once particles are updated, DE process will keep in iterations until the termination condition is reached. For minimization problems, the DE selection can be presented as follows.

$$X_{n+1}^i = \begin{cases} U_n^i & \text{if } f(U_n^i) \le f(X_n^i) \\ X_n^i & \text{otherwise} \end{cases} \tag{6}$$

2.2 Related Works of DE

In this section, five typical DE algorithms will be further described. These DE algorithms will then be taken into comparison with the proposed method in the experiments.

1. SaDE

In SaDE [6], four effective trial vector generation strategies namely the DE/rand/1/bin, DE/rand-to-best/2/bin, DE/rand/2/bin and finally DE/current-to-rand/1 are chosen to constitute a strategy candidate pool. In the SaDE algorithm, for each target vector in the current population, one trial vector generation strategy is selected from the candidate pool according to the probability learned from its success rate in generating individuals within a certain number of previous generations. The selected strategy is subsequently applied to the corresponding target vector to generate a trial vector. The performance of SaDE was compared with SDE [12], and jDE [5] over a suite of 26 bound constrained numerical optimization problems, and the authors reported that SaDE was more effective in obtaining better quality solutions with the relatively smaller standard deviations and higher success rates.

2. jDE

Brest *et al.* proposes a self-adaptation scheme for the DE control parameters, called Jde [5]. They encode control parameters F and CR with the individual and adjust them by introducing two new parameters $\tau 1$ and $\tau 2$. In their algorithm, a set of F and CR

values are assigned to each individual in the population, augmenting the dimensions of each vector. The better values of these encoded control parameters lead to better individuals that in turn, are more likely to survive and produce offspring and propagate these better parameter values. The new control parameters for the next generation are computed as follows:

$$F_{i,G+1} = \begin{cases} F_l + rand_1 * F_u & \text{if } rand_2 < \tau_1 \\ F_{i,G} & \text{otherwise} \end{cases} \tag{7}$$

$$CR_{i,G+1} = \begin{cases} rand_3 & \text{if } rand_4 < \tau_2 \\ CR_{i,G} & \text{otherwise} \end{cases} \tag{8}$$

where F_l and F_u are the lower and upper limits of F and both lie in [0, 1]. Brest *et al.* use $\tau 1 = \tau 2 = 0.1$. As $Fl = 0.1$ and $Fu = 0.9$, the new F takes a value from [0.1, 0.9] while the new CR takes a value from [0, 1]. As $F_{i,G+1}$ and $CR_{i,G+1}$ values are obtained before the mutation is performed, they influence the mutation, crossover, and selection operations for the new vector $\vec{X}_{i,G+1}$.

3. DEGL

Das *et al.* [7] propose two kinds of topological neighborhood models for DE in order to achieve better balance between its explorative and exploitative tendencies called DEGL to improvement over the DE/target-to-best/1 scheme. For example, there is a DE population $P_G = [\vec{X}_{1,G}, \vec{X}_{2,G}, \ ..., \vec{X}_{NP,G}]$ at generation G. The vector indices are sorted only randomly. Furthermore, for every vector $\vec{X}_{i,G}$, a neighborhood of radius k (where k is a nonzero integer from 0 to $(NP - 1)/2$) is defined. Then, the vectors of $\vec{X}_{i-k,G}, \ldots, \vec{X}_{i,G}, \ldots, \vec{X}_{i+k,G}$ are organized on a ring topology with their indices respectively. Such as vectors $\vec{X}_{NP,G}$ and $\vec{X}_{2,G}$ are the two immediate neighbors of vector $\vec{X}_{1,G}$. As a consequence, for each member of the population, a local donor vector is created by employing the best fittest vector in the neighborhood of that member and any two other vectors chosen from the same neighborhood. The model is defined as

$$\vec{L}_{i,G} = \vec{X}_{i,G} + \alpha * (\vec{X}_{n\_best_i,G} - \vec{X}_{i,G}) + \beta * (\vec{X}_{p,G} - \vec{X}_{q,G}) \tag{9}$$

