A New Binomial Crossover Considering Correlation Among Decision Variables for Adaptive Differential Evolution

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

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

***Abstract*—In population-based optimization methods such as evolutionary algorithms, various information can be obtained from the distribution of good search points. When problems with strong dependency among decision variables are optimized, a characteristic distribution, which is a thin elliptical distribution, may appear. In order to generate good children, it is necessary to change the variables simultaneously along the long axis of the elliptical distribution. A similar distribution also may appear when the search points are far from the optimal solution even in problems with independent variables. In this study, we propose a new crossover CBX which uses correlation coefficients of search points in order to detect such distribution and realizes efficient movement toward the optimal solution. The crossover points are decided so that highly correlated variables are inherited at the same time. However, if only CBX is used, the diversity of the search points tends to be lost rapidly. The adaptive control of the probability for applying CBX is also proposed. The advantage of the proposed method is shown by solving several benchmark problems.**

***Index Terms*—differential evolution; correlating binomial crossover; correlation coefficient; adaptive 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).

In population-based optimization methods such as EAs, various information can be obtained from the distribution of good search points. Problems with strong dependency among decision variables are typical difficult optimization problems. When such problems are optimized, a characteristic distribution, which is a thin elliptical distribution as shown in Fig.1, may appear. In order to generate good children in this case, it is necessary to change the variables simultaneously along the long axis of the elliptical distribution so as to approach the optimal solution. A similar distribution also may appear when the search points are far from the optimal solution even in problems with independent variables. It is considered that the search points can be move toward the optimal solution efficiently by simultaneously changing the variables.

In this study, we propose to use correlation coefficients of search points in order to detect such distribution. The correlation coefficient matrix between decision variables can be obtained from the search points. The crossover points are decided so that highly correlated variables are inherited at the same time. It is expected that the problems with strong dependency among variables can be efficiently solved. Also, search points far from the optimal solution can be efficiently moved toward the optimal solution.

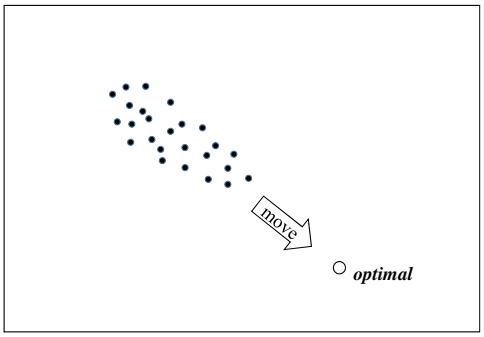

Fig. 1. The search points moving toward the optimal solution

The binomial crossover (BX) is a crossover operation used in differential evolution. In BX, a gene, which is inherited to the child unconditionally, are randomly selected, and other genes are inherited with the probability of the crossover rate. In this study, we propose CBX (correlating binomial crossover) which is a binomial crossover operation using correlation coefficients. In CBX, the crossed position where a gene was inherited to the child and the non-crossed position where a gene was not inherited to the child are stored. When judging whether a gene will be inherited or not, if the correlation coefficient between the crossed position and the position of the gene is high, the gene is inherited. If the correlation coefficient between the non-crossed position and the position of the gene is high, the gene is not inherited. Otherwise, the gene is inherited with the probability of the crossover rate.

However, if only CBX is used as a crossover operation, the diversity of the search points tends to be lost rapidly. In order to keep the diversity, the probability of applying CBX is adaptively adjusted in this study. Success cases, where the child is better than the parent, are observed. The success rates of CBX and BX are compared. The probability of the crossover operation which has higher success rate is increased.

Differential evolution (DE), which is an EA proposed by

Storn and Price [1], has been successfully applied to optimization problems including non-linear, non-differentiable, non-convex and multimodal functions [2], [3]. It has been shown that DE is a very fast and robust algorithm. The performance of DE is affected by algorithm parameters such as a scaling factor F, a crossover rate CR and so on. Many studies have been done to control the parameters and the strategies. One of the most successful studies on controlling the parameters is JADE(adaptive DE with optional external archive) [4]. In this study, CBX and adaptive adjustment of the probability of applying CBX is introduced to JADE. The advantage of the proposed method is shown by solving thirteen benchmark problems.

