Proceedings of the 55th ISCIE International Symposium
on Stochastic Systems Theory and Its Applications
Adachi, Tokyo, Japan, Nov. 17-18, 2023

Adaptive Parameter Control Using Search State Estimation and Extreme Individuals for Differential Evolution

Tetsuyuki Takahama[1] and Setsuko Sakai[2]

[1]Department of Intelligent Systems, Hiroshima City University
3-4-1, Ozuka-Higashi, Asaminami-ku, Hiroshima, 731-3194 Japan
[2]Faculty of Commercial Sciences, Hiroshima Shudo University
1-1-1, Ozuka-Higashi, Asaminami-ku, Hiroshima, 731-3195 Japan
E-mail: takahama@hiroshima-cu.ac.jp, setuko@shudo-u.ac.jp

Abstract

The performance of differential evolution (DE) is influenced by various factors such as algorithm parameters and mutation strategies. JADE (adaptive DE with optional external archive) is recognized as one of the most successful studies on parameter control in DE. In this study, in order to improve the search efficiency of JADE, we propose to integrate two methods we have proposed. One is the method to estimate whether the population of candidate solutions is converging or moving for particle swarm optimization. The normalized distance between the population center and the best solution is used for the estimation. When the popu tion is converging, the search mode is adjusted to hance the convergence. When the population is movi the search mode is adjusted to enhance the moveme The other is the method to control DE parameters extreme individuals in order to improve the search ficiency. The methods are modified to use the for method in JADE and to reduce the interference tween the two methods. The effectiveness of JADE corporating the proposed methods is shown by solv thirteen benchmark problems.

1 Introduction

Optimization problems are very important and quently appear in the real world. Many studies h been conducted for solving optimization problems ing population-based optimization algorithms (POAs) [1] such as evolutionary algorithms (EAs) [2], differential evolution (DE) [3], and particle swarm optimization (PSO) [4]. POAs explore for an optimal solution by employing a population comprising multiple candidate solutions. Typically, they utilize stochastic operations to generate new candidate solutions.

The effectiveness of DE is influenced by various factors such as algorithm parameters and mutation strategies. JADE (adaptive DE with optional external archive) [5] is recognized as one of the most successful studies in parameter control for DE. In this study, to enhance the search efficiency of JADE, we propose to improve parameter control in JADE by integrating two methods that we have proposed as follows:

- A search state estimation method have proposed for PSO [6] to estimate whether the population is converging or moving. The search state is determined by the distance between the population center and the best solution normalized to the interval [0,1]. If the best solution is closest to the center, the normalized distance is 0, and if the best solution is farthest from the center, the distance is 1. If the normalized distance is small, the

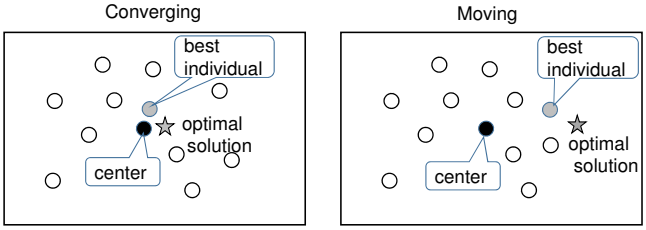

Fig. 1: The position of the best solution relative to the population center in converging and moving states.

- The extreme individuals method have proposed for JADE [7]. In the extreme individuals method, the parameter values generated by JADE are modified to accelerate the convergence of good individuals and realize global search by bad individuals.

In the search state estimation method, estimation thresholds are modified to work well in JADE. When the population is converging, a parameter called scaling factor is adjusted to enhance the convergence. When the population is moving, the parameter is adjusted to enhance the movement. The extreme individual method is simplified to reduce the interference

of the two methods where the parameters are modified only for the best individual and the worst individual. The advantage of JADE with the proposed method is demonstrated by solving thirteen benchmark problems.

Optimization problems and JADE are explained in Section 2. In Section 3, related works including parameter control and search state estimation are briefly reviewed. JADE with the search state estimation method and the extreme individual method is proposed in Section 4. In Section 5, experimental results on some problems are shown. Finally, conclusions are described in Section 6.

2 Optimization by Differential Evolution

2.1 Optimization Problems

This study will address the optimization problem with lower and upper bound constraints as described below.

