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.

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 $p$BX (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 $\Delta f_i$, that when only the $j$-th variable is perturbed $\Delta f_j$ and that when both variables are perturbed $\Delta f_{ij}$($i < j$) are obtained as follows:

\[
\begin{align*}
\Delta f_i & = f(\cdots, x_i', \cdots, x_j, \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (1) \\
\Delta f_j & = f(\cdots, x_i, \cdots, x_j', \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (2) \\
\Delta f_{ij} & = f(\cdots, x_i', \cdots, x_j', \cdots) - f(\cdots, x_i, \cdots, x_j, \cdots) \quad (3)
\end{align*}
\]
If the $i$-th variable and the $j$-th variable are independent, the following is satisfied.

$$
\triangle f_{ij} = \triangle f_i + \triangle f_j
$$

(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:

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

(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.

$$
\text{minimize } f(x)
$$

subject to

$$
l_j \leq x_j \leq u_j, \ j = 1, \ldots, D,
$$

where $x = (x_1, x_2, \ldots, x_D)$ is a $D$ dimensional vector and $f(x)$ 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 $x_i, i = 1, 2, \ldots, N$ 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 recombinated 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 $x_i$, three individuals $x_{p1}$, $x_{p2}$ and $x_{p3}$ are chosen randomly from the population without overlapping $x_i$ and each other. A new vector, or a mutant vector $m_i$ is generated by the base vector $x_{p1}$ and the difference vector $x_{p2} - x_{p3}$, where $F$ is the scaling factor.

$$
m_i = x_{p1} + F(x_{p2} - x_{p3})
$$

(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 $x_{i,j}^{\text{child}}$ is generated by the binomial crossover as follows:

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

(8)

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

C. JADE

In JADE, the mean value of the scaling factor $\mu_F$ and the mean value of the crossover rate $\mu_{CR}$ are learned to define two probability density functions, where initial values are $\mu_F = \mu_{CR} = 0.5$. The scaling factor $F_i$ and the crossover rate $CR_i$ for each individual $x_i$ are independently generated according to the two functions as follows:

$$
F_i \sim C(\mu_F, \sigma_F)
$$

(9)

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

(10)

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

$$
\mu_F = (1 - c)\mu_F + cS_F / S_F
$$

(11)

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

(12)

where $S_N$ is the number of success cases, $S_F$, $S_{F2}$ 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 $x_i$ 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:

\[ m_i = x_i + F_i(x_{pbest} - x_i) + F_i(x_{r2} - x_{r3}) \]  

(13)

where \( x_{pbest} \) is a randomly selected individual from the top 100% individuals.

In order to satisfy bound constraints, a child that is outside 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:

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

(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 dependency 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 \( \{ x_i | x_i = (x_{ij}), j = 1, 2, \cdots, D, i = 1, 2, \cdots, N \} \), 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:

\[ r_{kj} = \frac{1}{N} \sum_{i=1}^{N} (x_{ik} - \bar{x}_k)(x_{ij} - \bar{x}_j) \]  

(15)

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

(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:

\[ \rho_{kj} = |r_{kj}| \]  

(17)

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

\[ \rho_{kj} > \hat{\rho} + S_r \sigma_\rho \]  

(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 \( k_j \) 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 \( \rho_{kj} > \hat{\rho} + S_r \sigma_\rho \) 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 \( m_i \) to the child. If the condition \( \rho_{kj} > \hat{\rho} + S_r \sigma_\rho \) 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 \( m_i \) but from the parent \( x_i \). Otherwise, the gene is inherited with the probability of crossover rate \( CR \).

It is thought that proper means \( \mu_C \) and \( \mu_{CR} \) for CBX and for BX are different. Thus, the means for CBX and the means for BX is separately learned by JADE method. \( \mu_C^k \) and \( \mu_{CR}^k \) 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

\[ \rho_{kj} > \hat{\rho} + S_r \sigma_\rho \]  

(20)
$R_{CBX}$ 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 $\mu_F^k = 0.5$ and the mean values of crossover rate $\mu_{CR}^k = 0.5$, $k = 0, 1$, where $k = 0$ for BX and $k = 1$ for CBX. The scale parameter $\sigma_F = 0.1$ and the standard deviation $\sigma_{CR} = 0.1$. The probability of using CBX $R_{CBX} = 0.5$.

Step 1 Initialization of the individuals. $N$ individuals $\{x_i| i = 1, 2, \cdots, N\}$ 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 $F_{E\text{max}}$, 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 $\mu_F^k$. The crossover rate $CR_i$ is generated according to the normal distribution using $\mu_{CR}^k$. 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 $x_i$ becomes a survivor. Go back to Step 4 until all individuals are processed.

Step 6 Learning of parameters. The means of the scaling factor $\mu_F^k$ and the means of crossover rate $\mu_{CR}^k$ 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.
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].

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 best parameter $p = 0.05$, and the learning parameter $c = 0.1$. $S_r$ are selected from {0.0, 0.3, 0.6, 1} and $\Delta R = 0.01$ 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 $F_{E_{max}}$. In each function, different $F_{E_{max}}$ 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 $R_{CBX} = 0.01$.

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 $F_{E_{max}}$. 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 ($S_r = 0.6$) attained significantly better results than JADE in 11 functions except for $f_4$ and $f_7$. ADECBX ($S_r = 0.3$) attained significantly better results than JADE in 10 functions except for $f_4$, $f_7$ and $f_{11}$. ADECBX ($S_r = 1$) attained significantly better results than JADE in 9 functions except for $f_4$, $f_7$, $f_{11}$ and $f_{13}$. ADECBX 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 ($S_r = 0$) 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 ($S_r = 0.6$) attained best mean results in 2 functions $f_5$ and $f_9$. ADECBX ($S_r = 1$) 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 ($S_r = 0.3$) attained the best mean result in $f_{12}$. The average ranks of best mean results are 2.50 ($S_r = 0.3$), 2.69 ($S_r = 0.6$), 2.88 ($S_r = 0$), 3.27 ($S_r = 1$) and 3.65 (JADE).

Thus, it is thought that ADECBX ($S_r = 0.6$) 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 ($S_r = 0.6$) 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.


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.

