Grouping of Genes According to Correlation Coefficients and Grouping-Based Crossover for Adaptive Differential Evolution

Tetsuyuki Takahama
Department of Intelligent Systems, Hiroshima City University
Asaminami-ku, Hiroshima, 731-3194 Japan

E-mail: takahama@hiroshima-cu.ac.jp

Setsuko Sakai
Faculty of Commercial Sciences, Hiroshima Shudo University
Asaminami-ku, Hiroshima, 731-3195 Japan

E-mail: setuko@shudo-u.ac.jp

Abstract

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 (genes) simultaneously along the long axis of the elliptical distribution. Since binomial crossover in differential evolution determines whether each gene is crossed or not with the same probability, it is difficult to change some genes simultaneously. In this study, we propose a crossover operation GBX which uses correlation coefficients of search points in order to detect such distribution. The highly correlated genes are grouped and the genes in each group are crossed (or not crossed) simultaneously. However, if only GBX is used, the diversity of the search points tends to be lost rapidly. The adaptive control of the probability for applying GBX is also proposed. The advantage of the proposed method is shown by solving several benchmark problems.

1 Introduction

In population-based optimization methods such as evolutionary algorithms (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 correlated or non-separable variables.

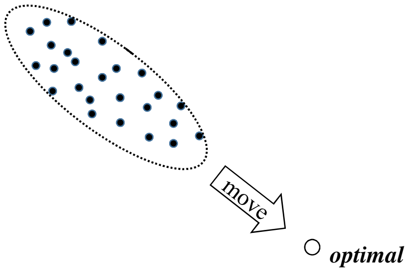

Fig. 1: The search points moving toward the optimal solution

In this study, we propose to use correlation coefficients of search points in order to detect such distribution. The correlation coefficients between decision variables can be obtained from the search points. The highly correlated variables (genes) are grouped and the grouped genes are crossed (or not crossed) simultaneously. 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 (DE), which is an EA proposed by Storn and Price [1] and has been successfully applied to optimization problems including nonlinear, non-differentiable, non-convex and multimodal functions [2, 3, 4]. In BX, a gene, which is crossed unconditionally, is randomly selected, and other genes are crossed with the probability of the crossover rate. In this study, we propose GBX (grouping-based binomial crossover) which is a binomial crossover operation where genes are grouped according to correlation coefficients and each group of genes is crossed or not crossed simultaneously. In GBX, a gene, which is crossed unconditionally, is randomly selected as in BX. The gene and genes which have large correlation coefficients with it are grouped and are crossed. Regarding other genes, each gene and the gene which has the largest correlation coefficient with it are grouped if the genes does not belong to any group yet. The genes in the same group are crossed (or not crossed) with the crossover rate.

However, if only GBX 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 GBX is adaptively adjusted in this study. Success cases, where the child is better than the parent, are observed. The success rates of GBX and BX are compared. The probability of the crossover operation which has higher success rate is increased.

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) [5]. In this study, GBX and adaptive adjustment of the probability of applying GBX 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, GBX and its adaptive control is proposed. The experimental results are shown in Section V. Finally, conclusions are described in Section VI.

2 Related Works

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 [6], LINC(Linkage Identification by Nonlinearity Check) is proposed for genetic algorithm. In [7], 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 [8], learning of linkage matrix, which is different from [7], is proposed for differential evolution.

Only 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 followings are obtained:

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

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

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

where $\triangle f_i$ is the change of function value when only the i-th variable is perturbed, $\triangle f_j$ is the change when only the j-th variable is perturbed and $\triangle f_{ij} (i < j)$ is the change when both variables are perturbed. If the i-th variable and the j-th variable are independent, the following is satisfied.

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

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:

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

LINC needs the number of function evaluations $O(D^2)$. 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.

3 Differential Evolution

3.1 Optimization Problems

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

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

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.

3.2 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

```
binomial crossover DE/·/·/bin
  $j_{rand}$=randint(1,$D$);
  for($k$=1; $k \le D$; $k$++) {
    if($k == j_{rand}$ || $u(0,1) < CR$) $x_k^{\text{child}}$=$m_k$;
    else $x_k^{\text{child}}$=$x_{ik}$;
  }
```

Fig. 2: Binomial crossover operation, where randint(1,D) generates an integer randomly from $[1, D]$ and $u(l, r)$ is a uniform random number generator in $[l, r]$.

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 $\boldsymbol{m}$ 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.

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

"*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. Figure 2 shows the binomial crossover. A new child $\boldsymbol{x}^{\text{child}}$ is generated from the parent $\boldsymbol{x}_i$ and the mutant vector $\boldsymbol{m}$, where CR is a crossover rate.

3.3 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:

$$F_i \sim C(\mu_F, \sigma_F) \tag{8}$$

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2) \tag{9}$$

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 $F_i \le 0$. The location μ_F and the mean μ_{CR} are updated as follows:

$$\mu_F = (1 - c)\mu_F + cS_{F^2}/S_F \tag{10}$$

$$\mu_{CR} = (1 - c)\mu_{CR} + cS_{CR}/S_N \tag{11}$$

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 an archive as follows:

$$\boldsymbol{m} = \boldsymbol{x}_i + F_i(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F_i(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{12}$$

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:

$$x_j^{\text{child}} = \begin{cases} \frac{1}{2}(l_j + x_{ij}) & (x_j^{\text{child}} < l_j) \\ \frac{1}{2}(u_j + x_{ij}) & (x_j^{\text{child}} > u_j) \end{cases} \tag{13}$$

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

4 Proposed method

4.1 Grouping-based binomial crossover (GBX)

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. Each gene is inherited to the child with the same probability. However, in problems where dependency among variables is strong, it is difficult to generate a good child unless genes with strong dependency are inherited simultaneously. Therefore, in this study, we propose a new crossover operation GBX where genes with large correlation coefficients are grouped and the genes in the same group are crossed simultaneously.