where the subscript n_best_i indicates the best vector in the neighborhood of $\vec{X}_{i,G}$ and $p, q \in [i - k, i + k]$ with $p \neq q \neq i$. Similarly, the global donor vector is created as

$$\vec{g}_{i,G} = \vec{X}_{i,G} + \alpha * (\vec{X}_{g_{best},G} - \vec{X}_{i,G}) + \beta * (\vec{X}_{r1,G} - \vec{X}_{r2,G}) \tag{10}$$

where the subscript g_best indicates the best vector in the entire population at iteration G and $r1, r2 \in [1, NP]$ with $r1 \neq r \neq i$. α and β are the scaling factors. The local and global donor vectors are combined with a scalar weight $w \in (0, 1)$ to form the actual donor vector of the proposed algorithm

$$\vec{V}_{i,G} = \omega * \vec{g}_{i,G} + (1 + \omega) * \vec{L}_{i,G}. \tag{11}$$

Generally, neighborhood connections are independent of the positions of vectors pointed to. There is a delay in the information spread through the population regarding the best position of each neighborhood. Therefore, the attraction to specific points is weaker, which reduces the chance of getting trapped in local minimum.

4. JADE

The JADE [8] is proposed by Zhang and Sanderson. It can avoid attraction from the global best individual in population and improves convergence characteristics of DE. In JADE, a new mutation strategy is proposed which referred as DE/current-to-*p*best and added an optional external archive to track the previous successes and failures. In addition, the control parameters in an adaptive adjustment method within each generation are updated. The DE/current-to-*p*best strategy is a less greedy generalization of the DE/current-to-best/strategy. Instead of only adopting the best individual in the DE/current-to-best/1 strategy, the current-to-pbest/1 strategy utilizes the information of other better fitness solutions. Moreover, the recently explored inferior solutions are incorporated with this strategy. The DE/current-to-*p*best/1 with external archive generates the donor vector is defined as

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i * (\vec{X}^{p}_{best,G} - \vec{X}_{i,G}) + F_i * (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{12}$$

$\vec{X}^{p}_{best,G}$ is randomly chosen as one of the top 100p% individuals of the current population with $p \in (0, 1]$. The F_i is the scale factor associated with ith individual and updated dynamically in each generation. With the external archive A stored the recently explored inferior solution, $\vec{X}_{r_2^i,G}$ is randomly selected from $P \cup A$. The P is the current population. The external archive operation is made simply to avoid significant computation overhead. Initially it is empty. Then, after first generation, the parent solutions that fail in the selection process are added to it. If its size exceeds a certain threshold, then some solutions are randomly eliminated from it to keep the archive size fixed.

5. MDE_*p*BX

In 2012, Islam *et al.* propose a new mutation strategy, a fitness induced parent selection scheme for the binomial crossover of DE and a simple but effective scheme of adapting two of its most important control parameters, called MDE_*p*BX [9]. The new mutation operator named DE/current-to-gr_best/1 is a variant of the classical DE/current-to-best/1 scheme as follows.

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i(\vec{X}_{gr\_best,G} - \vec{X}_{i,G} + \vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{13}$$

where $\vec{X}_{gr\_best,G}$ is the best solution from 15% of individuals selected randomly in the current generation. Then, normal binomial crossover is performed between the donor vector and the randomly selected p-best vector to generate the trial vector at the same index. Parameter p is linearly reduced by generations as follows.

$$p = ceil\left[\frac{N_p}{2} * (1 - \frac{G-1}{G_{\max}})\right] \tag{14}$$

where G is the current generation number, $G_{\max}$ is the maximum generations, and $ceil(y)$ is the "ceiling" function returning the lowest integer greater than its argument y. Finally, the parameter adaptation schemes in MDE_pBX are inspired by JADE, the scale factor adaptation is independently generated as follows.

$$F_i = Cauchy(F_m, 0.1) \tag{15}$$

where $Cauchy(F_m, 0.1)$ is a random number sampled from a Cauchy distribution with location parameter F_m and scale parameter 0.1. Denote $F_{SUCCESS}$ as the set of the successful scale factors based on the last generation, $F_{SUCCESS}$ influences the current generation to generate better trial vectors that are likely to advance to the next generation.