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & l_j \leq x_j \leq u_j,\ j = 1, \ldots, D, \end{array} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_D)$ is a D dimensional vector and $f(\boldsymbol{x})$ is an objective function. Values l_j and u_j are the lower bound and the upper bound of the variable x_j, respectively. The search space, where each point satisfies the lower and upper bound constraints, is denoted by $\mathcal{S}$.

2.2 Differential Evolution

DE, which is proposed by Storn and Price [3], is one of evolutionary algorithms. DE has been shown to be very effective in optimizing various types of functions. These include non-linear, non-differentiable, non-convex and multimodal functions. It has been shown that DE is fast and robust to these functions [8].

A typical DE algorithm called DE/rand/1/bin is as follows:

1) An initial population $P = \{\boldsymbol{x}_i, i = 1, 2, \cdots, N\}$ is generated randomly in the search space, where N is the population size and P denotes the population.

2) The algorithm is terminated when a predefined condition is satisfied.

3) Each individual $\boldsymbol{x}_i$ is selected as a parent. A mutant vector $\boldsymbol{m}_i$ is generated as follows:

$$\boldsymbol{m}_i = \boldsymbol{x}_{r1} + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{2}$$

where three integers $r1$, $r2$ and $r3$ are chosen randomly in $[1, N]$ without overlapping i and each other. F stands for a scaling factor. A trial vector (child) $\boldsymbol{v}_i$ is generated from $\boldsymbol{x}_i$ and $\boldsymbol{m}_i$ using the binomial crossover as follows:

$$v_{ij} = \begin{cases} m_{ij} & \text{if } j = j_{rand} \text{ or } u(0,1) < CR \\ x_{ij} & otherwise \end{cases} \tag{3}$$

where j_{rand} is a randomly selected integer in $[1, D]$ and $u(0, 1)$ is a uniformly distributed random number in $[0, 1]$. CR stands for a crossover rate. If the child outperforms the parent, the child survives. Otherwise the parent survives. Return to 3) until all individuals are selected as parents.

4) P is replaced by the survivors. Return to step 2).

2.3 JADE

In JADE, the mean values μ_F and μ_{CR} for the scaling factor and the crossover rate are adaptively learned, respectively. Initially, μ_F=μ_{CR}=0.5. The scaling factor F_i and the crossover rate CR_i for each individual $\boldsymbol{x}_i$ are independently generated according to the mean values as follows:

$$F_i \sim C(\mu_F, \sigma_F) \tag{4}$$

$$CR_i \sim N(\mu_{CR}, \sigma^2_{CR}) \tag{5}$$

where F_i is generated according to a Cauchy distribution with location μ_F and scale σ_F=0.1. F_i is truncated to be 1 if F_i is greater than 1 and is regenerated if F_i is zero or negative. CR_i is generated according to a normal distribution with mean μ_{CR} and standard deviation σ_{CR}=0.1. CR_i is truncated to $[0, 1]$. The mean values are updated using parameter values in success cases, when a better child than the parent is generated, as follows:

$$\mu_F = (1 - c)\mu_F + cS_{F^2}/S_F \tag{6}$$

$$\mu_{CR} = (1 - c)\mu_{CR} + cS_{CR}/S_N \tag{7}$$

where S_N is the number of success cases, S_F, S_{F^2} and S_{CR} are the sum of F, F^2 and CR in success cases, respectively. A constant c serves as a weight of update in (0,1] and the recommended value is 0.1.

JADE adopts a strategy called "current-to-pbest". When an external archive is adopted, a mutant vector is generated by current-to-pbest with an archive as follows:

$$\boldsymbol{m}_i = \boldsymbol{x}_i + F_i(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F_i(\boldsymbol{x}_{r2} - \widetilde{\boldsymbol{x}}_{r3}) \tag{8}$$

where $\boldsymbol{x}_{pbest}$ is selected randomly from the top $100p\%$ individuals, $r2$ is a randomly selected integer in $[1, N]$ excluding i, and $\widetilde{\boldsymbol{x}}_{r3}$ is selected randomly from the union of the current population and the archive so that $r3 \notin \{i, r2\}$. In many cases, the archive size N_A is same as N. The archive is initially empty. Parents defeated by the children are then added to the archive. If the number of individuals in the archive exceeds the archive size, randomly selected individuals are removed to maintain the archive size.