In Section II, related works are described. DE and JADE are briefly explained in Section III. In Section IV, CBX and its adaptive control is proposed. The experimental results are shown in Section V. Finally, conclusions are described in Section VI.

II. Related Works

A. Parameter Control

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 controlling algorithm parameters can be classified into some categories as follows:

(1) selection-based control: Strategies and parameter values are selected regardless of current search state. CoDE (composite DE) [5] generates three trial vectors using three strategies with randomly selected parameter values from parameter candidate sets and the best trial vector will head to the survivor selection.

(2) observation-based control: The current search state is observed, proper parameter values are inferred according to the observation, and parameters and/or strategies are dynamically controlled. FADE (Fuzzy Adaptive DE) [6] 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) [7] adopts fuzzy clustering, observes partition entropy of search points, and controls CR and the mutation strategies between the rand and the species-best strategy. LMDE (DE with detecting Landscape Modality) [8] detects the landscape modality such as unimodal or multimodal using the change of the objective values at sampling points which are equally spaced along a line. If the landscape is unimodal, greedy parameter settings for local search are selected. Otherwise, parameter settings for global search are selected.

(3) 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 (DE with Self-Adapting Populations) [9] controls F, CR and N self-adaptively. SaDE (Self-adaptive DE) [10] controls the selection probability of the mutation strategies according to the success rates and controls the mean value of CR for each strategy according to the mean value in success case. jDE (self-adaptive DE algorithm) [11] controls F and CR self-adaptively. JADE (adaptive DE with optional external archive) [4] and MDE_pBX (modified DE with p-best crossover) [12] control the mean or power mean values of F and CR according to the mean values in success cases. CADE (Correlation-based Adaptive DE) [13] introduces the correlation of F and CR to JADE.

In the category (1), useful knowledge to improve the search efficiency is ignored. In the category (2), it is difficult to select proper type of observation which is independent of the optimization problem and its scale. In the category (3), 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 to improve JADE in the category (3) by introducing CBX. It is thought that CBX is a kind of observation-based control in the category (2) because CBX modifies BX by observing the correlation coefficients. Thus, the proposed method is a hybrid method of the category (2) and (3). It is expected that CBX improves efficiency in problems with ridge landscape.

CBX is partly similar to CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [14] because CMA-ES uses a covariance matrix and CBX uses a correlation matrix.

B. Linkage Identification

Identifying the dependency among variables is called linkage identification, and is very important issue in search process. There are some studies for linkage identification: In [15], LINC(Linkage Identification by Nonlinearity Check) is proposed for genetic algorithm. In [16], learning of linkage matrix, of which elements indicate the strength of the linkage between the i-th variable and the j-th variables is proposed for particle swarm optimization. In [17], learning of linkage matrix, which is different from [16], is proposed for differential evolution.

LINC is explained because these studies adopted a similar idea. In order to obtain the strength of linkage between the i-th variable and the j-th variable, the change of function value when only the i-th variable is perturbed $\triangle f_i$, that when only the j-th variable is perturbed $\triangle f_j$ and that when both variables are perturbed $\triangle f_{ij} (i < j)$ are obtained as follows:

$$\triangle f_i = f(\cdots, x_i', \cdots, x_j, \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (1)$$

$$\triangle f_j = f(\cdots, x_i, \cdots, x_j', \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (2)$$

$$\triangle f_{ij} = f(\cdots, x_i', \cdots, x_j', \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (3)$$

If the i-th variable and the j-th variable are independent, the following is satisfied.

$$\triangle f_{ij} \quad = \quad \triangle f_i + \triangle f_j \tag{4}$$

Conversely, if this condition is not satisfied, it is thought that there is a linkage between the i-th variable and the j-th variable. The strength of the linkage e_{ij} can be defined as follows:

$$e_{ij} \quad = \quad |\triangle f_{ij} - (\triangle f_i + \triangle f_j)| \tag{5}$$

