

## IMPROVING THE SEARCH PERFORMANCE OF RANK-BASED DIFFERENTIAL EVOLUTION WITH MULTIVECTOR MUTATION

JUN-ICHI KUSHIDA, AKIRA HARA AND TETSUYUKI TAKAHAMA

Graduate School of Information Sciences  
Hiroshima City University  
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima-shi 731-3194, Japan  
{kushida; ahara; takahama}@hiroshima-cu.ac.jp

Received June 2017; revised October 2017

**ABSTRACT.** *Differential evolution (DE) is a simple and efficient algorithm for solving optimization problems in a continuous space. Although DE shows relatively good performance, its search performance depends greatly on the mutation strategy. In this research, we focus on the number of difference vectors in strategy of differential mutation. The characteristics of the differential mutation using multiple difference vectors differ from mutation of single vector. In this paper, we firstly analyze the search characteristics of DE with multivector mutation. Next, we improve an algorithm of rank-based differential evolution (RDE) by introducing multivector mutation. RDE is one of the DE variants where control parameters are adaptively controlled during the search. The proposed method, called RDE-MM, allocates different control parameter values and the number of difference vectors for each individual based on the ranking information in the current population. Through the experiments using standard benchmark functions, we show that the multivector mutation can enhance the search ability of RDE.*

**Keywords:** Differential evolution, Multivector mutation, Population diversity

1. **Introduction.** Differential evolution (DE) [1, 2] is a population-based stochastic search method for solving global optimization problems in continuous domain. It has been successfully applied to various standard benchmark problems and has found several real-world applications. However, the performance of DE mainly depends on mutation strategies and crossover operators and their associated control parameters (i.e., population size  $NP$ , scaling factor  $F$ , and crossover rate  $CR$ ). There are some variants of DE that have been proposed. The variants are classified based on the notation DE/*base*/*num*/*cross*, where “*base*” specifies a way of selecting an individual that will form the base vector, “*num*” specifies the number of difference vectors used to perturb the base vector, and “*cross*” specifies the type of crossover.

In DE, the most popular strategy is DE/rand/1 where only one difference vector is generated and the base vector is chosen by random from population. Meanwhile, it is also possible to have multiple difference vectors in mutation. Opara and Arabas, in [3], showed that the difference vector distribution for the differential mutation using multiple difference vectors tends to a normal distribution, whose covariance matrix is proportional to the covariance matrix of the current population. Furthermore, a generalization of DE/rand/ $k$  mutation schemes for large  $k$  is defined in [4].

Rank-based differential evolution (RDE) [5], which is one of the adaptive parameter controlling methods, employs RDE/rand/1 strategy. In RDE, different  $F$  and  $CR$  are allocated for each individual by using ranking information in the current population. Due to its parameter allocation scheme, RDE has a well balance between exploration and exploitation which is necessary for solving black box optimization. Additionally, we have

proposed several variants of RDE to improve the search ability. For large scale global optimization, we introduced a selection mechanism of the mutation strategy to RDE with landscape modality detection [6]. Also, we incorporated the eigenvector-based (EIG) crossover operator to RDE and proposed a control scheme to select EIG or exponential crossover [7]. These methods outperformed the original RDE on a set of benchmark functions; however, one difference vector was used in their mutation operators.

In this paper, we focus on the ability of multivector mutation which can maintain population diversity, and aim to improve the performance of RDE by incorporating multivector mutation. At first, the search characteristics of DE with multivector mutation are analyzed on standard benchmark functions. Next, we propose RDE with multivector mutation (RDE-MM) where  $F$ ,  $CR$  and the number of difference vectors are controlled at each individual based on the ranking information in the current population. This paper is organized as follows. A brief description of the DE algorithm is given in Section 2. Section 3 describes impact of multivector mutation in DE. In Section 4, we present the proposed RDE with multivector mutation. In Section 5 we present the result of experiments. Finally, the conclusion is given in Section 6.

## 2. Optimization by Differential Evolution.

**2.1. Optimization problems.** In this study, the following optimization problem with lower bound and upper bound constraints will be discussed.

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && L_j \leq x_j \leq U_j, \quad j = 1, \dots, D \end{aligned} \quad (1)$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_D)$  is a  $D$ -dimensional vector and  $f(\mathbf{x})$  is an objective function. Values  $L_j$  and  $U_j$  are the lower bound and the upper bound of  $x_j$ , respectively. Let the search space in which every point satisfies the lower and upper bound constraints be denoted by  $S$ .