To satisfy the bound constraints, a child that falls outside the search space $\mathcal{S}$ is relocated inside $\mathcal{S}$. In JADE, each out-of-bounds element of the child is adjusted to be the midpoint between the corresponding boundary and the corresponding element of the parent, as follows:

$$v_{ij} = \begin{cases} \frac{1}{2}(l_j + x_{ij}) & \text{if } v_{ij} < l_j \\ \frac{1}{2}(u_j + x_{ij}) & \text{if } v_{ij} > u_j \end{cases} \quad (9)$$

This operation is applied after a child is generated through JADE operations.

3 Related Works

3.1 Studies on Algorithm Parameters

The performance of DE is influenced by algorithm parameters such as F, CR, and N, as well as mutation strategies. Methods for controlling algorithm parameters can be categorized as follows:

1) selection-based control: Parameter values and strategies are selected irrespective of current search state such as CoDE [9].

2) observation-based control: Proper parameter values are inferred based on the observed current search state, and parameters and/or strategies are dynamically controlled such as FADE [10], DESFC [11], and LMDE [12, 13].

3) success-based control: The parameters and/or strategies are adjusted to favor the values observed in success cases such as SaDE [14], JADE [5], SHADE [15], CADE [16] and jSO [17]. The self-adaptation, where parameters are contained in individuals and are evolved through the application of evolutionary operators to the parameters, is included in this category such as DESAP [18] and jDE [19].

4) hybrid control: Multiple control methods from different categories are combined to adjust algorithm parameters such as ADEGL [20] and JADEadm [21]. Our proposed method is included in this category.

3.2 Studies on Search State Estimation

Search state estimation is closely related to controlling the balance between exploitation and exploration [22]. When the population diversity is high, such as in the initial population, the search mode is set to exploitation and the vicinity of promising solutions are searched. When the diversity is low, the mode is set to exploration and a new area is searched. For example, in [23], the normalized distance from the center of gravity of the population is adopted as the diversity measure. If the measure exceeds the threshold d_{high}, the search mode is switched to exploitation. If the measure falls below d_{low}, the mode is switched to exploration.

Various measures have been proposed as diversity measures. The distance-based diversity measures are as follows [24], where $||\boldsymbol{x} - \boldsymbol{y}||$ is the distance between the two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and the Euclidean distance is often used.

- Diameter: Maximum distance in the population $\max_{i \neq j \in [1,N]} ||\boldsymbol{x}_i - \boldsymbol{x}_j||$.

- Radius: Maximum distance from the center of gravity of the population $\max_{i \neq j \in [1,N]} ||\boldsymbol{x}_i - \boldsymbol{g}||$, $\boldsymbol{g} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$.

- Normalized average distance: Average distance from the center of gravity of the population normalized by the size of the search space $\frac{1}{LN} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{g}||$. L is the size of the search space $L = ||\boldsymbol{u} - \boldsymbol{l}||$.

However, even if normalized, the threshold is influenced by the population size and so on. It is still difficult to select proper thresholds.

The goal of search state estimation is not merely to observe diversity but rather to estimate the search state of the population. In [25], the state is classified into four states: Convergence, Exploitation, Exploration, and Jumping-out. The positional relationship between the best individual and other individuals is employed for the classification. It is assumed that the best individual is closest to other individuals when the population is converging, and the best individual is farthest from other individuals when the population is moving, and the following measure is proposed.

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \quad (10)$$

$$DBR = \frac{d_{best} - d_{\min}}{d_{\max} - d_{\min}} \quad (11)$$

where d_i is the average distance between $\boldsymbol{x}_i$ and other individuals, d_{best} is the average distance between the best individual and other individuals, and $d_{\min}$ and $d_{\max}$ are the minimum and maximum values of d_i. If DBR is close to 0, the state is classified as Convergence, and if DBR is close to 1, the state is classified as Jumping-out. The time complexity is $O(N^2 D)$ because the distances between all individuals are calculated. Since N is usually set to a multiple of D, $O(N^2 D) = O(N^3)$. It is thought that a measure with low computational complexity is necessary.

