

# Large Scale Optimization by Differential Evolution with Landscape Modality Detection and a Diversity Archive

Tetsuyuki Takahama

Department of Intelligent Systems  
Hiroshima City University  
Asaminami-ku, Hiroshima, 731-3194 Japan  
Email: takahama@info.hiroshima-cu.ac.jp

Setsuko Sakai

Faculty of Commercial Sciences  
Hiroshima Shudo University  
Asaminami-ku, Hiroshima, 731-3195 Japan  
Email: setuko@shudo-u.ac.jp

**Abstract**—In this study, the performance of Differential Evolution with landscape modality detection and a diversity archive (LMDEa) is reported on the set of benchmark functions provided for the CEC2012 Special Session on Large Scale Global Optimization. In Differential Evolution (DE), large population size, which is much larger than the number of decision variables in problem to be solved, is adopted in order to keep the diversity of search. However, it is difficult to adopt such large size to solve large scaled optimization problems because the population size will become too large and the search efficiency will degrade. In this study, we propose to solve large scale optimization problems using small population size and a large archive for diversity. Also, we propose simple control of scaling factor by observing landscape modality of search points in order to keep diversity. The landscape of a problem to be optimized is often unknown and the landscape is changing dynamically while the search process proceeds. In LMDEa, some points on a line connecting the centroid of search points and a search point are sampled. When the objective values of the sampled points are changed decreasingly and then increasingly, it is thought that one valley exists. If there exists only one valley, the landscape is unimodal and small scaling factor is adopted. Otherwise, large scaling factor is adopted. Also, the sampled points realize global search in the region spanned by all search points and realize local search near the best search point. The effect of the proposed method is shown by solving the benchmark functions.

**Keywords**—differential evolution; large scale optimization; landscape modality; parameter control

## I. INTRODUCTION

Optimization problems, especially nonlinear optimization problems, are very important and frequently appear in the real world. There exist many studies on solving optimization problems using evolutionary algorithms (EAs). Differential evolution (DE) is a newly proposed EA by Storn and Price [1]. DE has been successfully applied to optimization problems including non-linear, non-differentiable, non-convex and multimodal functions [2]–[4]. It has been shown that DE is a very fast and robust algorithm.

In EAs, large population size is effective to keep the diversity of search points, although the efficiency of evolution becomes low. In DE, large population size, which is much larger than the number of decision variables in problem to be solved, is usually adopted in order to keep the diversity. However, in large scaled optimization, it is difficult to adopt such large size because the size becomes too large and the

search efficiency will degrade. If small population size is adopted, it needs to adopt methods for keeping the diversity. Some methods are proposed to improve the diversity: (1) using an external archive [5], or a diversity archive [6] in addition to the population, and (2) controlling algorithm parameters such as the scaling factor and the crossover rate properly.

In this study, we propose Differential Evolution with Landscape Modality detection and a diversity Archive (LMDEa) to solve large scale optimization problems using small population size. In order to keep the diversity, an archive, which is updated using defeated individuals in survivor selection of DE, is utilized in addition to the population.

A proper value of the scaling factor and also the crossover rate depends on the landscape of a problem to be optimized. However, the landscape of the problem is often unknown and the landscape is changing dynamically while the search process proceeds. Thus, it is difficult to select the proper value. In study, we propose simple control of scaling factor by detecting the landscape modality of search points: unimodal or not unimodal. In LMDEa, some points on a line connecting the centroid of search points and a search point are sampled. When the objective values of the sampled points are changed decreasingly and then increasingly, it is thought that one valley exists. If there exists only one valley, the landscape is unimodal. Otherwise, the landscape is not unimodal, or is multimodal. If the landscape is unimodal, small scaling factor is adopted. It is expected that the search points will converge to the valley of the unimodal landscape fast and the efficiency of the search is improved. If the landscape is not unimodal, large scaling factor is adopted. It is expected that the divergence of the search will be retained and premature convergence will be prevented. Also, the sampled points realize global search in the region spanned by all search points and realize local search near the best search point. The effect of the proposed method is shown by solving the benchmark functions including unimodal problems and multimodal problems in the CEC2012 Special Session on Large Scale Global Optimization [7].

In Section II, optimization problems and DE are explained. Related works are briefly reviewed in Section III. DE with landscape modality detection and a diversity archive is proposed in Section IV. In Section V, experimental results on some problems are shown. Finally, conclusions are described

in Section VI.

## II. OPTIMIZATION BY DIFFERENTIAL EVOLUTION

### A. Optimization Problems

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

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && l_i \leq x_i \leq u_i, \quad i = 1, \dots, D, \end{aligned} \quad (1)$$

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

### B. Differential Evolution

DE is a stochastic direct search method using a population or multiple search points.

In DE, initial individuals are randomly generated within given search space and form an initial population. Each individual contains  $D$  genes as decision variables. At each generation or iteration, all individuals are selected as parents. Each parent is processed as follows: The mutation operation begins by choosing several individuals from the population except for the parent in the processing. The first individual is a base vector. All subsequent individuals are paired to create difference vectors. The difference vectors are scaled by a scaling factor  $F$  and added to the base vector. The resulting vector, or a mutant vector, is then recombined with the parent. The probability of recombination at an element is controlled by a crossover rate  $CR$ . This crossover operation produces a trial vector. Finally, for survivor selection, the trial vector is accepted for the next generation if the trial vector is better than the parent.

There are some variants of DE that have been proposed. The variants are classified using the notation  $DE/base/num/cross$  such as  $DE/rand/1/bin$  and  $DE/rand/1/exp$ .

“*base*” specifies a way of selecting an individual that will form the base vector. For example,  $DE/rand$  selects an individual for the base vector at random from the population.  $DE/best$  selects the best individual in the population.