LINC needs $D+1$ function evaluations (FEs) for calculating Eq.(1) and Eq.(2), $\frac{1}{2}D(D-1)$ FEs for calculating Eq.(3), and $\frac{1}{2}D(D+1)+1$=$O(D^2)$ FEs in total. If the computing cost of the objective function is high, it is very difficult to identify the linkage many times. However, there are many problems of which landscape is very different in macroscopic view and microscopic view. In such case, it is difficult to use this type of linkage identification.

On the other hand, a correlation matrix is used in this study and does not require extra function evaluations. Therefore, the proposed method can be applied to such problems.

III. Optimization by Differential Evolution

A. Optimization Problems

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

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & l_j \leq x_j \leq u_j,\ j = 1, \ldots, D, \end{array} \tag{6}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_D)$ is a D dimensional vector and $f(\boldsymbol{x})$ is an objective function. The function f is a nonlinear real-valued function. Values l_j and u_j are the lower bound and the upper bound of x_j, respectively. The search space is the region that satisfies the lower and upper bound constraints.

B. Differential Evolution

In DE, initial individuals are randomly generated within given search space and form an initial population of size N. Each individual $\boldsymbol{x}_i, i = 1, 2, \cdots, N$ contains D genes as decision variables. At each generation, 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 child, or 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 $\boldsymbol{x}_i$, three individuals $\boldsymbol{x}_{p1}$, $\boldsymbol{x}_{p2}$ and $\boldsymbol{x}_{p3}$ are chosen randomly from the population without overlapping $\boldsymbol{x}_i$ and each other. A new vector, or a mutant vector $\boldsymbol{m}_i$ is generated by the base vector $\boldsymbol{x}_{p1}$ and the difference vector $\boldsymbol{x}_{p2} - \boldsymbol{x}_{p3}$, where F is the scaling factor.

$$\boldsymbol{m}_i = \boldsymbol{x}_{p1} + F(\boldsymbol{x}_{p2} - \boldsymbol{x}_{p3}) \tag{7}$$

"$cross$" 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.

A child, or a trial vector $\boldsymbol{x}_i^{\text{child}}$is generated by the binomial crossover as follows:

$$x_{ij}^{\text{child}} = \begin{cases} m_{ij}, & \text{if } u(0,1) < CR \text{ or } j = j_{rand} \\ x_{ij}, & \text{otherwise} \end{cases} \tag{8}$$

where CR is a crossover rate, $u(0,1)$ is a uniform random number generator in $[0,1]$, and j_{rand} is a random integer in $[1, D]$.

C. JADE

In JADE, the mean value of the scaling factor μ_F and the mean value of the crossover rate μ_{CR} are learned to define two probability density functions, where initial values are 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 two functions as follows:

$$F_i \quad \sim \quad C(\mu_F, \sigma_F) \tag{9}$$

$$CR_i \quad \sim \quad N(\mu_{CR}, \sigma_{CR}^2) \tag{10}$$

where F_i is a random variable according to a Cauchy distribution $C(\mu_F, \sigma_F)$ with a location parameter μ_F and a scale parameter σ_F=0.1. CR_i is a random variable according to a normal distribution $N(\mu_{CR}, \sigma_{CR}^2)$ of a mean μ_{CR} and a standard deviation σ_{CR}=0.1. CR_i is truncated to $[0,1]$ and F_i is truncated to be 1 if $F_i > 1$ or regenerated if $F_i \leq 0$. The location μ_F and the mean μ_{CR} are updated as follows:

$$\mu_F \quad = \quad (1-c)\mu_F + cS_{F^2}/S_F \tag{11}$$

$$\mu_{CR} \quad = \quad (1-c)\mu_{CR} + cS_{CR}/S_N \tag{12}$$

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 is a weight of update in (0,1] and the recommended value is 0.1.