In [26], the following measure IOS (Indicator of the Optimization State) is proposed to probabilistically classify the search state into two states: Exploitation and Exploration based on the distances to the best in-

dividual and the function values.

$$IOS = \sum_{i=1}^{N} ||f_i - d_i|| \quad (12)$$

$$\overline{IOS} = \frac{IOS - IOS_{\min}}{IOS_{\max} - IOS_{\min}} \quad (13)$$

where d_i is the rank of the distance between the individual $\boldsymbol{x}_i$ and the best individual with a smaller rank indicating smaller distance. f_i is the rank of the function value with a smaller rank indicating a smaller value. $IOS_{\min}$ and $IOS_{\max}$ are the minimum and maximum values of IOS, respectively. The state is classified as Exploration with the probability of IOS and as Exploitation with the probability of $1-IOS$. In a scenario where the function value deteriorates as the distance from the optimal solution increases, IOS is small because d_i and f_i are almost the same whether the best solution is close to or far from the optimal solution. However, it is suitable to estimate Converging when the best solution is near the optimal solution, and Jumping-out when it is far away.

4 Proposed Method

In this section, the modified search state estimation method and the modified extreme individuals method are proposed.

4.1 Normalized Distance Between the Center and the Best Solution

The center of the population $\boldsymbol{c}$ can be defined as follows:

$$\boldsymbol{c} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \quad (14)$$

Let the Euclidean distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ be denoted by $d(\boldsymbol{x}, \boldsymbol{y})$. The distance between the center $\boldsymbol{c}$ and each solution $\boldsymbol{x}_i$ is given by $d_i = d(\boldsymbol{c}, \boldsymbol{x}_i)$. The normalized distance between the center and the best solution, DCB [6] is defined as follows:

$$DCB = \frac{d_{best} - d_{\min}}{d_{\max} - d_{\min}} \quad (15)$$

where d_{best} is the distance between the center and the best solution, and $d_{\min}$ and $d_{\max}$ are the minimum and maximum values of d_i.

To avoid abrupt changes in DCB, the exponentially smoothed moving average with the smoothing constant 0.5 is employed. The time complexity is $O(ND) = O(N^2)$ because the distance between the center and each solution is calculated.

4.2 Parameter Control Using Search State Estimation

Based on DCB, the search state was classified into strongly Converging, Converging and Moving in PSO. In preliminary experiments, the benchmark functions are optimized and changes of DCB are observed using JADE. As a result, the search efficiency can be improved by classifying into Converging and Moving and controlling F based on the classification as follows:

- $DCB \in$ [0,0.05): The state is classified as "Converging", and the value of F_i is multiplied by 0.98 to speed up the convergence.
- $DCB \in$ [0.4,1.0]: The state is classified as "Moving", and the value of F_i is multiplied by 1.04. Also, the movement vector of the population center from previous generation to current generation $\boldsymbol{c}(t) - \boldsymbol{c}(t-1)$ is added to the generated child. As the result, the population diversity is kept and the moving speed is accelerated.

4.3 Parameter Control for Extreme Individuals

In the extreme individuals method, a population is divided into 3 parts, top individuals (local search applied), bottom individuals (global search applied), and other individuals (search by JADE applied). In this study, the population is divided into the best individual, worst individual, and other individuals in order to simplify the method and to reduce the interference with parameter control using the search state estimation. The following parameter control is adopted:

- Exploitation by the best individual: Local search can be realized by adopting small F and large CR. Therefore, a smaller value than μ_F and a larger value than μ_{CR} are generated for F_i and CR_i, respectively, where μ_F and μ_{CR} are means in JADE defined by Eqs. (6) and (7).

$$F_i \sim u(0.2, \mu_F) \quad (16)$$

$$CR_i \sim u(\mu_{CR}, 1) \quad (17)$$

  where $u(l, u)$ is a uniform random number in $[l, u]$.

- Exploration by the worst individuals: Global search can be realized by adopting large F and random CR. Therefore, a larger value than μ_F and a random value are generated for F_i and CR_i, respectively.

$$F_i \sim u(\mu_F, 1) \quad (18)$$

$$CR_i \sim u(0, 1) \quad (19)$$

4.4 Proposed Algorithm

The proposed algorithm "JADE with search state estimation using DCB and extreme individuals" (JADEdcb+ex) is defined as follows:

1. Initialization: Each individual $\boldsymbol{x}_i$ is randomly generated in the search space where x_{ij} is a uniform random number in $[l_j, u_j]$. The archive A is made empty.

2. The algorithm is stopped when a termination condition is satisfied: The terminal condition in this study is when the number of function evaluations reaches FE_{max}.