“*num*” specifies the number of difference vectors used to perturb the base vector. In case of  $DE/rand/1$ , for example, for each parent  $\mathbf{x}^i$ , three individuals  $\mathbf{x}^{p1}$ ,  $\mathbf{x}^{p2}$  and  $\mathbf{x}^{p3}$  are chosen randomly from the population without overlapping  $\mathbf{x}^i$  and each other. A new vector, or a mutant vector  $\mathbf{x}'$  is generated by the base vector  $\mathbf{x}^{p1}$  and the difference vector  $\mathbf{x}^{p2} - \mathbf{x}^{p3}$ , where  $F$  is the scaling factor.

$$\mathbf{x}' = \mathbf{x}^{p1} + F(\mathbf{x}^{p2} - \mathbf{x}^{p3}) \quad (2)$$

“*cross*” specifies the type of crossover that is used to create a child. For example, ‘bin’ indicates that the crossover is controlled by the binomial crossover using a constant crossover rate, and ‘exp’ indicates that the crossover is controlled by a kind of two-point crossover using exponentially decreasing

the crossover rate. Fig. 1 shows the binomial and exponential crossover. A new child  $\mathbf{x}^{child}$  is generated from the parent  $\mathbf{x}^i$  and the mutant vector  $\mathbf{x}'$ , where  $CR$  is a crossover rate.

```

binomial crossover DE/./bin
jrand=randint(1,D);
for(k=1; k ≤ D; k++) {
    if(k == jrand || u(0,1) < CR) x_k^child = x'_k;
    else x_k^child = x_k^i;
}
exponential crossover DE/./exp
k=1; j=randint(1,D);
do {
    x_j^child = x'_j;
    k=k+1; j=j+1; if(j > D) j=1;
} while(k ≤ D && u(0,1) < CR);
while(k ≤ D) {
    x_j^child = x_k^i;
    k=k+1; j=j+1; if(j > D) j=1;
}

```

Fig. 1. Binomial and exponential crossover operation, where  $\text{randint}(1,D)$  generates an integer randomly from  $[1, D]$  and  $u(l, r)$  is a uniform random number generator in  $[l, r]$ .

### C. The Algorithm of Differential Evolution

The algorithm of DE is as follows:

- Step1 Initialization of a population. Initial  $N$  individuals  $P = \{\mathbf{x}^i, i = 1, 2, \dots, N\}$  are generated randomly in search space and form an initial population.
- Step2 Termination condition. If the number of function evaluations exceeds the maximum number of evaluation  $FE_{max}$ , the algorithm is terminated.
- Step3 DE operations. Each individual  $\mathbf{x}^i$  is selected as a parent. If all individuals are selected, go to Step4. A mutant vector  $\mathbf{x}'$  is generated according to Eq. (2). A trial vector (child) is generated from the parent  $\mathbf{x}^i$  and the mutant vector  $\mathbf{x}'$  using a crossover operation shown in Fig. 1. If the child is better than or equal to the parent, or the DE operation is succeeded, the child survives. Otherwise the parent survives. Go back to Step3 and the next individual is selected as a parent.
- Step4 Survivor selection (generation change). The population is organized by the survivors. Go back to Step2.

Fig. 2 shows a pseudo-code of  $DE/rand/1$ .

## III. RELATED WORKS

The performance of DE is affected by control parameters such as the scaling factor  $F$ , the crossover rate  $CR$  and the population size  $N$ , and by mutation strategies such as the rand strategy and the best strategy. Many researchers have been studying on controlling the parameters and the strategies. The methods of the control can be classified into two categories: observation-based and success-based control.

- 1) observation-based control: The current search state is observed, proper parameter values are inferred according to the observation, and parameters and/or strategies are

```

DE/rand/1()
{
// Initialize an population
P=N individuals generated randomly in S;
for(t=1; FE ≤ FEmax; t++) {
for(i=1; i ≤ N; i++) {
// DE operation
xp1=Randomly selected from P(p1 ≠ i);
xp2=Randomly selected from P(p2 ≠ i ≠ p1);
xp3=Randomly selected from P(p3 ≠ i ≠ p1 ≠ p2);
x' = xp1 + F(xp2 - xp3);
xchild=trial vector is generated from
xi and x' by the crossover operation;
// Survivor selection
if(f(xchild) ≤ f(xi)) zi = xchild;
else zi = xi;
FE = FE + 1;
}
P = {zi, i = 1, 2, ..., N};
}
}

```

Fig. 2. The pseudo-code of DE,  $FE$  is the number of function evaluations.

dynamically controlled. FADE(Fuzzy Adaptive DE) [8] observes the movement of search points and the change of function values between successive generations, and controls  $F$  and  $CR$ . DESFC(DE with Speciation and Fuzzy Clustering) [9] adopts fuzzy clustering, observes partition entropy of search points, and controls  $CR$  and the mutation strategies between the rand and the species-best strategy.

- 2) success-based control: It is recognized as a success case when a better search point than the parent is generated. The parameters and/or strategies are adjusted so that the values in the success cases are frequently used. It is thought that the self-adaptation, where parameters are contained in individuals and are evolved by applying evolutionary operators to the parameters, is included in this category. DESAP(Differential Evolution with Self-Adapting Populations) [10] controls  $F$ ,  $CR$  and  $N$  self-adaptively. SaDE(Self-adaptive DE) [11] controls the mean value of  $CR$  according to the mean value in success cases and controls the selection probability of the mutation strategies according to the success rates. jDE(self-adaptive DE algorithm) [12] controls  $F$  and  $CR$  self-adaptively. JADE(adaptive DE with optional external archive) [5] controls the mean values of  $F$  and  $CR$  according to the mean values in success cases.

In the category 1), it is difficult to select proper type of observation which is independent of the optimization problem and its scale. In the category 2), when a new good search point is found near the parent, parameters are adjusted to the direction of convergence. In problems with ridge landscape or multimodal landscape, where good search points exist in small region, parameters are tuned for small success and big success will be missed. Thus, search process would be trapped at a local optimal solution.