Location parameter F_m of the Cauchy distribution is initialized to be 0.5 and updated at the end of each generation by following manner.

$$F_m = w_F * F_m + (1 - w_F) * mean_{POW}(F_{SUCCESS}) \tag{16}$$

The weight factor w_F varies randomly between 0.8 and 1 by following equation.

$$w_F = 0.8 + 0.2 * rand(0, 1) \tag{17}$$

and $mean_{POW}$ stands for the power mean is given by

$$mean_{POW}(F_{SUCCESS}) = \left| \sum_{x \in F_{SUCCESS}} \left(\frac{x^{1.5}}{|F_{SUCCESS}|} \right)^{\frac{1}{1.5}} \right|. \tag{18}$$

The crossover probability adaptation is similar with scale factor adaptation except the Gaussian instead of Cauchy.

3. THE PROPOSED METHOD

In this section, two new mutation methods are proposed to prevent solutions from falling into the local minimum. In addition, sharing mutation is also involved to increase particles' searching ability. Finally, the focused search is used to fine tune the searching direction around the global best particle.

3.1 DE/current-to-better/1

The original DE mutation schemes is DE/rand/1/bin, introduced by Storn and Price [3, 4], it is widely used scheme in the literature [13]; However, some papers [14, 15] indicate that DE/best/2 and DE/best/1 may have some advantages over DE/rand/1. Recently, Islam *et al.* [9] proposed a scheme, which called DE/current-to-gr_best/1; Although this new mutation method performs better results than many recent methods, the population may lose its diversity and global exploration abilities within a relatively small number of generations; furthermore it is getting trapped to some locally optimal point in the search space. In this paper, the new scheme, which we call DE/current-to-better/1, can be expressed as

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i(\vec{X}_{better,G} - \vec{X}_{i,G} + \vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}). \tag{19}$$

In Eq. (19), $\vec{X}_{i,G}$ is known as the target vector, $\vec{V}_{i,G}$ is known as the donor vector, the scaling factor F is a positive control parameter for scaling the difference vectors. The $\vec{X}_{r_1^i,G}$ and $\vec{X}_{r_2^i,G}$ are two distinct vectors picked up randomly from the current population, and none of them is equal to $\vec{X}_{better,G}$ or the target vector. $\vec{X}_{better,G}$ is chosen from top q% individual of current population. This method is similar with the mutation method in MDE_pBX, but the $\vec{X}_{better,G}$ for every individuals will be differed, it drives the popupulation to the better direction instead of convergence to the best individual, so this mutation scheme force DE not to fall into local optimum quickly.

3.2 DE/real-random/1

When the population falls into the local optimum, that is, most individuals in the same dimension are very similar. This causes the second term of right side in function (19) to be close to zero, which will lose the capabilities of search. In the scheme of Real Random Mutation, mutation function is defined as follows:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i(\vec{X}_n - \vec{X}_{i,G}). \tag{20}$$

In (20), $\vec{X}_n$ is a new born particle and not belongs to current population. The contents of this particle are random generated between upper and lower bound of search range. The difference between $\vec{X}_n$ and $\vec{X}_{i,G}$ will be diverged significantly. Thus, the new born particle can provide useful information to help clustered particles escape local optimum. Fig. 1 shows the particle's movement, driven by Eq. (20) in a two-dimensional search space.

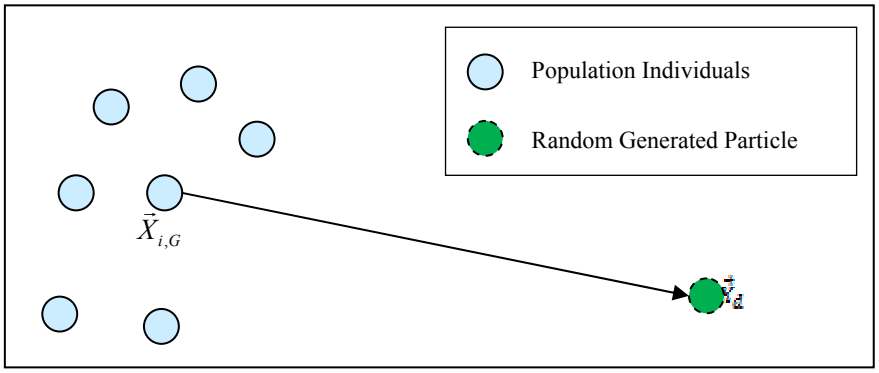

Fig. 1. Particles' movement by real random mutation.