**2.2. Differential evolution.** DE is one of the variants of evolutionary algorithms that use a population. There are some variants of DE that have been proposed. The variants are denoted as DE/*base*/*num*/*cross*, where “*base*” denotes the manner of constructing the mutant vector, “*num*” denotes the number of difference vectors, and “*cross*” indicates crossover method.

The pseudo-code of DE/*rand*/*1*/*-* is presented in Algorithm 1, where  $G_{\max}$  denotes the maximum number of generations. In the initialization phase,  $NP$  individuals  $P = \{\mathbf{x}_i, i = 1, 2, \dots, NP\}$  are randomly generated in a given search space. Each individual contains  $D$  genes as decision variables. At each generation, DE creates a mutant vector  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  for each individual  $\mathbf{x}_i$  (called a target vector) in the current population. Some well-known mutation operations are listed as follows.

“**rand/1**”:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (2)$$

“**best/1**”:

$$\mathbf{v}_i = \mathbf{x}_{best} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (3)$$

“**current-to-best/1**”:

$$\mathbf{v}_i = \mathbf{x}_i + F(\mathbf{x}_{best} - \mathbf{x}_i) + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (4)$$

In the above equations,  $\mathbf{x}_{best}$  is the best individual in the current population, and the indices  $r_1$ ,  $r_2$  and  $r_3$  are distinct integers uniformly chosen from the set  $\{1, 2, \dots, NP\} \setminus \{i\}$ . The parameter  $F$  is called the scaling factor, which amplifies the difference vectors.

**Algorithm 1** DE/rand/1/-

---

```

1: /*Initialize a population*/
2:  $P = NP$  individuals  $\{\mathbf{x}_i\}$  generated randomly in  $S$ ;
3: Set scaling factor  $F$  and crossover rate  $CR$ ;
4: for  $G = 1$  to  $G_{\max}$  do
5:   for  $i = 1$  to  $NP$  do
6:     /*DE operation*/
7:      $(\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}) =$  randomly selected from  $P$ 
8:     s.t.,  $r_1 \neq r_2 \neq r_3 \neq i$ ;
9:      $\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$ ;
10:     $\mathbf{u}_i =$  trial vector generated from  $\mathbf{x}_i$  and  $\mathbf{v}_i$  by a crossover;
11:   end for
12:   for  $i = 1$  to  $NP$  do
13:     if  $f(\mathbf{u}_i) \leq f(\mathbf{x}_i)$  then
14:        $\mathbf{x}_i^{new} = \mathbf{u}_i$ ;
15:     else
16:        $\mathbf{x}_i^{new} = \mathbf{x}_i$ ;
17:     end if
18:   end for
19:    $P = \{\mathbf{x}_i^{new}, i = 1, 2, \dots, NP\}$ ;
20: end for

```

---

After mutation, DE performs the crossover operator between target vector and mutant vector, and generates a trial vector  $\mathbf{u}_i = (u_{i1}, u_{i2}, \dots, u_{iD})$ . In DE, there are two main crossovers: exponential (exp) crossover and binomial (bin) crossover. In this paper, we use shuffled exponential crossover, which does not rely on arbitrary dependencies between adjacent variables [8]. In the crossover,  $CR$  is the crossover rate within the range  $[0, 1)$  and presents the probability of generating genes for a trial vector  $\mathbf{u}_i$  from a mutant vector  $\mathbf{v}_i$ . If the  $j$ th element  $u_{ij}$  of the trial vector  $\mathbf{u}_i$  is infeasible (i.e., out of the boundary  $[L_j, U_j]$ ), it is reset as follows:

$$u_{ij} = \begin{cases} (L_j + x_{ij})/2 & (u_j < L_j) \\ (U_j + x_{ij})/2 & (u_j > U_j) \end{cases}$$

After all of the trial vectors have been generated, the selection operator is performed to select a better one from the target vector  $\mathbf{x}_i$  and its corresponding trial vector  $\mathbf{u}_i$  according to their fitness values  $f(\cdot)$ . The selected vector is given by

$$\mathbf{x}_i^{new} = \begin{cases} \mathbf{u}_i & \text{if } f(\mathbf{u}_i) \leq f(\mathbf{x}_i) \\ \mathbf{x}_i & \text{otherwise} \end{cases} \quad (5)$$

and  $\mathbf{x}_i^{new}$  is used as a target vector in the next generation.