In this study, we propose new observation-based control in the category 1). In the control,  $F$  is selected according to the landscape modality which is inferred by a kind of line search.

#### IV. DIFFERENTIAL EVOLUTION WITH DETECTING LANDSCAPE MODALITY

In this section, landscape modality detection is explained.

##### A. Detecting Landscape Modality

In this study, whether the search points are in unimodal landscape or not is detected using the current search points  $P = \{\mathbf{x}^i | i = 1, 2, \dots, N\}$ . The objective values are examined along the following line, which connects the centroid of search points  $\mathbf{x}^g$  and the best search point  $\mathbf{x}^b$ .

$$\mathbf{x} = \mathbf{x}^g + \lambda(\mathbf{x}^b - \mathbf{x}^g) \quad (3)$$

$$\mathbf{x}^g = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^i \quad (4)$$

$$\mathbf{x}^b = \arg \min_i f(\mathbf{x}^i) \quad (5)$$

where  $\lambda$  is a parameter for deciding the position of a point on the line. The range of the search points  $[\mathbf{x}^{\min}, \mathbf{x}^{\max}]$  can be given as follows:

$$x_j^{\min} = \min_i x_j^i \quad (6)$$

$$x_j^{\max} = \max_i x_j^i \quad (7)$$

The range of the  $\lambda$ ,  $[\lambda^{\min}, \lambda^{\max}]$  satisfies the following condition:

$$x_j^{\min} \leq x_j^g + \lambda(x_j^b - x_j^g) \leq x_j^{\max} \quad (8)$$

Thus, the range of the  $\lambda$  is given by the following:

$$\lambda^{\min} = \max_j \frac{x_j^{\min} - x_j^g}{x_j^b - x_j^g} \quad (9)$$

$$\lambda^{\max} = \min_j \frac{x_j^{\max} - x_j^g}{x_j^b - x_j^g} \quad (10)$$

If  $(x_j^b - x_j^g)$  is negative,  $x_j^{\min}$  and  $x_j^{\max}$  in the equations are exchanged.

In order to decide  $M$  sampling points  $\{\mathbf{x}_k | k = 1, 2, \dots, M\}$ ,  $\lambda_k$  is given as follows:

$$\lambda_k = \lambda^{\min} + \frac{\lambda^{\max} - \lambda^{\min}}{M - 1}(k - 1) \quad (11)$$

$$\mathbf{x}_k = \mathbf{x}^g + \lambda_k(\mathbf{x}^b - \mathbf{x}^g) \quad (12)$$

Figure 3 shows an example of the sampling, where search points are shown by black circles, the centroid is shown by a white circle, sampling points are shown by triangles in case of  $M = 6$ .

In the sequence  $\{f(\mathbf{x}_k) | k = 1, 2, \dots, M\}$ , hill-valley relation is examined. For each point, the function  $dir$  is introduced in order to judge whether the change is increasing or decreasing.

$$dir(\mathbf{x}_k) = \begin{cases} 1 & (f(\mathbf{x}_{k+1}) > f(\mathbf{x}_k)) \\ -1 & (f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)) \\ dir(\mathbf{x}_{k-1}) & (\text{otherwise}) \end{cases} \quad (13)$$

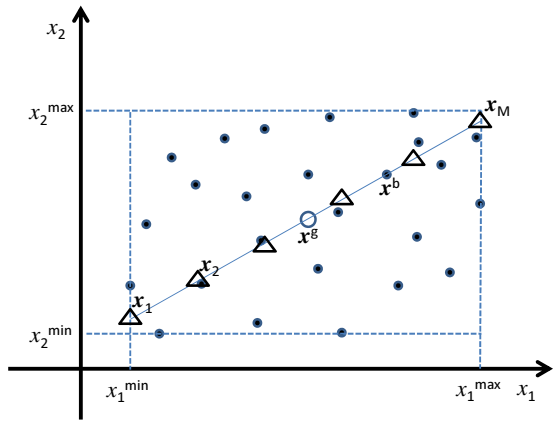


Fig. 3. An example of sampling for detecting landscape modality

If the value of  $dir$  changed from -1 to 1 only once, it is thought that one valley exists and the landscape is unimodal. Otherwise, the landscape is not unimodal. Figure 4 shows an example of detecting unimodal landscape, where the objective values are shown by the function of  $\lambda$ .

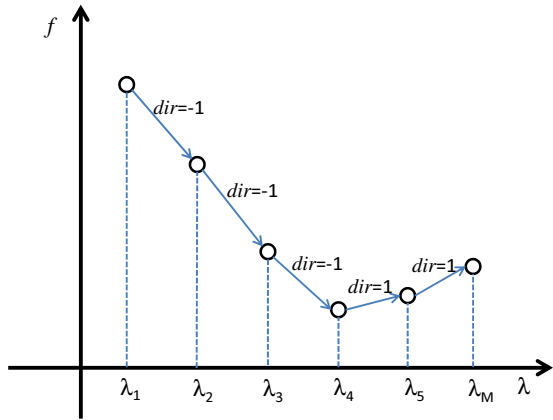


Fig. 4. An example of detecting unimodal landscape

If the best value of  $f(\mathbf{x}_k)$  is better than  $f(\mathbf{x}^b)$ ,  $\mathbf{x}^b$  is replaced by the  $\mathbf{x}_k$ .

$$\mathbf{x}^b = \begin{cases} \arg \min_k f(\mathbf{x}_k) & (\min_k f(\mathbf{x}_k) < f(\mathbf{x}^b)) \\ \mathbf{x}^b & (\text{otherwise}) \end{cases} \quad (14)$$

It is thought that the sampling performs global search in the region spanned by all search points and also perform local search near the best search point.

### B. Algorithm of LMDEa

Fig. 5 shows the pseudo-code of LMDEa.