3.3 Sharing Mutation

In order to increase efficiency while exploring the solution space, the *Sharing Mutation* (SM) [16] is adopted. For the SM, one of the dimensions will be picked

randomly; the mutating individual corresponding to this dimension is perturbed and restricted as this dimension's solution for all chromosomes. For example, a randomly selected dimension (d) of the individual i gets perturbed in the range between [$S_{d\min}$, $S_{d\max}$], where $S_{d\min}$ and $S_{d\max}$ are the minimal and maximal solution of d of all individuals respectively. In other words, it is the sharing of searching ranges of selected dimensions among chromosomes to efficiently generate new solutions.

3.4 Focused Search

According to DE's search behavior, it can be found that DE can perform well performance on widely search for exploring unsearched solution space but weakly on perform deeply search. In order to make particles more efficient to approach potential global best solution, in this paper, the focused search strategy is proposed to gradual reduce the scale of searching vector of the global best particle.

```
The initial moving vector v_m^d = (Initial_range_max -Initial_range_min)/2
For each dimension d of the global best particle x
    If (fitness(x_d + v_m^d) < fitness(x_d))
        x_d = x_i + v_m^d;
    elseif (fitness(x_d + (–v_m^d)) < fitness(x_d))
        x_i = x_i – v_m^d;
    else
        v_m^d = –v_m^d / 2;
    Endif
Endfor
```

Fig. 2. Pseudo code of focused search.

After mutation and crossover, the best particle will then be selected. Each dimension's moving vector is according to of current previous moving vector. In first stage, the moving is according to initial search range. For example, if the initial search range is [−60, 100], the moving vector will be (100 – (−60))/2 = 80. Thus, the moving vector will be added to one of dimensions of the best particle. If the particle's new position is better than original position, thus, the particle's new position will be kept and this moving vector will be applied in next iteration. In other words, if the particle's new position is worth than original position or over the searching boundary, the applied moving vector will be change to negative, *i.e.* −80. Similar to previous step, if the particle's new position is better than original position, thus, the particle's new position will be kept. The moving vector will be applied in next iteration. In other words, if the particle's new position is worth than original position or over the searching boundary, the applied moving vector will be reduce to as half previous moving vector, *i.e.* −40. If this moving vector can help particle find better solution, it will be kept. On the contrary, moving vector will be change to negative, *i.e.* 40. The rest steps of moving vector adjustment are the same previous. Thus, the moving vector will gradual reduce according to current search state. The pseudo code of focused search strategy is shown in Fig. 2. The focused

search will help particle to search the area getting closer to itself once a dimension. It seems waste much time on single dimension search but it will fine tune the searching direction for finding global best solution.

3.5 Flowchart of Proposed Method

The complete flowchart of the proposed method is given in Fig. 3. After initializing, the DE/current-to-better is the first performed mutation for all particles of the population. Then, the crossover is to combine particles' information. If there is no better solution can be found after continuous G generations, the T% elements of donor vectors will perform DE/real-random/1 mutation and sharing mutation. The rest particles will also perform DE/current-to-better mutation.

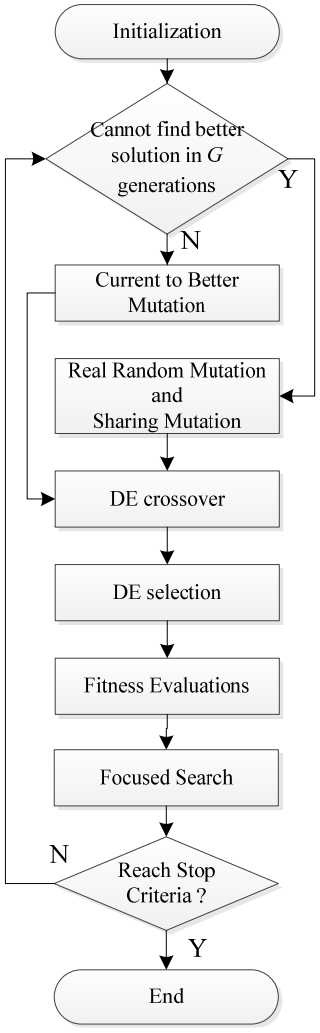

Fig. 3. Flowchart of the proposed method.