**2.3. Multivector mutation.** In a classical DE, a mechanism of using one difference vector is the most common choices. When a pair of individuals is used for difference mutation, the distribution of the differential vector becomes a triangular distribution centered on the base vector. Also, the distribution of the differential vectors generated in a certain generation approaches the normal distribution centered on the center of gravity of the population. Distribution of scaled difference vector depends on the population contents and can be expressed by means of convolution of distributions [3]. This distribution is symmetric with respect to origin, and has zero mean and its covariance matrix is proportional to the covariance matrix of vectors in the current population. In DE/rand/1, if a

distribution of target vectors  $\mathbf{x}_i$  has mean  $\mu_x$  and covariance  $\mathbf{C}_x$ , a distribution of mutant vector has mean  $\mu_y = \mu_x$  and covariance  $\mathbf{C}_y = (1 + 2F^2)\mathbf{C}_x$ .

On the other hand, it is also possible to use multiple difference vectors for mutation. Multivector mutation is generalized by using  $k$  difference vectors, which is denoted by DE/rand/ $k$ .

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}) + \cdots + F(\mathbf{x}_{r_{2k}} - \mathbf{x}_{r_{2k+1}}) \quad (6)$$

Indices  $i, r_1, r_2, \dots, r_{2k+1}$  are distinct integers. The covariance matrix of mutant vectors for DE/rand/ $k$  equals  $(1 + 2kF^2)\mathbf{C}_x$ . Therefore, as the number of difference vectors increases, the distribution of mutation vectors becomes wider. This effect can be eliminated by dividing the sum of scaled difference vectors by  $\sqrt{k}$ :

$$\mathbf{v}_i = \mathbf{x}_{r_1} + \frac{F}{\sqrt{k}} \sum_{j=1}^k (\mathbf{x}_{r_{2j}} - \mathbf{x}_{r_{2j+1}}) \quad (7)$$

That is, by setting scaling factor to  $F/\sqrt{k}$ , mutation range of  $k$  difference vectors is equivalent to DE/rand/1 with  $F$ .

Figure 1 shows the one-dimensional frequency distribution when 10,000 individuals are uniformly generated in the interval of  $[-1.707, 1.707]$  and the mutant vectors generated by rand/1 and rand/5 with  $F = 1.0$ . Figure 1(a) shows the distribution of mutant vectors with Equation (2) (normal  $F$ ) and Figure 1(b) shows the distribution of mutant vectors with Equation (7) (adjusted  $F$ ). As shown in this figure, when the number of difference vectors increases, it can be adjusted to almost the same range as rand/1 by using Equation (7). Next, Figure 2 shows two-dimensional distribution of mutant vectors generated from bimodal population distribution. In DE, if the objective function landscape is multimodal, the distribution may be distributed separately in several valleys. When the population distribution is bimodal, the features of the two methods differ remarkably. As shown in Figure 2 if the distribution of population is divided, the distribution of mutant vector differs greatly in  $k = 1$  and  $k = 5$  with adjusted  $F$ . In rand/1, the distribution of mutant vectors is also divided into several subgroups. Contrastly, in rand/5, regardless of the distribution of the population, the distribution of the mutation vectors is close to the normal distribution. This is because of the central limit theorem. Even if the population is not normally distributed as shown in Figure 2(b), when the sample size ( $k$ )

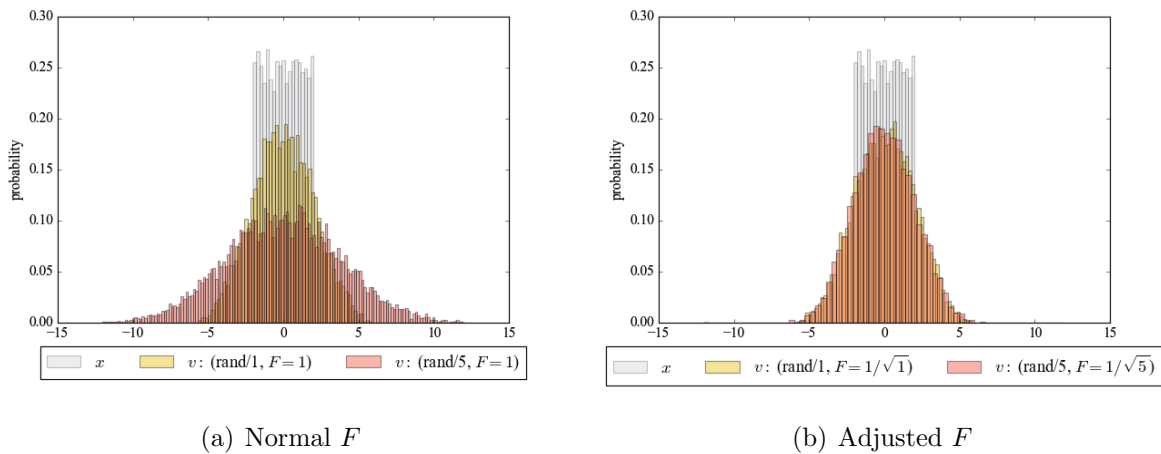


FIGURE 1. Distribution of mutant vectors (rand/1 and rand/5,  $F = 1.0$ ) generated from uniform distribution