JADE adopts a strategy called "current-to-pbest" where an intermediate point between a parent $\boldsymbol{x}_i$ and a randomly selected point from top individuals is used as a base vector.

A mutation vector is generated by current-to-pbest without archive as follows:

$$\boldsymbol{m}_i = \boldsymbol{x}_i + F_i(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F_i(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \quad (13)$$

where $\boldsymbol{x}_{pbest}$ is a randomly selected individual from the top $100p\%$ individuals.

In order to satisfy bound constraints, a child that is outside of the search space is moved into the inside of the search space. In JADE, each outside element of the child is set to be the middle between the corresponding boundary and the element of the parent as follows:

$$x_{ij}^{\text{child}} = \begin{cases} \frac{1}{2}(l_j + x_{ij}) & (x_{ij}^{\text{child}} < l_j) \\ \frac{1}{2}(u_j + x_{ij}) & (x_{ij}^{\text{child}} > u_j) \end{cases} \quad (14)$$

This operation is applied when a new point is generated by JADE operations.

IV. Proposed Method

A. Correlating Binomial Crossover (CBX)

In the usual binomial crossover, the probability that genes of the mutant vector are inherited to the child is specified by the crossover probability CR. All genes are inherited to the child with the same probability. However, in problems where dependency between variables is strong, it is difficult to generate a good child unless genes with high dependency are inherited simultaneously. Therefore, in this study, we propose a new crossover CBX that can inherit genes with large correlation coefficients simultaneously based on the correlation coefficients between variables.

A correlation coefficient is an index for measuring the correlation between two variables. A correlation matrix is defined by extending this to multiple variables and is composed of the correlation coefficients. Let a population be denoted by $\{\boldsymbol{x}_i | \, \boldsymbol{x}_i = (x_{ij}), j = 1, 2, \cdots, D, i = 1, 2, \cdots, N\}$, where D is the dimension of the problem and N is the number of individuals. The correlation matrix $R = (r_{kj})$, where r_{kj} is the correlation coefficient between the k-th variable (x_k) and the j-th variable (x_j), can be defined as follows:

$$r_{kj} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_{ik} - \bar{x}_k)(x_{ij} - \bar{x}_j)}{\sigma_k \sigma_j} \quad (15)$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)^2}, \; \bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \quad (16)$$

If the correlation coefficient is 1, there is strong positive correlation. If the correlation coefficient is -1, there is strong negative correlation. From the viewpoint of gene inheritance, it is thought that there is strong dependency in both cases. Therefore, the absolute value of the correlation coefficient is used for measuring the strength of the dependency as follows:

$$\rho_{kj} = |r_{kj}| \quad (17)$$

```
CBX(j_rand, x, m)
{
  k0=-1;      // index of the non-crossed gene
  k1=j_rand;  // index of the crossed gene
  j=j_rand%D+1;
  for(l=1; l≤D; l++) {
    if(j == j_rand) cross=1;
    else if(ρ_{k1,j} > ρ̄ + S_r σ_ρ) cross=1;
    else if(k0!=-1 && ρ_{k0,j} > ρ̄ + S_r σ_ρ) cross=0;
    else if(u(0,1) < CR_i) cross=1; else cross=0;
    if(cross) { x_j^child=m_j; k1=j; }
    else      { x_j^child=x_j; k0=j; }
    j=j%D+1;
  }
  return x^child;
}
```

Fig. 2. The algorithm of CBX

In this study, the average of ρ_{kj}, or $\bar{\rho}$ and standard deviation of ρ_{kj}, or σ_ρ are used to judge whether the dependency is strong or not.

$$\bar{\rho} = \frac{2}{D(D-1)}\sum_{k=1}^{D}\sum_{j<k}\rho_{kj} \quad (18)$$

$$\sigma_\rho = \sqrt{\frac{2}{D(D-1)}\sum_{k=1}^{D}\sum_{j<k}(\rho_{kj} - \bar{\rho})^2} \quad (19)$$