3. Initialization for each generation including search state estimation: The list of success cases S is made empty. The population center $\boldsymbol{c}(t)$ is obtained, DCB is calculated and the search state is estimated.

4. JADE operation with parameter control using the estimated state: For each individual $\boldsymbol{x}_i$, F_i and CR_i are generated according to Eq.(4) and Eq.(5), respectively. F_i and CR_i are modified when the estimated state is Converging or Moving. JADE operation is executed according to Eq.(8) and the mutant vector $\boldsymbol{m}_i$ is generated. A new child is generated from $\boldsymbol{x}_i$ and $\boldsymbol{m}_i$ using binomial crossover. When the state is Moving, $\boldsymbol{c}(t) - \boldsymbol{c}(t-1)$ is added to the child.

5. Survivor selection: If the child is better than the parent, the child becomes a survivor. The successful combination of parameter values (F_i, CR_i) is added to S. The defeated parent is added to A. Otherwise, the parent $\boldsymbol{x}_i$ becomes a survivor. Return to 4. until all individuals are processed.

6. Resizing the archive and learning means: When the size of the archive exceeds N_A, randomly selected elements in the archive are deleted until the size becomes N_A. The means μ_F and μ_{CR} are updated using S according to Eqs.(6) and (7). Return to 2.

Fig. 2 shows the proposed algorithm named JADEdcb+ex.

```
JADEdcb+ex()
{
  μ_F=μ_CR=0.5; σ_F=σ_CR=0.1; N_A=N;
// Initialization
  P=N individuals generated randomly in S;
  FE=N; A=∅;
  for(t=1; FE < FE_max; t++) {
    S=∅;
+   c(t) is obtained according to Eq.(14);
+   DCB is obtained according to Eq.(15);
+   The search state is estimated;
    Indexes I = {I_i} is sorted according to f(x_i);
    for(i=1; i ≤ N; i++) {
// JADE operation
+     if(i==I_1) { // best individual
+       F_i=u(0.2, μ_F); CR_i=u(μ_CR, 1);
+     }
+     else if(i==I_N) { // worst individual
+       F_i=u(μ_F, 1); CR_i=u(0, 1);
+     }
+     else {
        CR_i = μ_CR + N(0, σ²_CR);
        truncate CR_i to [0,1];
        do {
          F_i=μ_F + C(0, σ_F);
        } while(F_i ≤ 0);
        if(F_i > 1) F_i = 1;
+       if(The state is Converging or Moving)
+         F_i and CR_i are modified;
        m_i=generated according to Eq.(8);
        v_i=generated from x_i and m_i by crossover;
+       if(The state is Moving) v_i=v_i+c(t)-c(t-1);
+     }
      FE=FE+1;
// Survivor selection
      if(f(v_i) < f(x_i)) {
        z_i=v_i;
        S=S ∪ {(F_i, CR_i)}; // add a success case
        A=A ∪ {x_i};
      }
      else z_i = x_i;
    }
    P = {z_i};
// Resizing the archive
    while(|A|>N_A)
      remove a randomly selected element from A;
// Learning means
    if(|S| > 0) {
      μ_F = (1 − c)μ_F + c Σ_{F_i∈S} F_i² / Σ_{F_i∈S} F_i;
      μ_CR = (1 − c)μ_CR + c Σ_{CR_i∈S} CR_i/|S|;
    }
  }
}
```

Fig. 2: Algorithm of JADEdcb+ex

5 Solving Optimization Problems

In this study, well-known thirteen benchmark problems are solved.

5.1 Test Problems and Experimental Conditions

The 13 scalable benchmark functions are shown in Table 1 [5]. All functions have an optimal value 0. Some characteristics are briefly summarized as follows: Functions f_1 to f_4 are continuous unimodal functions. The function f_5 has ridge landscape, f_6 is a discontinuous step function, and f_7 is a noisy quartic function. Functions f_8 to f_{13} are multimodal functions.

Experimental conditions are same as JADE as follows: Population size $N = 100$, the pbest parameter $p = 0.05$, and the learning parameter $c = 0.1$. Independent 50 runs are performed for 13 problems. The number of dimensions for the problems is 30 ($D = 30$). Each run stops when the number of function evaluations exceeds the maximum number of evaluations $FE_{\max}$. In each function, different $FE_{\max}$ is adopted.