Once any better solution is found, T value will be set as zero. All the particles will be moved by DE/current-to-better. After selection and fitness evaluation, the global best particle will be pickup to perform focused search. The previous steps will repeat multiple iterations until the stop criteria are reached.

4. EXPERIMENT RESULTS

4.1 Test Functions

In order to test the proposed method and to compare it with DE related works, CEC 2005 benchmarks [17] were selected, which include five unimodal, seven basic multi-modal, two expanded multimodal and eleven hybrid composition multimodal functions. The global optimum (equal to function bias), search range, initialization range and function types of each test function are presented in Table 1.

Table 1. Related parameter settings of the CEC 2005 test functions.

Functions	Global Optimum	Initial Range	Search Range	Function Types
F_1	–450	$[-100, 100]^D$	$[-100, 100]^D$	Unimodal
F_2	–450	$[-100, 100]^D$	$[-100, 100]^D$	Unimodal
F_3	–450	$[-100, 100]^D$	$[-100, 100]^D$	Unimodal
F_4	–450	$[-100, 100]^D$	$[-100, 100]^D$	Unimodal
F_5	–310	$[-100, 100]^D$	$[-100, 100]^D$	Unimodal
F_6	–390	$[-100, 100]^D$	$[-100, 100]^D$	Basic multimodal
F_7	–180	$[0, 600]^D$	No Boundary	Basic multimodal
F_8	–140	$[-32, 32]^D$	[–32, 32]D	Basic multimodal
F_9	–330	$[-5, 5]^D$	$[-5, 5]^D$	Basic multimodal
F_{10}	–330	$[-5, 5]^D$	$[-5, 5]^D$	Basic multimodal
F_{11}	90	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	Basic multimodal
F_{12}	–460	$[-\pi, \pi]^D$	$[-\pi, \pi]^D$	Basic multimodal
F_{13}	–130	$[-5, 5]^D$	$[-5, 5]^D$	Expanded multimodal
F_{14}	–300	$[-100, 100]^D$	$[-100, 100]^D$	Expanded multimodal
F_{15}	120	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{16}	120	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{17}	120	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{18}	10	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{19}	10	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{20}	10	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{21}	360	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{22}	360	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{23}	360	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{24}	260	$[-5, 5]^D$	$[-5, 5]^D$	Hybrid Composition
F_{25}	260	$[2, 5]^D$	No Boundary	Hybrid Composition

4.2 Parameters Settings

In the experiments, twenty five test functions with 50 dimensions are conducted for

comparing the proposed method with five related works jDE [5], SaDE [6], DEGL [7], JADE [8] and MDE_*p*BX [9]. The parameters of these methods are according to their original settings. Except the initial population size of proposed method is set as 100 and the scale factor (F) and crossover probability (C_r) adaptation of proposed method are referring to MDE_*p*BX [9]. Each algorithm is executed for 25 times independently. The maximum fitness evaluation (FEs) is set as 500,000. G, T and q are set to 3, 5 and 15 in proposed method. The mean values and standard deviation of are calculated. If the value is less than 10^{-12}, the value is deemed to 0.

4.3 Experimental Results

The experiments results, which include the mean and standard deviation, 25 runs of the six optimizers on the 25 test functions with 50 dimensions, are listed in Table 2. The best results among the six approaches are shown in bold.

Table 2. Experiment results of the CEC 2005 test functions.