A new algorithm parameter S_r is introduced for the judgment. If the following condition is satisfied, it is judged that the dependency is strong:

$$\rho_{kj} > \bar{\rho} + S_r \sigma_\rho \quad (20)$$

The algorithm of CBX is shown in Fig.2. j_{rand} is a randomly selected position where the gene is crossed, that is, inherited from the mutant vector unconditionally. The crossed position k_1 where a gene was inherited and the non-crossed position k_0 where a gene was not inherited are stored. Each j-th gene is checked from the next position of j_{rand} to j_{rand}. If the condition $\rho_{k_1,j} > \bar{\rho} + S_r\sigma_\rho$ is satisfied, the dependency between the crossed gene and the j-th gene is strong and the j-th gene is inherited from the mutant vector $\boldsymbol{m}_i$ to the child. If the condition $\rho_{k_0,j} > \bar{\rho} + S_r\sigma_\rho$ is satisfied, the dependency between the non-crossed gene and the j-th gene is strong and the j-th gene is not inherited from $\boldsymbol{m}_i$ but from the parent $\boldsymbol{x}_i$. Otherwise, the gene is inherited with the probability of crossover rate CR.

It is thought that proper means μ_F and μ_{CR} for CBX and for BX are different. Thus, the means for CBX and the means for BX is separately learned by JADE method. μ_F^k and μ_{CR}^k are introduced where k=1 for CBX and k=0 for BX, which are similar to the group learning in [18].

B. Adaptive Control of the Rate of CBX

In CBX, the diversity of the search points tends to be lost rapidly. In this study, the probability of applying CBX is adaptively controlled. Let the probability of CBX be denoted by R_{CBX}. The initial value of R_{CBX} is 0.5 and the range of

R_{CBX} is in [0.05,0.95] in order to give opportunities to both of CBX and BX. Success cases, where the child is better than the parent, are observed. The success rate of CBX (s^1) and that of BX (s^0) are compared. if s^1 is greater than s^0, R_{CBX} is increased. In the opposite case, R_{CBX} is decreased.

C. Algorithm

The algorithm of the proposed method ADECBX (Adaptive DE with CBX) can be described as follows:

Step 0 Parameter setup. The mean values of scaling factor μ_F^k=0.5 and the mean values of crossover rate μ_{CR}^k=0.5, $k = 0, 1$, where $k = 0$ for BX and $k = 1$ for CBX. The scale parameter σ_F=0.1 and the standard deviation σ_{CR}=0.1. The probability of using CBX R_{CBX}=0.5.

Step 1 Initialization of the individuals. N individuals $\{\boldsymbol{x}_i | i = 1, 2, \cdots, N\}$ are generated randomly in the search space and form an initial population.

Step 2 Termination condition. If the number of function evaluations exceeds the maximum number of evaluations $FE_{\max}$, the algorithm is terminated.

Step 3 Initialization for each generation. The list of success cases S^k is made empty ($k = 0, 1$). The number of trials for BX and CBX, or m^k is initialized ($k = 0, 1$).

Step 4 DE operation with adaptive parameters. CBX or BX is selected according to R_{CBX}. K is set to 0 in case of BX and is set to 1 in case of CBX. m^K is incremented by 1. The scaling factor F_i is generated according to Cauchy distribution using μ_F^K. The crossover rate CR_i is generated according to the normal distribution using μ_{CR}^K. CBX or BX is applied and a new child is generated.

Step 5 Survivor selection. If the child is better than the parent, the operation is treated as a success case and the child becomes a survivor. The successful pair of parameter values (F_i, CR_i) is added to success cases S^K. Otherwise, the parent $\boldsymbol{x}_i$ becomes a survivor. Go back to Step 4 until all individuals are processed.