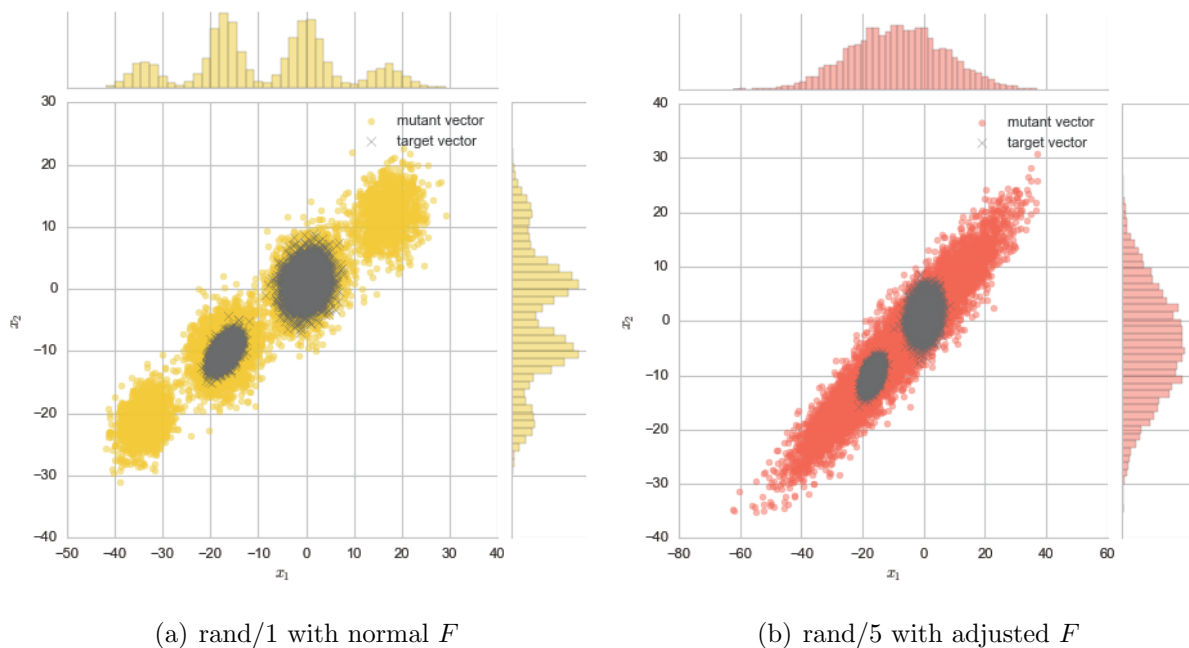


FIGURE 2. Distribution of mutant vectors (rand/1 and rand/5,  $F = 1.0$ ) generated from bimodal distribution

is sufficiently large, then the sample means (i.e., mutant vectors distribution) will have an approximately normal distribution.

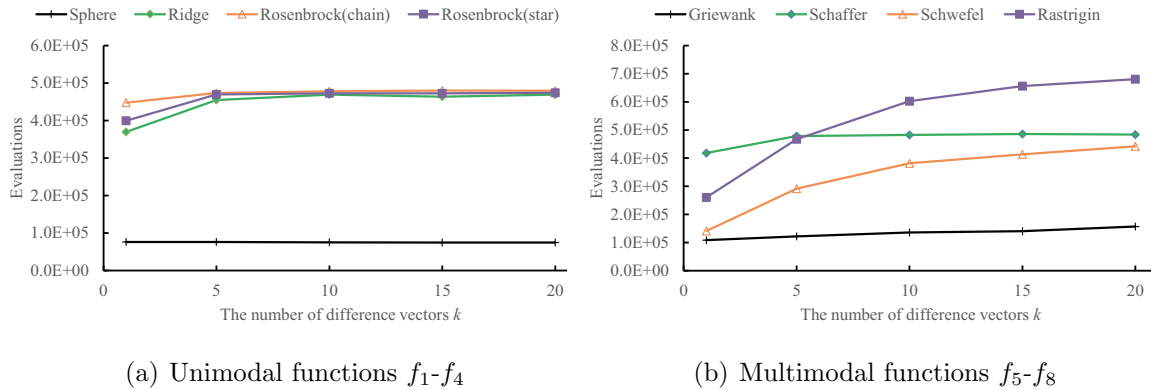
From above results, multivector mutation enables unbiased search around means of current population even if population is distributed on multimodal landscape. We think that such a search property of multivector mutation has ability to maintain population diversity and prevents falling into a local optima in multimodal functions.