Methods	**Results**	F_1	F_2	F_3	F_4	F_5
JADE	Mean	**0.0000E+00**	5.6310E-04	8.7156E+04	3.1600E+03	3.0550E+03
	Std.	**0.0000E+00**	7.8233E-06	3.6847E+04	4.1340E-01	5.4850E+02
jDE	Mean	3.1544E-09	5.2020E+03	2.9770E+07	1.0194E+04	4.2060E+03
	Std.	4.9946E-09	1.4860E+03	5.7440E+06	2.1828E+00	5.0880E+02
SaDE	Mean	1.4872E-11	2.2800E-03	7.1790E+05	9.7780E+04	5.9920E+03
	Std.	2.8335E-12	8.5450E-03	1.0070E+06	9.8350E+01	4.4640E+02
DEGL	Mean	6.4679E-10	1.2960E-05	2.3110E+05	2.8851E+04	6.0930E+03
	Std.	9.8640E-11	9.5612E-06	1.0320E+05	1.8932E+01	6.8400E+02
MDE_*p*BX	Mean	**0.0000E+00**	4.4563E-06	**3.5438E+04**	**1.2075E+02**	2.0758E+03
	Std.	**0.0000E+00**	8.7963E-11	**1.7823E+04**	**2.3453E-06**	1.9111E+02
Proposed	Mean	**0.0000E+00**	**4.4862E-10**	1.8787E+05	8.7260E+03	**1.9887E+03**
Method	Std.	**0.0000E+00**	**6.9845E-10**	1.2100E+05	4.4927E+03	**4.6488E+02**
Methods	**Results**	F_6	F_7	F_8	F_9	F_{10}
JADE	Mean	1.5413E+01	6.1932E+03	2.1136E+01	1.3520E+02	1.9350E+02
	Std.	1.0642E+01	1.8400E+00	3.2510E-02	2.5910E+00	2.0600E+01
jDE	Mean	4.1758E+01	6.3114E+03	2.1132E+01	1.7160E+02	1.9597E+02
	Std.	8.9100E+00	1.5960E+01	3.8070E-02	1.4090E+01	5.6236E+01
SaDE	Mean	1.1337E+01	6.1951E+03	2.1132E+01	1.1480E+02	6.3420E+01
	Std.	1.0440E+01	4.5940E-12	3.4580E-02	1.2660E+01	1.2870E+01
DEGL	Mean	1.3452E+01	6.1953E+03	2.1131E+01	1.6200E+02	1.0217E+02
	Std.	1.1080E+01	4.5940E-12	3.9170E-02	1.7430E+01	3.5590E+01
MDE_*p*BX	Mean	**7.9745E-01**	6.1835E+03	2.0422E+01	1.0642E+02	**3.6818E+01**
	Std.	**1.0112E+00**	2.0912E+00	3.1241E-02	1.2346E+01	**1.3496E+01**
Proposed	Mean	3.4166E+00	**5.6102E-03**	**2.0173E+01**	**2.1838E+01**	7.8880E+01
Metho**d**	Std.	5.0378E+00	**1.0000E-02**	**2.7679E-01**	**8.2991E+00**	1.7897E+01
Methods	**Results**	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
JADE	Mean	6.2080E+01	1.7680E+05	2.3112E+01	2.2840E+01	3.7690E+02
	Std.	1.7440E+00	7.1050E+04	4.7840E-01	2.5486E-01	8.7640E+01
jDE	Mean	7.3300E+01	1.4730E+05	2.5603E+01	2.3090E+01	4.0000E+02
	Std.	1.0080E+00	1.9280E+05	1.3220E+00	2.8437E-01	0.0000E+00
SaDE	Mean	6.6340E+01	8.7810E+03	2.7710E+01	2.2840E+01	**3.8827E+01**
	Std.	1.4850E+00	7.0920E+03	4.1120E+00	2.0634E-01	**1.0755E+02**

Table 2. (Cont'd) Experiment results of the CEC 2005 test functions.