Step 6 Learning of parameters. The means of the scaling factor μ_F^k and the means of crossover rate μ_{CR}^k are updated using S^k ($k = 0, 1$) according to Eqs. (11) and (12). Success rates of CBX and BX are obtained as $|S^k|/m^k$, where $|\cdot|$ is the number of elements. When the success rate of CBX is greater than that of BX, R_{CBX} is increased. In the opposite case, R_{CBX} is decreased.

Step 7 Go back to Step 2.

Fig. 3 shows the pseudo-code of the proposed method.

V. Numerical Experiments

In this paper, 13 well-known benchmark problems are solved.

```
JADE/current-to-pbest/1/adaptive CBX+BX()
{
+ μ_F^k=μ_CR^k=0.5, k=0,1;
  σ_F = σ_CR=0.1;
+ R_CBX=0.5; △R=0.01;
// Initialize a population
  P=N individuals generated randomly in the search space;
  FE=N;
  for(t=1; FE < FE_max; t++) {
+   (r_kj)=Correlation matrix of P is obtained by Eq.(15);
+   (ρ_kj) is obtained by Eq.(17);
+   ρ̄ and σ_ρ are obtained by Eqs.(18) and (19);
+   S^k=∅, k=0,1;
+   m^k=0, k=0,1;   // number of trials for BX and CBX
    for(i=1; i ≤ N; i++) {
+     if(u(0,1) < R_CBX) K=1;  // CBX
+     else K=0;                // BX
+     m^K++;
+     CR_i = μ_CR^K + N(0, σ_CR^2);
      if(CR_i < 0) CR_i=0;
      else if(CR_i > 1) CR_i=1;
      do {
+       F_i=μ_F^K + C(0, σ_F);
      } while(F_i ≤ 0);
      if(F_i > 1) F_i = 1;
      x_pbest = Randomly selected from top 100p% in P;
      x_r1 = Randomly selected from P(r1 ∉ {i});
      x_r2 = Randomly selected from P(r2 ∉ {i, r1});
      m_i = x_i+F_i(x_pbest − x_i)+F_i(x_r1 − x_r2);
      j_rand=randint(1,D);
+     if(K==1)
// correlating binomial crossover
+       x_i^child=CBX(j_rand, x_i, m_i);
+     else
// binomial crossover
        x_i^child=binomial crossover between x_i and m_i;
// Survivor selection
      if(f(x_i^child) < f(x_i)) {
        z_i = x_i^child;
        S^K = S^K ∪ {(F_i, CR_i)}; // a success case is added
      }
      else z_i = x_i;
      FE++;
    }
    P = {z_i};
+   for(k=0; k < 2; k++)
+     if(|S^k| > 0) {
+       μ_F^k = (1 − c)μ_F^k + c∑_{F_i∈S^k} F_i^2/∑_{F_i∈S^k} F_i;
+       μ_CR^k = (1 − c)μ_CR^k + c∑_{CR_i∈S^k} CR_i/|S^k|;
+     }
+   if(m^0 > 0 && m^1 > 0) {
+     s^0 = |S^0|/m^0; // success rate of BX
+     s^1 = |S^1|/m^1; // success rate of CBX
+     if(s^1 > s^0)
+       R_CBX = R_CBX + △R;
+     else if(s^0 > s^1)
+       R_CBX = R_CBX − △R;
+     if(R_CBX > 0.95) R_CBX=0.95;
+     else if(R_CBX < 0.05) R_CBX=0.05;
+   }
  }
}
```

Fig. 3. The algorithm of proposed method

A. Test Problems and Experimental Conditions

The 13 scalable benchmark functions are shown in Table I [4]. Every function has an optimal objective value 0. Some characteristics are briefly summarized as follows: Functions f_1 to f_4 are continuous unimodal functions. The function f_5 is Rosenbrock function which is unimodal for 2- and 3-dimensions but may have multiple minima in high dimension cases [19]. The function f_6 is a discontinuous step function, and f_7 is a noisy quartic function. Functions f_8 to f_{13} are multimodal functions and the number of their local minima increases exponentially with the problem dimension [20].

TABLE I
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.