Some modifications to standard DE are applied for proposed method as follows:

- 1) A diversity archive  $A$  with the maximum size  $N_A$  is introduced. Defeated trial vectors in survivor selection are added to the archive. If the archive size exceeds  $N_A$ , a randomly selected element is deleted. In the mutation

```

LMDEa/rand/1/exp+bin()
{
  F=F0; FE=0; A=phi;
  // Initialize a population
  P=N individuals generated randomly in S;
  FE=FE+N;
  for(t=1; FE <= FE_max; t++) {
    if(t%Td==Td-1) {
      Detecting landscape modality;
      FE=FE+M;
      if(landscape is unimodal) F=F0;
      else F=F0+0.2;
    }
    for(i=1; i <= N; i++)
      for(k=1; k <= 2; k++) {
        if(k==1) {
          // DE/rand/1/exp operation
          exponential crossover is selected;
          CR=0.8+0.2u(0,1);
        }
        else {
          // DE/rand/1/bin operation
          binomial crossover is selected;
          CR=u(0,1);
        }
        xp1=Randomly selected from P(p1 != i);
        xp2=Randomly selected from P(p2 != {i,p1});
        xp3=Randomly selected from P union A(p3 not in {p1,p2});
        x'=xp1+F(xp2-xp3);
        xchild=trial vector is generated from xi and x'
          by the selected crossover operation;
        FE=FE+1;
        // Survivor selection
        if(f(xchild) <= f(xi)) {
          xi=xchild;
          break;
        }
        else {
          if(|A| < NA) A=A union {xchild};
          else replace a random element in A with xchild;
        }
      }
    }
  }
}

```

Fig. 5. The pseudo-code of LMDEa where  $A$  is a diversity archive,  $N_A$  is the maximum size of the archive and  $T_d$  is the term of detecting landscape modality.

- operation, the second vector  $\mathbf{x}^{p3}$  in a difference vector is selected from the population and the archive ( $P \cup A$ ).
- 2) Dynamic parameter selection are performed according to landscape modality. If current landscape is unimodal, the base scaling factor  $F_0$  is adopted as  $F$ . Otherwise,  $F = F_0 + 0.2$ .
- 3) Two types of crossover operations, exponential and binomial crossover operations, are adopted: DE/rand/1/exp operation with a random crossover rate in  $[0.8, 1]$  is used and a new child is generated ( $k = 1$ ). If the new one is not better than the parent, DE/rand/1/bin operation with a random crossover rate in  $[0, 1]$  is used and another new child is generated ( $k = 2$ ).

- 4) Continuous generation model [13] is adopted. Usually discrete generation model is adopted in DE and when the child is better than the parent, the child survives in the next generation. In this study, when the child is better than the parent, the parent is immediately replaced by the child. It is thought that the continuous generation model improves efficiency because the model can use newer information than the discrete model.
- 5) Reflecting back out-of-bound solutions [14] is adopted. In order to keep bound constraints, an operation to move a point outside of the search space  $\mathcal{S}$  into the inside of  $\mathcal{S}$  is required. There are some ways to realize the movement: generating solutions again, cutting off the solutions on the boundary, and reflecting points back to the inside of the boundary [15]. In this study, reflecting back is used:

$$x_{ij} = \begin{cases} l_i + (l_i - x_{ij}) - \left\lfloor \frac{l_i - x_{ij}}{u_i - l_i} \right\rfloor (u_i - l_i) & (x_{ij} < l_i) \\ u_i - (x_{ij} - u_i) + \left\lfloor \frac{x_{ij} - u_i}{u_i - l_i} \right\rfloor (u_i - l_i) & (x_{ij} > u_i) \\ x_{ij} & (\text{otherwise}) \end{cases} \quad (15)$$

where  $\lfloor z \rfloor$  is the maximum integer smaller than or equal to  $z$ . This operation is applied when a new point is generated by DE operations.

## V. SOLVING OPTIMIZATION PROBLEMS

In this paper, twenty benchmark functions are solved.

### A. Test Problems and Experimental Conditions

The twenty scalable benchmark functions can be classified to five types of functions as follows:

- 1) Separable Functions (3)
  - $F_1$ : Shifted Elliptic Function
  - $F_2$ : Shifted Rastrigin's Function
  - $F_3$ : Shifted Ackley's Function
- 2) Single-group  $m$ -nonseparable Functions (5)
  - $F_4$ : Single-group Shifted and  $m$ -rotated Elliptic Function
  - $F_5$ : Single-group Shifted and  $m$ -rotated Rastrigin's Function
  - $F_6$ : Single-group Shifted and  $m$ -rotated Ackley's Function
  - $F_7$ : Single-group Shifted  $m$ -dimensional Schwefel's Problem 1.2
  - $F_8$ : Single-group Shifted  $m$ -dimensional Rosenbrock's Function
- 3)  $D/2m$ -group  $m$ -nonseparable Functions (5)
  - $F_9$ :  $D/2m$ -group Shifted and  $m$ -rotated Elliptic Function
  - $F_{10}$ :  $D/2m$ -group Shifted and  $m$ -rotated Rastrigin's Function
  - $F_{11}$ :  $D/2m$ -group Shifted and  $m$ -rotated Ackley's Function
  - $F_{12}$ :  $D/2m$ -group Shifted  $m$ -dimensional Schwefel's Problem 1.2

$F_{13}$ :  $D/2m$ -group Shifted  $m$ -dimensional Rosenbrock's Function

- 4)  $D/m$ -group  $m$ -nonseparable Functions (5)
  - $F_{14}$ :  $D/m$ -group Shifted and  $m$ -rotated Elliptic Function
  - $F_{15}$ :  $D/m$ -group Shifted and  $m$ -rotated Rastrigin's Function
  - $F_{16}$ :  $D/m$ -group Shifted and  $m$ -rotated Ackley's Function
  - $F_{17}$ :  $D/m$ -group Shifted  $m$ -dimensional Schwefel's Problem 1.2
  - $F_{18}$ :  $D/m$ -group Shifted  $m$ -dimensional Rosenbrock's Function
- 5) Nonseparable Functions (2)
  - $F_{19}$ : Shifted Schwefel's Problem 1.2
  - $F_{20}$ : Shifted Rosenbrock's Function

where dimension  $D = 1000$  and group size  $m = 50$ . The optimum function values are 0 for all the problems.