DEGL	Mean	6.2900E+01	5.7810E+04	3.0630E+01	2.2620E+01	3.8982E+02
	Std.	1.3600E+01	4.5660E+04	4.3610E+00	3.3750E-01	4.9284E+01
MDE_*p*BX	Mean	4.1328E+01	1.0779E+04	2.0628E+01	2.1720E+01	3.6670E+02
	Std.	1.5450E+00	8.4902E+03	1.2654E+00	2.0801E-01	6.1264E+01
Proposed Method	Mean	**3.0107E+01**	**7.6800E+03**	**4.2014E+00**	**2.1715E+01**	2.7579E+02
	Std.	**5.7684E+00**	**8.9040E+03**	**9.2838E-01**	**7.9225E-01**	9.1441E+01
Methods	**Results**	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
JADE	Mean	1.4370E+02	1.8960E+02	9.2060E+02	9.6031E+02	9.8672E+02
	Std.	5.2267E+01	3.8745E+01	1.8930E+00	2.5236E+01	1.8675E+02
jDE	Mean	2.7160E+02	3.0590E+02	9.1450E+02	9.2090E+02	9.9121E+02
	Std.	4.7190E+00	1.1630E+01	3.1630E+01	1.0406E+01	1.5365E+01
SaDE	Mean	**1.5420E+01**	1.9340E+02	9.0410E+02	9.3493E+02	9.3167E+02
	Std.	**6.1686E+01**	2.9679E+00	5.2080E+01	1.9639E+01	2.0137E+01
DEGL	Mean	1.3153E+02	1.7659E+02	9.6067E+02	9.1430E+02	9.2196E+02
	Std.	1.9986E+01	2.3653E+01	2.8458E+01	2.0105E+01	4.5874E+01
MDE_*p*BX	Mean	1.1142E+02	1.2502E+02	**8.2640E+02**	9.0496E+02	9.0280E+02
	Std.	3.5706E+01	1.8344E+01	**8.5217E+01**	2.8204E+01	2.5078E+01
Proposed Method	Mean	7.0344E+01	**1.0729E+02**	8.5957E+02	**8.3812E+02**	**8.4738E+02**
	Std.	6.9722E+01	**1.0058E+02**	4.3799E+01	**1.0016E+00**	**3.0807E+01**
Methods	**Results**	F_{21}	F_{22}	F_{23}	F_{24}	F_{25}
JADE	Mean	8.5230E+02	9.1370E+02	8.1030E+02	**2.0000E+02**	1.6632E+03
	Std.	3.5175E+02	2.4356E+01	2.4572E+02	**0.0000E+00**	5.5842E+00
jDE	Mean	8.0619E+02	9.7960E+02	8.3044E+02	**2.0000E+02**	1.7280E+03
	Std.	1.0896E+02	1.4851E+01	1.0787E+02	**0.0000E+00**	6.2562E+00
SaDE	Mean	8.6400E+02	9.7245E+02	8.6405E+02	**2.0000E+02**	1.7586E+03
	Std.	1.5779E+02	3.3383E+01	1.5266E+02	**0.0000E+00**	3.1453E+00
DEGL	Mean	8.3600E+02	9.4242E+02	8.3934E+02	7.2465E+02	1.6710E+03
	Std.	2.1772E+02	3.5647E+01	1.6620E+02	8.3066E+01	6.5096E+00
MDE_*p*BX	Mean	**5.0000E+02**	8.9870E+02	**5.0000E+00**	**2.0000E+02**	9.6149E+02
	Std.	**0.0000E+00**	9.0750E+00	**0.0000E+00**	**0.0000E+00**	6.0040E+00
Proposed Method	Mean	6.0916E+02	**5.0008E+02**	6.3911E+02	3.6558E+02	**4.8622E+02**
	Std.	1.1598E+02	**8.1383E-02**	9.8057E+01	3.4886E+02	**4.2724E+02**

From the results, it can be observed that the proposed method performs better results than related works. JADE, MDE_*p*BX and proposed method archive the same results which real optimum solutions are found in function 1. To sum up, in 25 test functions, the proposed method leads in 15 functions, and JADE, jDE, SaDE, DEGL and MDE_*p*BX lead in 2, 1, 3, 0 and 9 functions respectively.

5. CONCLUSIONS

In this paper, an improved differential evolution (DE) is proposed for solving global numerical optimization problems. The proposed method performs better results on most test functions. The proposed method contains three major parts which are differentiates from the original DE. The current-to-better mutation will pick up better particles for guiding other particles toward to potential solution space. Then, the sharing mutation will find tune to evolution direction for depth search. Also, the real-random mutation can

prevent particles from falling into the local optimum. Finally, the focused search can fine tune the searching direction around the global best particle.