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

Experimental conditions are same as JADE as follows: Population size $N = 100$, initial mean for scaling factor $\mu_F = 0.5$ and initial mean for crossover rate $\mu_{CR} = 0.5$, the pbest parameter p=0.05, and the learning parameter c=0.1. S_r are selected from $\{0.0, 0.3, 0.6, 1\}$ and $\triangle R$=0.01 for ADECBX.

50 independent 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.

B. Experimental Results

Table II shows the experimental results on JADE and ADECBX in case of S_r=0, 0.3, 0.6 and 1. The results of JADE can be obtained by ADECBX with fixing R_{CBX}=0. The mean value and the standard deviation of best objective values in 50 runs are shown for each function. The maximum number of function evaluations is selected for each function and is shown in column labeled $FE_{\max}$. The best result among algorithms is highlighted using bold face fonts. Also, Wilcoxon signed rank test [21] is performed and the result for each function is shown under the mean value. Symbols '+', '$-$' and '=' are shown when ADECBX is significantly better than JADE, is significantly worse than JADE, and is not significantly different from JADE, respectively. Note that the symbols '+' and '$-$' denote that the significance level is 5% and the symbols '++' and '$--$' denote that the significance level is 1%.

ADECBX (Sr=0.6) attained significantly better results than JADE in 11 functions except for f_4 and f_7. ADECBX (Sr=0 and 0.3) attained significantly better results than JADE in 10 functions except for f_4, f_7 and f_{11}. ADECBX (Sr=1) attained significantly better results than JADE in 9 functions except for f_4, f_7, f_{11} and f_{13}. JADE attained significantly better results than ADECBX (S_r=0, 0.3) in one function f_7 and ADECBX (S_r=0.6, 1) in no function.

Also, ADECBX (Sr=0) attained best mean results in 6 functions f_1, f_2, f_6, f_9, f_{10} and f_{13} out of 13 functions. ADECBX (Sr=0.6) attained best mean results in 2 functions f_5 and f_{11}. ADECBX (Sr=1) attained best mean results in 2 functions f_3 and f_8. JADE attained best mean results in 2 functions f_4 and f_7. ADECBX (Sr=0.3) attained the best mean result in f_{12}. The average ranks of best mean results are 2.50 (Sr=0.3), 2.69 (Sr=0.6), 2.88 (Sr=0), 3.27 (Sr=1) and 3.65 (JADE).

Thus, it is thought that ADECBX (Sr=0.6) is the best one among 4 methods.

VI. CONCLUSIONS

We proposed a new crossover CBX which uses correlation coefficients of search points in order to identify dependency among decision variables and solve problems with strong dependency. The crossover points are decided so that highly correlated variables are inherited at the same time. Also, adaptive control of the probability of applying CBX is proposed. From numerical experiments, it is shown that ADECBX (Sr=0.6) attained significantly better results compared with JADE in many problems.

In this paper, all search points are used to obtain the correlation matrix. If some good search points are used for the matrix, it is expected that the identification of dependency may become more accurate. Also, a method that groups of variables are extracted based on correlation coefficients and the groups are used for the binomial crossover operation will be considered. It is expected that CBX can be introduced into various DE variants other than JADE in order to improve the performance.

ACKNOWLEDGMENT

This study is supported by JSPS KAKENHI Grant Numbers 26350443 and 17K00311.