Independent 25 runs are performed for 20 problems. Each run stops when the number of function evaluations (FEs) exceeds the maximum number of evaluations  $FE_{\max}=3,000,000$ .

### B. Experimental Results on the Proposed Method

The parameters settings for LMDEa are as follows: The population size  $N=60$ , the archive size  $N_A=3000$ , the initial scaling factor  $F_0=0.6$ , the term of detecting landscape modality  $T_d=20$ , and the number of sampling points is same as the population size or  $M=N$ .

Table I shows the experimental results on LMDEa. The best, median, worst, mean values and the standard deviation of best objective value in each run at 120,000FEs, 600,000FEs and 3,000,000FEs are shown for each function.

LMDEa found very good solutions less than 1 on average in 5 functions  $F_1, F_3, F_6, F_7, F_8$ .

Figures 6 to 13 show the convergence graph of functions  $F_2, F_5, F_8, F_{10}, F_{13}, F_{15}, F_{18}$  and  $F_{20}$ .

### C. Comparison with Other Methods

In this section the results obtained by LMDEa are compared with the ones obtained by other methods. Table II compares LMDEa with DECC-CG [16] and MLCC [17]. The best results are highlighted using boldface. From the table, the features of the results can be summarized as follows:

- LMDEa attained the best average results among all methods in 15 problems  $F_4-F_{15}$  and  $F_{18}-F_{20}$ . LMDEa

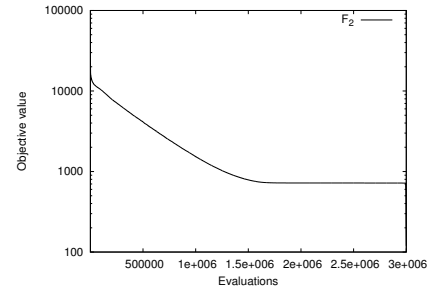


Fig. 6. The graph of  $F_2$

TABLE I  
EXPERIMENTAL RESULTS ON LMDEA. BEST, MEDIAN, WORST, MEAN VALUES AND STANDARD DEVIATIONS IN 25 RUNS ARE SHOWN

		$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$
FEs=1.2e+05	Best	4.40e+08	9.68e+03	1.43e+01	2.60e+13	2.35e+08	4.09e+04	6.05e+09
	Median	4.92e+08	9.84e+03	1.51e+01	6.08e+13	2.98e+08	6.44e+04	1.46e+10
	Worst	6.07e+08	1.02e+04	1.55e+01	9.40e+13	3.41e+08	1.53e+05	2.34e+10
	Mean	5.08e+08	9.89e+03	1.51e+01	6.25e+13	2.94e+08	6.90e+04	1.52e+10
	Std	4.76e+07	1.37e+02	2.50e-01	1.72e+13	2.51e+07	2.30e+04	3.94e+09
FEs=6.0e+05	Best	3.00e+02	3.23e+03	6.17e-01	1.65e+12	3.68e+07	4.73e+00	2.37e+07
	Median	4.32e+02	3.35e+03	9.23e-01	4.23e+12	6.17e+07	5.50e+00	5.03e+07
	Worst	6.90e+02	3.53e+03	1.10e+00	9.52e+12	1.50e+08	6.46e+00	1.34e+08
	Mean	4.59e+02	3.37e+03	9.16e-01	4.49e+12	7.21e+07	5.60e+00	5.85e+07
	Std	1.09e+02	6.88e+01	1.10e-01	1.81e+12	2.74e+07	3.91e-01	2.99e+07
FEs=3.0e+06	Best	2.42e-24	5.31e+02	7.86e-14	1.14e+11	3.68e+07	4.00e-09	3.79e-02
	Median	7.36e-24	6.87e+02	8.79e-01	2.06e+11	6.07e+07	4.02e-09	1.91e-01
	Worst	1.55e-22	8.62e+02	1.09e+00	3.15e+11	1.13e+08	1.01e+00	6.42e-01
	Mean	1.35e-23	6.97e+02	6.44e-01	2.08e+11	6.62e+07	2.63e-01	2.45e-01
	Std	2.91e-23	8.22e+01	4.46e-01	5.89e+10	2.02e+07	4.22e-01	1.68e-01
		$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$
FEs=1.2e+05	Best	6.79e+07	4.06e+09	1.22e+04	2.16e+02	2.67e+06	2.11e+07	7.54e+09
	Median	1.71e+08	5.25e+09	1.27e+04	2.26e+02	2.84e+06	2.87e+07	9.59e+09
	Worst	4.46e+08	6.33e+09	1.34e+04	2.29e+02	3.05e+06	4.58e+07	1.16e+10
	Mean	1.88e+08	5.21e+09	1.27e+04	2.25e+02	2.85e+06	2.90e+07	9.64e+09
	Std	9.84e+07	5.87e+08	2.67e+02	2.90e+00	1.07e+05	5.51e+06	8.85e+08
FEs=6.0e+05	Best	3.19e+07	2.26e+08	7.82e+03	4.18e+01	3.91e+05	1.48e+03	6.49e+08
	Median	3.30e+07	2.76e+08	9.68e+03	6.70e+01	4.51e+05	2.07e+03	7.55e+08
	Worst	3.49e+07	3.25e+08	1.02e+04	1.14e+02	5.17e+05	5.72e+03	1.01e+09
	Mean	3.32e+07	2.71e+08	9.65e+03	6.91e+01	4.50e+05	2.34e+03	7.70e+08
	Std	8.55e+05	2.40e+07	4.76e+02	1.71e+01	3.21e+04	9.90e+02	7.45e+07
FEs=3.0e+06	Best	7.80e-05	2.25e+07	2.45e+03	6.49e-11	1.35e+04	4.56e+02	7.65e+07
	Median	3.29e-04	2.65e+07	2.76e+03	2.74e+00	1.81e+04	5.64e+02	8.67e+07
	Worst	9.20e-04	2.98e+07	3.97e+03	5.40e+01	2.49e+04	9.38e+02	1.03e+08
	Mean	3.61e-04	2.64e+07	2.80e+03	1.19e+01	1.83e+04	5.95e+02	8.63e+07
	Std	2.33e-04	1.89e+06	2.84e+02	1.50e+01	2.62e+03	1.06e+02	6.30e+06
		$F_{15}$	$F_{16}$	$F_{17}$	$F_{18}$	$F_{19}$	$F_{20}$	
FEs=1.2e+05	Best	1.32e+04	4.14e+02	4.21e+06	2.38e+09	8.75e+06	2.40e+09	
	Median	1.37e+04	4.17e+02	4.70e+06	2.95e+09	9.96e+06	3.52e+09	
	Worst	1.42e+04	4.18e+02	5.06e+06	3.97e+09	1.10e+07	4.80e+09	
	Mean	1.37e+04	4.16e+02	4.65e+06	3.07e+09	1.00e+07	3.57e+09	
	Std	2.66e+02	7.76e-01	2.37e+05	4.01e+08	5.88e+05	5.43e+08	
FEs=6.0e+05	Best	1.14e+04	4.09e+02	1.27e+06	6.11e+03	2.91e+06	3.33e+03	
	Median	1.21e+04	4.13e+02	1.38e+06	1.80e+04	3.15e+06	3.99e+03	
	Worst	1.24e+04	4.14e+02	1.48e+06	3.24e+04	3.40e+06	4.56e+03	
	Mean	1.20e+04	4.13e+02	1.38e+06	1.90e+04	3.17e+06	4.02e+03	
	Std	3.04e+02	1.21e+00	5.84e+04	6.80e+03	1.44e+05	2.94e+02	
FEs=3.0e+06	Best	5.15e+03	3.75e+02	1.92e+05	1.34e+03	4.07e+05	1.11e+03	
	Median	5.64e+03	3.85e+02	2.13e+05	1.65e+03	4.43e+05	1.38e+03	
	Worst	6.34e+03	4.00e+02	2.43e+05	2.24e+03	4.93e+05	1.60e+03	
	Mean	5.63e+03	3.87e+02	2.14e+05	1.68e+03	4.42e+05	1.38e+03	
	Std	2.81e+02	5.24e+00	1.47e+04	2.09e+02	1.85e+04	1.16e+02	