In experiments, twenty five test functions of CEC 2005 benchmark with 50 dimensions were selected for testing performance of proposed method and related works. From the results, it can be observed that the proposed method exhibit better performance on unimodal, multimodal and hybrid composition functions than other related works.

REFERENCES

1. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
2. J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceeding of the 4th IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
3. R. Storn and K. V. Price, "Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces," Technical Report TR-95-012, ICSI, Berkeley, CA.
4. R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol. 11, 1997, pp. 341-359.
5. J. Brest, S. Greiner, B. Boškovi´c, M. Mernik, and V. Žumer, "Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, Vol. 10, 2006, pp. 646-657.
6. A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 13, 2009, pp. 398-417.
7. S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood based mutation operator," *IEEE Transactions on Evolutionary Computation*, Vol. 13, no. 3, 2009, pp. 526-553.
8. J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, Vol. 13, 2009, pp. 945-958.
9. S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on System*, *Man*, *Cybernetics – Part B: Cybernetics*, Vol. 42, 2012, pp. 482-500.
10. K. Price, R. Storn, and J. Lampinen, *Differential Evolution – A Practical Approach to Global Optimization*, Berlin, Germany, Springer, 2005.
11. K. V. Price, "An introduction to differential evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and V. Glover, eds., McGraw-Hill, London, U.K., 1999, pp. 79-108.
12. M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Self-adaptive differential evolution," in *Proceedings of International Conference on Computational Intelligence and Security*, LNAI 3801. 2005, pp. 192-199.
13. S. Das and P. N. Suganthan, "Differential evolution – A survey of the state-of-the-

art," *IEEE Transactions on Evolutionary Computation*, Vol. 15, 2011, pp. 4-31.
14. E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, 2006, pp. 485-492.
15. R. Gamperle, S. D. Muller, and A. Koumoutsakos, "Parameter study for differential evolution," in *Proceedings of WSEAS International conference on Advances in Intelligent Systems*, *Fuzzy Systems*, *Evolutionary Computation*, 2002, pp. 293-298.
16. S.-T Hsieh, T.-Y Sun, C.-C. Liu, and S.-J. Tsai, "Potential offspring production Strategies: An improved genetic algorithm for global numerical optimization," *Expert Systems with Applications*, Vol. 36, 2009, pp. 11088-11098.
17. http://www3.ntu.edu.sg/home/epnsugan/.

Sheng-Ta Hsieh (謝昇達) received the B.S. degree in Electrical Engineering from the National Taiwan University of Science and Technology in 2002 and received Ph.D. degrees in Electrical Engineering from National Dong Hwa University, Hualien, Taiwan, in 2007. He is currently an Associate Professor in the Department of Communication Engineering, Oriental Institute of Technology, New Taipei City, Taiwan. His interests in research include embedded systems, evolutionary computation, and multi-objective optimization.

Shih-Yuan Chiu (丘世元) received Ph.D. degrees in the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, in 2013. His research interests include computer games, puzzle game solver and computational intelligence. He is also a 4-dan Go player.

Shi-Jim Yen (顏士淨) is a Professor of Department of Computer Science and Information Engineering in National Dong Hwa University. He received a Ph.D degree in Computer Science and Information Engineering from National Taiwan University, Taiwan. He is an IEEE CIS member. He specialized in Artificial Intelligence and computer games. In these areas, he has published over 50 papers in international journals or conference proceedings. He is a 6-dan Go player. He served as a workshop chair on 5th international conference on Grid and Pervasive Computing in 2010, and a workshop chair of 2010 International Taiwanese

Association for Artificial Intelligence (TAAI) conference. He serves as a workshop co-chair of 2011 IEEE International Conference on Fuzzy Systems. He is the Chair of the IEEE Computational Intelligence Society. (CIS) Emergent Technologies Technical Committee (ETTC) Task Force on Emerging Technologies for Computer Go in 2010.