5.2 Experimental Results

In the experiment, JADE, JADE with parameter control using the search state estimation only (JADE+DCB), JADE with the extreme individual method only (JADE+extreme), and the proposed method (JADEdcb+ex). Table 2 shows the experimen-

Table 1: Test functions of dimension D. These are sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, step, noisy quartic, Schwefel 2.26, Rastrigin, Ackley, Griewank, and two penalized functions, respectively[27].

Test functions	Search space
$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
$f_2(\boldsymbol{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$
$f_3(\boldsymbol{x}) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^D$
$f_4(\boldsymbol{x}) = \max_i \{ \lvert x_i \rvert \}$	$[-100, 100]^D$
$f_5(\boldsymbol{x}) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]^D$
$f_6(\boldsymbol{x}) = \sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor^2$	$[-100, 100]^D$
$f_7(\boldsymbol{x}) = \sum_{i=1}^{D} i x_i^4 + rand[0, 1)$	$[-1.28, 1.28]^D$
$f_8(\boldsymbol{x}) = \sum_{i=1}^{D} -x_i \sin \sqrt{\lvert x_i \rvert}$ $+ D \cdot 418.98288727243369$	$[-500, 500]^D$
$f_9(\boldsymbol{x}) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	$[-5.12, 5.12]^D$
$f_{10}(\boldsymbol{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right)$ $- \exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$
$f_{12}(x) = \frac{\pi}{D} [10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2$ $\{1 + 10 \sin^2(\pi y_{i+1})\} + (y_D - 1)^2]$ $+ \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) =$ $\begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$
$f_{13}(x) = 0.1[\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2$ $\{1 + \sin^2(3\pi x_{i+1})\} + (x_D - 1)^2$ $\{1 + \sin^2(2\pi x_D)\}] + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$

tal result. The mean value and standard deviation of best objective values over 50 runs are shown in the top row for each function. The median value is shown in the bottom row for each function. Because a small number of failure trials were observed, median values are compared rather than mean values between algorithms. Also, Wilcoxon signed rank test is conducted, and the results for each function are displayed adjacent to the median value in parentheses. Symbols '+', '−', and '=' indicate that each algorithm is significantly better than JADE, significantly worse than JADE, and not significantly different from JADE, respectively. Symbols '++' and '−−' denote a significance level of 1%, while '+' and '−' indicate a significance level of 5%.

The result of the experiment is first evaluated by the median values. JADEdcb+ex attained the best median results in 9 functions f_1, f_2, f_5, f_6, f_8 and $f_{10} - f_{13}$. JADE+DCB attained the best median results in 4 functions f_3, f_4, f_6 and f_7. JADE+extreme attained the best result in function f_9. Therefore, it is thought that the proposed method succeeded to strengthen the convergence and the movement of the individuals.

JADEdcb+ex attained significantly better results than JADE in 12 functions excluding f_7 and attained no significantly worse result. Since the function f_7 is a noisy function, it is thought that the noise makes it difficult to determine whether the state is Converging. JADE+DCB attained significantly better results than JADE in 10 functions excluding f_5, f_7 and f_9, and attained significantly worse result in f_9. The JADE+extreme attained significantly better results than JADE in 7 functions and no significantly worse result. It is thought that JADEdcb+ex is the best algorithm followed by JADE+DCB, JADE+extreme and JADE.

Fig. 3 shows how μ_F is controlled based on the search state estimated using DCB in f_8. The vertical axis represents μ_F (left) and DCB (right), and the horizontal axis represents the number of function evaluations. The function f_8 has the optimal solution near the edge of the search space. In early generations, Moving state is detected and μ_F becomes larger than that in JADE to move toward the optimal solution faster. In later generations, Converging state is detected and μ_F becomes smaller than that in JADE to speed up the convergence. It is thought that the search state estimation and the control of μ_F is appropriate in f_8.