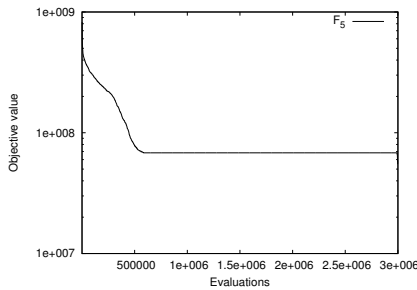


Fig. 7. The graph of  $F_5$

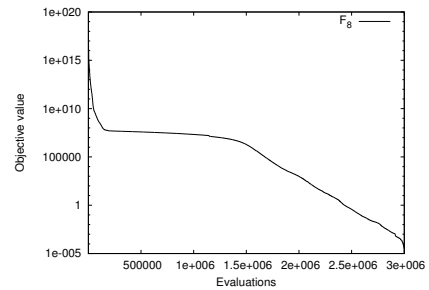


Fig. 8. The graph of  $F_8$

attained the best median results among all methods in 16 problems  $F_4$ – $F_{16}$  and  $F_{18}$ – $F_{20}$ .

- MLCC attained the best average results among all meth-

ods in 5 problems  $F_1$ – $F_3$ ,  $F_{16}$  and  $F_{17}$ . MLCC attained the best median results among all methods in 4 problems  $F_1$ – $F_3$  and  $F_{17}$ .  $F_1$ – $F_3$  are separable functions.