**3. Impact of Multivector Mutation in DE.** In this chapter, we investigate the influence of multiple vector mutation on the performance of conventional DE using standard benchmark functions. The mathematical formulas and properties of these functions are shown in Table 1, where dimension  $D = 30$ . All functions are chosen for the minimization problems. The parameters settings for DE are as follows – the strategy is DE/rand/ $k$ /exp with adjusted  $F$ , the population size  $NP = 50$ ,  $F = 0.7$ , and  $CR = 0.95$ .

We present the relative performance of DE/rand/ $k$ /exp with  $k \in \{1, 5, 10, 15, 20\}$ . Performance at each setting is measured as the number of function evaluations (NFEs) needed for satisfying the success condition over 20 runs. The success condition defined as  $f(\mathbf{x}) - f(\mathbf{x}^*)$  is less than  $\varepsilon = 10^{-7}$ , where  $\mathbf{x}^*$  is the global optimum of function  $f$ . Figure 3 shows the average NFEs for each benchmark function. As can be seen from these figures, the impact of  $k$  depends on the objective function’s modality. In Sphere function, the performance is almost the same for any  $k$ . For other unimodal functions, as  $k$  increases to 5, the search speed decreases. However, when  $k > 5$  the search performance is almost constant. In Rastrigin and Schwefel functions, the search speed decreases with increasing  $k$ . They have many local optima, and their fitness landscapes are very rugged. In these functions, the distribution of the population may be divided to some valleys during the search. In multimodal landscape where population is divided into several subpopulations, the search speed by multivector mutation obeying approximate normal distribution decreases. This is because multivector mutation does not intensively search the neighborhood of each subpopulation. In this situation, multivector mutation also

TABLE 1. Benchmark functions

Name	Expression	Domain
$F_1$ : Sphere	$f(\mathbf{x}) = \sum_{i=1}^D x_i^2$	$[-5.12, 5.12]^D$
$F_2$ : Ridge	$f(\mathbf{x}) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$	$[-64, 64]^D$
$F_3$ : Rosenbrock (chain)	$f(\mathbf{x}) = \sum_{i=1}^{D-1} \left\{ 100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right\}$	$[-2.048, 2.048]^D$
$F_4$ : Rosenbrock (star)	$f(\mathbf{x}) = \sum_{i=2}^D \left\{ 100 (x_1 - x_i^2)^2 + (x_i - 1)^2 \right\}$	$[-2.048, 2.048]^D$
$F_5$ : Griewank	$f(\mathbf{x}) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \left( \cos \left( \frac{x_i}{\sqrt{i}} \right) \right)$	$[-512, 512]^D$
$F_6$ : Schaffer	$f(\mathbf{x}) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{0.25} \times \left\{ \sin^2 \left( 50 (x_i^2 + x_{i+1}^2)^{0.1} \right) + 1 \right\}$	$[-100, 100]^D$
$F_7$ : Schwefel	$f(\mathbf{x}) = \sum_{i=1}^D -x_i \sin \left( \sqrt{ x_i } \right)$	$[-512, 512]^D$
$F_8$ : Rastrigin	$f(\mathbf{x}) = 10D + \sum_{i=1}^D x_i^2 - 10 \cos(2\pi x_i)$	$[-5.12, 5.12]^D$

FIGURE 3. Comparison of search performance of DE/rand/ $k$ 

searches for hill points that will exist between subpopulations. Although the Griewank function is multimodal landscape, the impact of  $k$  is very small, because the ruggedness of landscape is small compared to other multimodal functions.

Next, we present the transition of fitness at each  $k$  in Rastrigin function and Schwefel function which are multimodal functions. Figure 4 shows the evolution graph of the average best fitness in 20 runs. In Rastrigin function, individuals distribute on multimodal landscape at the first stage of the search. Toward the end of the search, individuals gather in one valley and the landscape will seem to be unimodal landscape. Therefore, as  $k$  increases, the first phase where the fitness gently decreases becomes longer. Also, the transition of fitness in the last stage where the function value rapidly decreases is almost the same for any  $k$ . On the other hand, the landscape of Schwefel function is multimodal during the search, and thus overall convergence speed is delayed as  $k$  increases.