TABLE II
EXPERIMENTAL RESULTS

	FE_{max}	JADE	ADECBX ($S_r = 0$)	ADECBX ($S_r = 0.3$)	ADECBX ($S_r = 0.6$)	ADECBX ($S_r = 1$)
f_1	150,000	1.40e-59 ± 9.74e-59	**2.97e-67 ± 1.50e-66**	7.77e-66 ± 4.40e-65	4.81e-64 ± 3.17e-63	1.57e-63 ± 6.67e-63
			++	++	++	++
f_2	200,000	6.81e-25 ± 4.71e-24	**4.07e-45 ± 1.59e-44**	6.35e-43 ± 2.26e-42	3.75e-41 ± 1.99e-40	8.26e-36 ± 5.66e-35
			++	++	++	++
f_3	500,000	1.59e-62 ± 5.33e-62	8.81e-73 ± 2.95e-72	4.73e-79 ± 2.55e-78	1.64e-82 ± 1.14e-81	**6.91e-86 ± 2.87e-85**
			++	++	++	++
f_4	500,000	**9.43e-24 ± 2.71e-23**	4.27e-23 ± 2.25e-22	9.57e-24 ± 3.80e-23	1.99e-23 ± 6.83e-23	2.41e-23 ± 7.65e-23
			=	=	=	=
f_5	150,000	3.19e-01 ± 1.08e+00	5.58e-01 ± 1.38e+00	1.59e-01 ± 7.81e-01	**7.01e-22 ± 4.90e-21**	2.39e-01 ± 9.47e-01
			++	++	++	++
f_6	10,000	3.06e+00 ± 1.05e+00	**7.40e-01 ± 8.90e-01**	1.14e+00 ± 8.00e-01	2.04e+00 ± 1.02e+00	2.50e+00 ± 1.06e+00
			++	++	++	++
f_7	300,000	**6.36e-04 ± 3.08e-04**	7.92e-04 ± 3.22e-04	7.39e-04 ± 2.69e-04	7.16e-04 ± 2.88e-04	7.18e-04 ± 3.35e-04
			−	−	=	=
f_8	100,000	2.37e+00 ± 1.66e+01	2.37e+00 ± 1.66e+01	4.74e+00 ± 2.32e+01	4.74e+00 ± 2.32e+01	**1.13e-08 ± 3.19e-08**
			++	++	++	++
f_9	100,000	9.96e-05 ± 5.06e-05	**2.93e-09 ± 8.59e-09**	2.03e-08 ± 4.07e-08	7.81e-07 ± 3.34e-06	4.67e-06 ± 8.55e-06
			++	++	++	++
f_{10}	50,000	9.90e-10 ± 7.21e-10	**7.10e-11 ± 3.44e-11**	1.54e-10 ± 1.02e-10	3.33e-10 ± 3.52e-10	5.19e-10 ± 3.88e-10
			++	++	++	++
f_{11}	50,000	4.19e-11 ± 2.89e-10	5.43e-04 ± 2.17e-03	4.44e-04 ± 2.19e-03	**2.55e-16 ± 1.59e-15**	1.03e-07 ± 7.19e-07
			=	=	+	=
f_{12}	50,000	1.57e-17 ± 4.10e-17	4.15e-03 ± 2.03e-02	**1.85e-19 ± 5.64e-19**	2.07e-03 ± 1.45e-02	2.07e-03 ± 1.45e-02
			++	++	++	++
f_{13}	50,000	1.83e-16 ± 5.82e-16	**7.62e-20 ± 9.70e-20**	1.37e-18 ± 4.56e-18	3.70e-18 ± 6.18e-18	1.89e-17 ± 7.29e-17
			++	++	++	=
+		—	10	10	11	9
=		—	2	2	2	4
−		—	1	1	0	0

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

[13] 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.

[14] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[15] M. Munetomo and D. E. Goldberg, "A genetic algorithm using linkage identification by nonlinearity check," in *Proc. of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, 1999, pp. 595–600.

[16] D. Devicharan and C. K. Mohan, "Particle swarm optimization with adaptive linkage learning," in *Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 1, June 2004, pp. 530–535.

[17] Y. Cai and J. Wang, "Differential evolution with hybrid linkage crossover," *Information Sciences*, vol. 320, pp. 244–287, Nov. 2015.

[18] 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.

[19] Y.-W. Shang and Y.-H. Qiu, "A note on the extended Rosenbrock function," *Evolutionary Computation*, vol. 14, no. 1, pp. 119–126, Apr. 2006.

[20] X. Yao, Y. Liu, , and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 82–102, Jul. 1999.

[21] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.