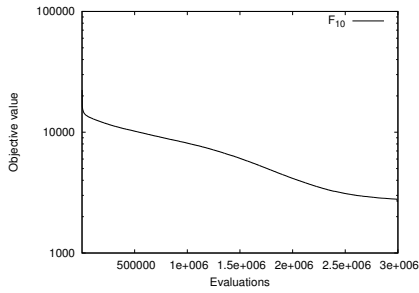


Fig. 9. The graph of  $F_{10}$

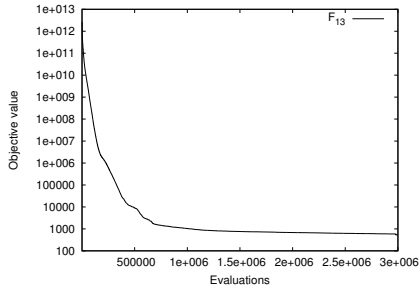


Fig. 10. The graph of  $F_{14}$

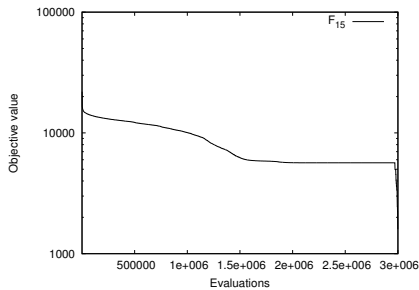


Fig. 11. The graph of  $F_{15}$

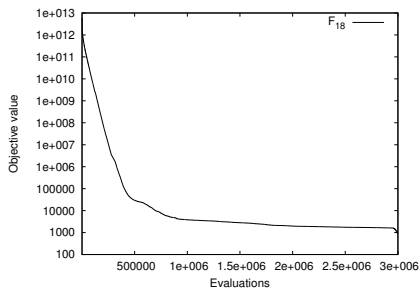


Fig. 12. The graph of  $F_{18}$

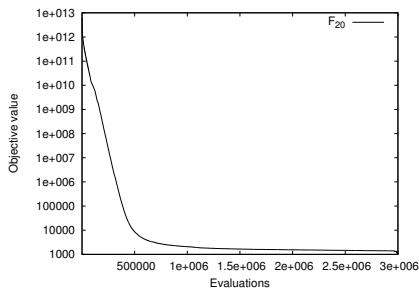


Fig. 13. The graph of  $F_{20}$

- The great differences of results in  $F_7$  and  $F_8$  between LMDEa and the others are very clear.  $F_7$  and  $F_8$  are single-group  $m$ -nonseparable functions.

Thus, it is thought that LMDEa is effective to various problems.

## VI. CONCLUSION

Differential evolution is known as a simple, efficient and robust search algorithm that can solve nonlinear optimization problems. In this study, we proposed LMDEa algorithm. LMDEa utilize a diversity archive to improve the diversity of search when small population for large scale optimization problems is used. Also, LMDEa utilized the landscape modality detection to select a proper value of the scaling factor. It was shown that LMDEa can improve the search efficiency compared with DECC-CG and MLCC in 15 problems on average out of 20 problems. Thus, it is thought that LMDEa is a very efficient optimization algorithm compared with other methods.

In the future, we will design more dynamic control of parameters, especially the crossover rate, for LMDEa.

## ACKNOWLEDGMENT

This research is supported in part by Grant-in-Aid for Scientific Research (C) (No. 20500138,22510166) of Japan society for the promotion of science and Hiroshima City University Grant for Special Academic Research (General Studies).

## REFERENCES

- [1] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
- [2] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.
- [3] U. K. Chakraborty, Ed., *Advances in Differential Evolution*. Springer, 2008.
- [4] S. Das and P. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [5] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
- [6] T. Takahama and S. Sakai, "Constrained optimization by the  $\varepsilon$  constrained differential evolution with an archive and gradient-based mutation," in *Proc. of the 2010 IEEE Congress on Evolutionary Computation*, Jul. 2010, pp. 1680–1688.
- [7] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large scale global optimization," Nature Inspired Computation and Applications Laboratory, USTC, China, Tech. Rep., 2009. [Online]. Available: <http://nical.ustc.edu.cn/cec10ss.php>
- [8] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput.*, vol. 9, no. 6, pp. 448–462, 2005.
- [9] T. Takahama and S. Sakai, "Fuzzy c-means clustering and partition entropy for species-best strategy and search mode selection in nonlinear optimization by differential evolution," in *Proc. of the 2011 IEEE International Conference on Fuzzy Systems*, Jun. 2011, pp. 290–297.
- [10] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Comput.*, vol. 10, no. 8, pp. 673–686, 2006.
- [11] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1785–1791.

TABLE II  
EXPERIMENTAL RESULTS ON LMDEA AND OTHER METHODS. BEST, MEDIAN, WORST, MEAN VALUES AND STANDARD DEVIATIONS IN 25 RUNS ARE SHOWN