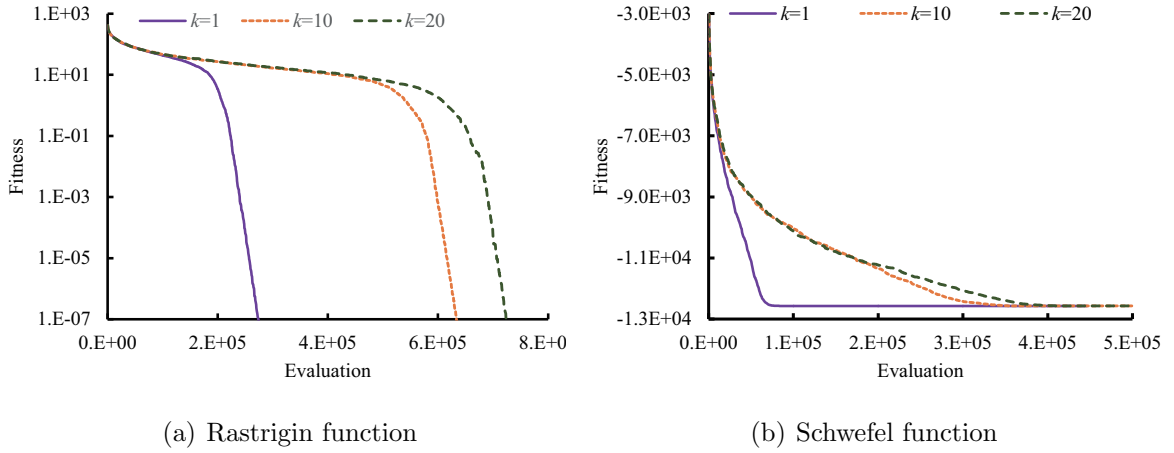


FIGURE 4. The evolution graph of the average best fitness in 20 runs

**4. RDE with Multivector Mutation.** The rank-based DE (RDE) is one of the DE variants that adopts an observation-based control of algorithm parameters and a simple modification in offspring generation [5]. In the original RDE, strategy is rand/1/exp and continuous generation model is adopted. In RDE, different parameter values are assigned based on goodness of base vector. When the base vector is good, a small scaling factor and a large crossover rate are selected and convergence is realized. Also, when the base vector is bad, a large scaling factor and a small crossover rate are selected and the divergence is realized. At the beginning of each generation, the ranks  $R_i$  of the individual vectors  $\mathbf{x}_i$  are given according to the fitness. First, the population is sorted in ascending order (i.e., from the best to the worst) based on the fitness of each individual. Then, the ranking of a vector is assigned as  $R_i = i$  ( $i = 1, 2, \dots, NP$ ), where the best vector in the current population will obtain the highest ranking ( $R_i = 1$ ). Before mutation, different values of  $F$  and  $CR$  are assigned to each target vector  $\mathbf{x}_i$  according to the rank of the base vector  $\mathbf{x}_{r_1}$ . The scaling factor  $F_i$  and the crossover rate  $CR_i$  for  $\mathbf{x}_i$  can be defined by the following equations:

$$F_i = F_{\min} + (F_{\max} - F_{\min}) \frac{R_{r_1} - 1}{NP - 1} \quad (8)$$

$$CR_i = CR_{\max} - (CR_{\max} - CR_{\min}) \frac{R_{r_1} - 1}{NP - 1} \quad (9)$$

where  $R_{r_1}$  is the rank of base vector,  $F_{\min}$ ,  $F_{\max}$  are parameters to specify the minimum and maximum value of  $F$ , and  $CR_{\min}$ ,  $CR_{\max}$  are parameters to specify the minimum and maximum value of  $CR$ . If the base vector is the best individual,  $F$  becomes the minimum value and  $CR$  becomes the maximum value. If the base vector is the worst individual,  $F$  becomes the maximum value and  $CR$  becomes the minimum value.

In this paper, we incorporate multivector mutation to the RDE (RDE-MM). A large value of  $k$  improves the exploration abilities as shown in Section 3, while the convergence speed may degrade. Therefore, we apply the parameter assignment mechanism of RDE to control of  $k$ . To balance between exploration and exploitation, the following two equations are used.

“Increasing  $k$  with a demotion of rank”:

$$k_i = k_{\min} + (k_{\max} - k_{\min}) \frac{R_{r_1} - 1}{NP - 1} \quad (10)$$

“Decreasing  $k$  with a demotion of rank”:

$$k_i = k_{\max} - (k_{\max} - k_{\min}) \frac{R_{r_1} - 1}{NP - 1} \quad (11)$$

where  $k_{\min}$ ,  $k_{\max}$  are parameters to specify the minimum and maximum value of  $k$ . In Equation (10), the better the base vector’s rank is, the smaller the number of  $k$  is. Equation (11) is opposite setting of Equation (10), i.e., the better the base vector’s rank is, the larger the number of  $k$  is. After allocating  $F_i$ ,  $CR_i$  and  $k_i$  for each target vector  $\mathbf{x}_i$ , mutant vector  $\mathbf{v}_i$  is generated by the following:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + \frac{F_i}{\sqrt{k_i}} \sum_{j=1}^{k_i} (\mathbf{x}_{r_{2j}} - \mathbf{x}_{r_{2j+1}}) \quad (12)$$

The algorithm of the RDE-MM is shown in Algorithm 2. Except for allocation of parameter  $k$ , it is the same procedure as normal RDE algorithm.