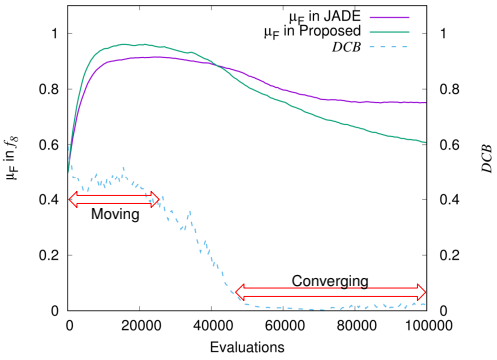

Fig. 3: Control of μ_F based on DCB

6 Conclusions

In this study, we proposed a new search state estimation method using DCB which is the normalized distance between the population center and the best solution in the population and which has low time complexity. If DCB is small, the population is estimated to be Converging. If DCB is large, the population is estimated to be Moving. In order to improve the performance of JADE, which is a representative adaptive DE, we proposed the modified control of F_i according to the estimated search state. If the state is Converging, F_i is decreased to speed up the convergence. If the state is Moving, F_i is increased and the movement vector of the population center is added the the child to strengthen the movement. Also, simplified extreme individual method is introduced where the best indi-

Table 2: Experimental result on JADE and the proposed methods

func.	$FE_{\max}$	JADE	JADE+DCB	JADE+extreme	proposed
f_1	150000	2.18e-57 ± 1.2e-56	8.62e-62 ± 5.9e-61	1.42e-60 ± 9.0e-60	1.02e-65 ± 4.9e-65
		3.6019e-63	6.0304e-67 (++)	1.5273e-64 (++)	**1.0633e-68** (++)
f_2	200000	1.64e-25 ± 7.5e-25	1.01e-27 ± 6.5e-27	2.71e-38 ± 1.2e-37	4.41e-42 ± 1.0e-41
		1.3790e-35	2.5203e-40 (++)	1.4409e-40 (++)	**2.9647e-43** (++)
f_3	500000	4.20e-83 ± 2.3e-82	2.01e-85 ± 7.0e-85	1.33e-83 ± 6.9e-83	6.38e-84 ± 2.5e-83
		1.5392e-86	**4.0832e-88** (+)	7.9984e-87 (=)	6.9726e-88 (++)
f_4	500000	1.29e-62 ± 3.2e-62	3.92e-64 ± 2.4e-63	6.64e-63 ± 1.8e-62	5.68e-65 ± 1.5e-64
		3.5103e-64	**2.4886e-66** (++)	1.3677e-64 (+)	2.5811e-66 (++)
f_5	300000	7.97e-02 ± 5.6e-01	1.59e-01 ± 7.8e-01	1.42e-18 ± 5.7e-18	1.59e-01 ± 7.8e-01
		2.1842e-20	6.0227e-21 (=)	1.5050e-20 (=)	**1.5137e-21** (++)
f_6	10000	6.04e+00 ± 1.9e+00	3.94e+00 ± 1.4e+00	5.78e+00 ± 1.8e+00	4.32e+00 ± 1.6e+00
		3.0000e+00	**2.0000e+00** (++)	3.0000e+00 (=)	**2.0000e+00** (++)
f_7	300000	6.25e-04 ± 2.3e-04	5.81e-04 ± 2.5e-04	6.63e-04 ± 2.3e-04	6.81e-04 ± 2.2e-04
		2.9515e-04	**2.8437e-04** (=)	3.0475e-04 (=)	3.1435e-04 (=)
f_8	100000	7.11e+00 ± 2.8e+01	4.74e+00 ± 2.3e+01	2.37e+00 ± 1.7e+01	4.74e+00 ± 2.3e+01
		1.9616e-05	4.6628e-06 (++)	2.8411e-06 (++)	**2.4069e-07** (++)
f_9	100000	1.86e-04 ± 8.9e-05	2.78e-04 ± 1.8e-04	1.22e-04 ± 6.5e-05	1.49e-04 ± 1.5e-04
		8.0951e-05	1.1614e-04 (−−)	**5.0422e-05** (++)	5.6175e-05 (++)
f_{10}	50000	3.65e-09 ± 7.1e-09	8.47e-10 ± 6.4e-10	2.21e-09 ± 1.6e-09	5.59e-10 ± 3.4e-10
		1.0813e-09	2.8472e-10 (++)	8.2447e-10 (=)	**2.4260e-10** (++)
f_{11}	50000	3.77e-08 ± 1.8e-07	2.18e-06 ± 1.5e-05	3.62e-11 ± 8.2e-11	1.19e-06 ± 8.3e-06
		3.4603e-12	7.4690e-13 (++)	2.5610e-12 (=)	**2.0700e-13** (++)
f_{12}	50000	1.57e-16 ± 3.8e-16	4.41e-17 ± 2.4e-16	4.86e-17 ± 7.6e-17	2.13e-18 ± 4.9e-18
		2.8900e-17	1.2963e-18 (++)	5.8706e-18 (+)	**4.5530e-19** (++)
f_{13}	50000	1.25e-15 ± 2.1e-15	8.90e-17 ± 3.8e-16	2.59e-16 ± 9.6e-16	1.06e-17 ± 1.7e-17
		2.0153e-16	6.4041e-18 (++)	2.8884e-17 (++)	**2.0565e-18** (++)
+			10	7	12
=			2	6	1
−			1	0	0