		$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$
DECC-CG	Best	1.63e-07	1.25e+03	1.20e+00	7.78e+12	1.50e+08	3.89e+06	4.26e+07
	Median	2.86e-07	1.31e+03	1.39e+00	1.51e+13	2.38e+08	4.80e+06	1.07e+08
	Worst	4.84e-07	1.40e+03	1.68e+00	2.65e+13	4.12e+08	7.73e+06	6.23e+08
	Mean	2.93e-07	1.31e+03	1.39e+00	1.70e+13	2.63e+08	4.96e+06	1.63e+08
	Std	8.62e-08	3.26e+01	9.73e-02	5.37e+12	8.44e+07	8.02e+05	1.37e+08
MLCC	Best	<b>0.00e+00</b>	<b>1.73e-11</b>	1.28e-13	4.27e+12	2.15e+08	5.85e+06	4.16e+04
	Median	<b>0.00e+00</b>	<b>6.43e-11</b>	<b>1.46e-13</b>	1.03e+13	3.92e+08	1.95e+07	5.15e+05
	Worst	<b>3.83e-26</b>	<b>1.09e+01</b>	<b>1.86e-11</b>	1.62e+13	4.87e+08	1.98e+07	2.78e+06
	Mean	<b>1.53e-27</b>	<b>5.57e-01</b>	<b>9.88e-13</b>	9.61e+12	3.84e+08	1.62e+07	6.89e+05
	Std	<b>7.66e-27</b>	<b>2.21e+00</b>	<b>3.70e-12</b>	3.43e+12	6.93e+07	4.97e+06	7.37e+05
LMDEa	Best	2.42e-24	5.31e+02	<b>7.86e-14</b>	<b>1.14e+11</b>	<b>3.68e+07</b>	<b>4.00e-09</b>	<b>3.79e-02</b>
	Median	7.36e-24	6.87e+02	8.79e-01	<b>2.06e+11</b>	<b>6.07e+07</b>	<b>4.02e-09</b>	<b>1.91e-01</b>
	Worst	1.55e-22	8.62e+02	1.09e+00	<b>3.15e+11</b>	<b>1.13e+08</b>	<b>1.01e+00</b>	<b>6.42e-01</b>
	Mean	1.35e-23	6.97e+02	6.44e-01	<b>2.08e+11</b>	<b>6.62e+07</b>	<b>2.63e-01</b>	<b>2.45e-01</b>
	Std	2.91e-23	8.22e+01	4.46e-01	<b>5.89e+10</b>	<b>2.02e+07</b>	<b>4.22e-01</b>	<b>1.68e-01</b>
		$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$
DEC-CG	Best	6.37e+06	2.66e+08	1.03e+04	2.06e+01	7.78e+04	1.78e+03	6.96e+08
	Median	6.70e+07	3.18e+08	1.07e+04	2.33e+01	8.87e+04	3.00e+03	8.07e+08
	Worst	9.22e+07	3.87e+08	1.17e+04	<b>2.79e+01</b>	1.07e+05	1.66e+04	9.06e+08
	Mean	6.44e+07	3.21e+08	1.06e+04	2.34e+01	8.93e+04	5.12e+03	8.08e+08
	Std	2.89e+07	3.38e+07	2.95e+02	<b>1.78e+00</b>	6.87e+03	3.95e+03	6.07e+07
MLCC	Best	4.51e+04	8.96e+07	2.52e+03	1.96e+02	2.42e+04	1.01e+03	2.62e+08
	Median	4.67e+07	1.24e+08	3.16e+03	1.98e+02	3.47e+04	1.91e+03	3.16e+08
	Worst	9.06e+07	1.46e+08	5.90e+03	1.98e+02	4.25e+04	3.47e+03	3.77e+08
	Mean	4.38e+07	1.23e+08	3.43e+03	1.98e+02	3.49e+04	2.08e+03	3.16e+08
	Std	3.45e+07	1.33e+07	8.72e+02	6.98e-01	4.92e+03	7.27e+02	2.77e+07
LMDEa	Best	<b>7.80e-05</b>	<b>2.25e+07</b>	<b>2.45e+03</b>	<b>6.49e-11</b>	<b>1.35e+04</b>	<b>4.56e+02</b>	<b>7.65e+07</b>
	Median	<b>3.29e-04</b>	<b>2.65e+07</b>	<b>2.76e+03</b>	<b>2.74e+00</b>	<b>1.81e+04</b>	<b>5.64e+02</b>	<b>8.67e+07</b>
	Worst	<b>9.20e-04</b>	<b>2.98e+07</b>	<b>3.97e+03</b>	5.40e+01	<b>2.49e+04</b>	<b>9.38e+02</b>	<b>1.03e+08</b>
	Mean	<b>3.61e-04</b>	<b>2.64e+07</b>	<b>2.80e+03</b>	<b>1.19e+01</b>	<b>1.83e+04</b>	<b>5.95e+02</b>	<b>8.63e+07</b>
	Std	<b>2.33e-04</b>	<b>1.89e+06</b>	<b>2.84e+02</b>	1.50e+01	<b>2.62e+03</b>	<b>1.06e+02</b>	<b>6.30e+06</b>
		$F_{15}$	$F_{16}$	$F_{17}$	$F_{18}$	$F_{19}$	$F_{20}$	
DECC-CG	Best	1.09e+04	5.97e+01	2.50e+05	5.61e+03	1.02e+06	3.59e+03	
	Median	1.18e+04	7.51e+01	2.89e+05	2.30e+04	1.11e+06	3.98e+03	
	Worst	1.39e+04	9.24e+01	3.26e+05	4.71e+04	1.20e+06	5.32e+03	
	Mean	1.22e+04	7.66e+01	2.87e+05	2.46e+04	1.11e+06	4.06e+03	
	Std	8.97e+02	8.14e+00	1.98e+04	1.05e+04	5.15e+04	3.66e+02	
MLCC	Best	5.30e+03	<b>2.08e+02</b>	<b>1.38e+05</b>	2.51e+03	1.21e+06	1.70e+03	
	Median	6.89e+03	3.95e+02	<b>1.59e+05</b>	4.17e+03	1.36e+06	2.04e+03	
	Worst	1.04e+04	<b>3.97e+02</b>	<b>1.86e+05</b>	1.62e+04	1.54e+06	2.34e+03	
	Mean	7.11e+03	<b>3.76e+02</b>	<b>1.59e+05</b>	7.09e+03	1.36e+06	2.05e+03	
	Std	1.34e+03	<b>4.71e+01</b>	<b>1.43e+04</b>	4.77e+03	7.35e+04	1.80e+02	
LMDEa	Best	<b>5.15e+03</b>	3.75e+02	1.92e+05	<b>1.34e+03</b>	<b>4.07e+05</b>	<b>1.11e+03</b>	
	Median	<b>5.64e+03</b>	<b>3.85e+02</b>	2.13e+05	<b>1.65e+03</b>	<b>4.43e+05</b>	<b>1.38e+03</b>	
	Worst	<b>6.34e+03</b>	4.00e+02	2.43e+05	<b>2.24e+03</b>	<b>4.93e+05</b>	<b>1.60e+03</b>	
	Mean	<b>5.63e+03</b>	3.87e+02	2.14e+05	<b>1.68e+03</b>	<b>4.42e+05</b>	<b>1.38e+03</b>	
	Std	<b>2.81e+02</b>	<b>5.24e+00</b>	1.47e+04	<b>2.09e+02</b>	<b>1.85e+04</b>	<b>1.16e+02</b>	

- [12] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transaction on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [13] T. Takahama and S. Sakai, "Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*, Jul. 2006, pp. 308–315.
- [14] S. Kukkonen and J. Lampinen, "Constrained real-parameter optimization with generalized differential evolution," in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*. Vancouver, BC, Canada: IEEE Press, 16–21 July 2006, pp. 207–214.
- [15] T. Takahama and S. Sakai, "Solving difficult constrained optimization problems by the  $\epsilon$  constrained differential evolution with gradient-based mutation," in *Constraint-Handling in Evolutionary Optimization*, E. Mezura-Montes, Ed. Springer-Verlag, 2009, pp. 51–72.
- [16] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.
- [17] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. of the 2008 IEEE Congress on Evolutionary Computation*, 2008, pp. 1663–1670.