---

#### Algorithm 2 RDE-MM

---

```

1: /*Initialize a population*/
2:  $G = 0$ ;
3:  $P = NP$  individuals  $\{\mathbf{x}_i\}$  generated randomly in  $S$ ;
4: Set  $F_{\min}$ ,  $F_{\max}$ ,  $CR_{\min}$ ,  $CR_{\max}$ ,  $k_{\min}$ ,  $k_{\max}$ ;
5: for  $G = 1$  to  $G_{\max}$  do
6:    $\{R_i\} =$  ranks of  $\{\mathbf{x}_i\}$  according to fitness;
7:   for  $i = 1$  to  $NP$  do
8:     /*DE operation*/
9:      $\mathbf{x}_{r_1} =$  randomly selected from  $P$  ( $r_1 \neq i$ );
10:     $F_i$  and  $CR_i$  are determined by Equation (8) and Equation (9);
11:     $k_i$  is determined by Equation (10) or Equation (11);
12:     $\mathbf{v}_i =$  mutant vector generated by Equation (12);
13:     $\mathbf{u}_i =$  trial vector generated from  $\mathbf{x}_i$  and  $\mathbf{v}_i$  by a crossover;
14:    if  $f(\mathbf{u}_i) \leq f(\mathbf{x}_i)$  then
15:       $\mathbf{x}_i = \mathbf{u}_i$ ;
16:    else
17:       $\mathbf{x}_i = \mathbf{x}_i$ ;
18:    end if
19:  end for
20: end for

```

---

5. **Experiment.** In this section, we evaluate the performance of RDE-MM on the benchmark functions as shown in Table 1, where dimension  $D = 60$ . To investigate the effect of the multivector mutation on RDE, we run RDE-MM with Equation (10) (named RDE-MMi) and RDE-MM with Equation (11) (named RDE-MMd) for each function. The common parameters settings for RDE and RDE-MM are as follows – the strategy is RDE/rand/1/exp, the population size  $NP = 40$ ,  $F_{\min} = 0.5$ ,  $F_{\max} = 1.0$ ,  $CR_{\min} = 0.1$ ,  $CR_{\max} = 1.0$ . Normally,  $NP$  is recommended to be  $5D$  or more [1]; however, we set small  $NP$  in this experiment for confirming the ability to maintain population diversity of the multivector mutation. Performance at each setting is measured as the NFEs needed for satisfying the success condition over 50 runs. The success condition is the same as Section 3 and the maximum generation is  $G_{\max} = 10^5$ . In RDE-MMi and RDE-MMd  $k_{\min}$  is set to 1 and  $k_{\max}$  is across  $\{5, 10, 15\}$ .



TABLE 2. The average NFE of RDE and RDE-MM

Function	RDE/ rand/1	RDE-MMi			RDE-MMd		
		$k_{\max} = 5$	$k_{\max} = 10$	$k_{\max} = 15$	$k_{\max} = 5$	$k_{\max} = 10$	$k_{\max} = 15$
Sphere	<b>5.04E+04</b> [50]	5.08E+04 [50]	5.04E+04 [50]	5.08E+04 [50]	5.55E+04 [50]	5.64E+04 [50]	5.64E+04 [50]
Ridge	5.86E+05 [50]	5.84E+05 [50]	5.83E+05 [50]	5.85E+05 [50]	5.71E+05 [50]	5.71E+05 [50]	<b>5.66E+05</b> [50]
Rosenbrock (chain)	5.09E+05 [43]	5.20E+05 [49]	5.16E+05 [42]	5.17E+05 [43]	5.07E+05 [47]	<b>5.08E+05</b> [49]	5.09E+05 [49]
Rosenbrock (star)	— [0]	— [0]	— [0]	— [0]	9.40E+05 [17]	8.83E+05 [21]	<b>9.10E+05</b> [26]
Griewank	6.49E+04 [45]	6.54E+04 [44]	6.41E+04 [49]	6.48E+04 [43]	<b>7.16E+04</b> [50]	7.46E+04 [50]	7.57E+04 [50]
Schaffer	<b>3.25E+05</b> [50]	3.40E+05 [50]	3.48E+05 [50]	3.53E+05 [50]	3.65E+05 [50]	3.74E+05 [50]	3.78E+05 [50]
Schwefel	8.88E+04 [49]	1.08E+05 [49]	<b>1.29E+05</b> [50]	1.44E+05 [50]	1.13E+05 [49]	1.32E+05 [50]	1.47E+05 [50]
Rastrigin	<b>1.04E+05</b> [50]	1.29E+05 [50]	1.54E+05 [50]	1.71E+05 [50]	1.26E+05 [50]	1.45E+05 [50]	1.63E+05 [50]