vidual makes a local search and the worst individual makes a global search. The numerical experiment for 13 functions showed that the proposed method worked well in 12 functions and attained the better median results than JADE in 12 functions.

In the future, we will apply the proposed method to other population-based optimizers.

Acknowledgment

This study is supported by JSPS KAKENHI Grant Numbers Grant Numbers 19K04916 and 20K11977.

References

[1] A. E. Babalola, B. A. Ojokoh, and J. B. Odili, "A review of population-based optimization algorithms," in *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, 2020, pp. 1–7.

[2] A. E. Eiben and J. E. Smith, *What Is an Evolutionary Algorithm?* Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 25–48.

[3] 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.

[4] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco: Morgan Kaufmann, 2001.

[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] S. Sakai and T. Takahama, "A study on converging and moving detection using distance between the center and the best solution for particle swarm optimization," in *Economic History, Flow of Funds, Information Systems and Operations Research*, K.Ota, J.Maeda, and A.Nushimoto, Eds. Kyushu University Press, Mar. 2023, pp. 69–89.

[7] T. Takahama and S. Sakai, "An adaptive differential evolution with exploitation and exploration

by extreme individuals," in *Proc. of SICE Annual Conference 2017*, Sep. 2017, pp. 1147–1152.

[8] U. K. Chakraborty, Ed., *Advances in Differential Evolution*. Springer, 2008.

[9] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[10] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[11] 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.

[12] T. Takahama and S. Sakai, "Differential evolution with dynamic strategy and parameter selection by detecting landscape modality," in *Proc. of the 2012 IEEE Congress on Evolutionary Computation*, Jun. 2012, pp. 2114–2121.

[13] T. Takahama and S. Sakai, "Large scale optimization by differential evolution with landscape modality detection and a diversity archive," in *Proc. of the 2012 IEEE Congress on Evolutionary Computation*, Jun. 2012, pp. 2842–2849.

[14] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[15] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. of the 2013 IEEE Congress on Evolutionary Computation*, 2013, pp. 71–78.

[16] T. Takahama and S. Sakai, "An adaptive differential evolution considering correlation of two algorithm parameters," in *Proc. of the Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems (SCIS&ISIS2014)*, Dec. 2014, pp. 618–623.

[17] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jso," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 1311–1318.

[18] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Computing*, vol. 10, no. 8, pp. 673–686, Jun. 2006.

[19] 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, Dec. 2006.

[20] T. Takahama and S. Sakai, "An adaptive differential evolution with learning parameters according to groups defined by the rank of objective values," in *Proc. of the Eighth International Conference on Swarm Intelligence (ICSI2017)*, Jul. 2017, pp. 411–419.

[21] T. Takahama and S. Sakai, "Adaptive directional mutation for an adaptive differential evolution algorithm," in *Proc. of Joint 11th International Conference on Soft Computing and Intelligent Systems and 21th International Symposium on Advanced Intelligent Systems*, Dec. 2020, pp. 262–268.

[22] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM computing surveys (CSUR)*, vol. 45, no. 3, pp. 1–33, 2013.

[23] R. K. Ursem, "Diversity-guided evolutionary algorithms," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2002, pp. 462–471.

[24] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1128–1134.

[25] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.

[26] W.-J. Yu, M. Shen, W. neng Chen, Z. hui Zhan, Y.-J. Gong, Y. Lin, O. Liu, and J. Zhang, "Differential evolution with two-level parameter adaptation," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1080–1099, July 2014.

[27] X. Yao, Y. Liu, K.-H. Liang, and G. Lin, "Fast evolutionary algorithms," in *Advances in Evolutionary Computing: Theory and Applications*, A. Ghosh and S. Tsutsui, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 45–94.