Table 2 shows the result of RDE-MMi and RDE-MMd. For each function, the average NFEs over successful runs are shown in the top row. The number of success runs is shown in the bottom row. For Sphere function, RDE/rand/1 and RDE-MMi are faster than RDE-MMd. However, for Ridge function which has dependency between variables, RDE-MMd obtains better results than other methods. Also, for Rosenbrock functions with strong dependency between variables, the stability of all methods decreases and they cannot find the optimal solution in all trials. Especially in Rosenbrock (star) function, RDE and RDE-MMi fail in all trials. In the search of RDE, the population converged to a valley with a local solution in several dimensions. On the other hand, RDE-MMd succeeded in some trials, and the number of success trials becomes larger with increasing  $k$ . RDE successfully searches for Schaffer and Rastrigin functions in all trials; however, some failures occur for Griewank and Schwefel functions. While RDE MMd with  $k_{\max} = \{10, 15\}$  can find solutions in all trials.

From the above results, we conclude that RDE-MMd can realize more stable search than RDE and is a method effectively utilizing the characteristics of the multivector mutation. Although RDE-MMi converges more quickly than RDE-MMd, considering the results of Rosenbrock (star) function, RDE-MMd is better in terms of stability.

**6. Conclusion.** In this study, we focused on multivector mutation in DE algorithm and combined it with RDE. At first, we analyzed impact of the number of difference vectors in multivector mutation using standard benchmark functions. From experimental analysis, we confirmed that multivector mutation enables unbiased search around means of current population and it has ability to maintain population diversity. Next, we proposed controlling method of the number of difference vectors in RDE with multivector mutation. In the proposed RDE with multivector mutation, named RDE-MM, the number of difference vectors of each target vector is determined based on the ranking information of the current population. In the experiment, we compared original RDE and RDE-MM using standard benchmark functions. In RDE-MM, we used two controlling methods: increasing  $k$  or decreasing  $k$  with a demotion of base vector's rank. From the experimental results using basic benchmark functions, we showed that the RDE-MM with an adequate control of  $k$  can provide enough population diversity and improve the search ability of RDE.

In this paper, we have confirmed the validity of multivector mutation with a number of difference vectors, but we think that DE/rand/ $\infty$  which assumes  $k \rightarrow \infty$  [3] is also effective mutation operator. In the future, we will investigate the search performance and characteristics of DRE/rand/ $\infty$ . Furthermore, we intend to improve performance of our previous RDE algorithms [6, 7] by incorporating multivector mutation strategy.

#### REFERENCES

- [1] 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, no.4, pp.341-359, 1997.
- [2] S. Das and P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evolutionary Computation*, vol.15, no.1, pp.4-31, 2011.
- [3] K. Opara and J. Arabas, Differential mutation based on population covariance matrix, *Proc. of Parallel Problem Solving from Nature, PPSN XI*, pp.114-123, 2010.
- [4] K. Opara and J. Arabas, Decomposition and metaoptimization of mutation operator in differential evolution, *Swarm and Evolutionary Computation*, pp.110-118, 2012.
- [5] T. Takahama and S. Sakai, Efficient constrained optimization by the epsilon constrained rank-based differential evolution, *Proc. of 2012 IEEE Congress on Evolutionary Computation*, pp.62-69, 2012.
- [6] J. Kushida, A. Hara and T. Takahama, Rank-based differential evolution with multiple mutation strategies for large scale global optimization, *Proc. of 2015 IEEE Congress on Evolutionary Computation*, pp.353-360, 2015.
- [7] J. Kushida, A. Hara and T. Takahama, Rank-based differential evolution with eigenvector-based crossover operator, *ICIC Express Letters*, vol.10, no.11, pp.2547-2553, 2016.
- [8] R. Tanabe and F. Alex, Reevaluating exponential crossover in differential evolution, *Proc. of Parallel Problem Solving from Nature, PPSN XIII*, pp.201-210, 2014.