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:

$$r_{kj} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_{ik}-\bar{x}_k)(x_{ij}-\bar{x}_j)}{\sigma_k\sigma_j} \tag{14}$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)^2} \tag{15}$$

$$\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N}x_{ij} \tag{16}$$

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:

$$\rho_{kj} = |r_{kj}| \tag{17}$$

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.

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

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

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:

$$\rho_{kj} > \rho^{strong} = \max\{\bar{\rho} + S_r\sigma_\rho, 0.15\} \tag{20}$$

The ρ^{strong} is a threshold value. In order to avoid too small threshold value, the minimum value of ρ^{strong} is defined as 0.15. If S_r=0, the dependency is strong if the coefficient is greater than the average coefficients.

The algorithm of GBX is shown in Fig.3. The groups of genes are formed as follows and the genes in the same group are crossed (or not crossed) simultaneously. As in BX, j_{rand}, where the gene is inherited from the mutant vector unconditionally, is a randomly selected. The genes which have strong correlation coefficients with it are grouped and are crossed. Regarding other genes, each gene and the gene which has the largest correlation coefficient with it are grouped if the genes does not belong to any group yet. The genes in the same group are crossed (or not crossed) with the crossover rate CR_i. The gene, or variable $x_{m_{\rho_k}}$, which has the largest coefficient with each gene, or variable x_k is defined as follows:

$$m_{\rho_k} = \arg\max_{j \neq k}\rho_{kj} \tag{21}$$

```
for($j$=1; $j \leq D$; $j$++) cross[$j$]=-1; // crossover flags
$j_{rand}$=randint(1,$D$);
cross[$j_{rand}$]=1; // unconditionally crossed at $j_{rand}$
$j$=$j_{rand}$;
for($l$=1; $l \leq D$; $l$++) {
  if($cross[j] >= 0$); // already in a group
  else if($\rho_{j_{rand},j} > \rho^{strong}$) cross[$j$]=1;
  else if($u(0,1) < CR_i$) cross[$j$]=1;
  else cross[$j$]=0;
  if(cross[$m_{\rho_j}$]==-1 && $\rho_{j,m_{\rho_j}} > \rho^{strong}$)
    cross[$m_{\rho_j}$]=cross[$j$];
  $j$=($j$+1)%$D$;
}
for($j$=1; $j \leq D$; $j$++) {
  if(cross[$j$]==1) $x_j^{\mathrm{child}}$=$m_j$;
  else $x_j^{\mathrm{child}}$=$x_j$;
}
```

Fig. 3: The algorithm of GBX

It is thought that proper means μ_F and μ_{CR} for GBX and for BX are different. Thus, the means for GBX and the means for BX are separately learned by JADE method. μ_F^k and μ_{CR}^k are introduced where k=1 for GBX and k=0 for BX, which are similar to the group learning in [9].

4.2 Adaptive control of the rate of GBX

In GBX, the diversity of the search points tends to be lost rapidly. In this study, the probability of applying GBX is adaptively controlled. Let the probability of GBX be denoted by R_{GBX}. The initial value of R_{GBX} is 0.5 and the range of R_{GBX} is in [0.05,0.95] in order to give opportunities to both of GBX and BX. Success cases, where the child is better than the parent, are observed. The success rate of GBX (s^1) and that of BX (s^0) are compared. if s^1 is greater than s^0, R_{GBX} is increased. In the opposite case, R_{GBX} is decreased.

4.3 Algorithm

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

Step0 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 GBX. The scale parameter σ_F=0.1 and the standard deviation σ_{CR}=0.1. The probability of using GBX R_{GBX}=0.5.

Step1 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.

Step2 Termination condition. If the number of function evaluations exceeds the maximum number of evaluations $FE_{\max}$, the algorithm is terminated.

Step3 Initialization for each generation. The list of success cases S^k is made empty ($k = 0, 1$). The number of trials for BX and GBX, or n^k is initialized ($k = 0, 1$).

Step4 DE operation with adaptive parameters. GBX or BX is selected according to R_{GBX}. K is set to 0 in case of BX and is set to 1 in case of GBX. 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. GBX or BX is applied and a new child is generated.

Step5 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.

Step6 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. (10) and (11). Success rates of GBX and BX are obtained as $|S^k|/n^k$, where $|\cdot|$ is the number of elements. When the success rate of GBX is greater than that of BX, R_{GBX} is increased. In the opposite case, R_{GBX} is decreased.

Step7 Go back to Step2.

Fig.4 shows the pseudo-code of the proposed method.

5 Numerical Experiments

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

5.1 Test Problems

The 13 scalable benchmark functions are shown in Table 1[5]. 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[10]. 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[11]. Functions f_2, f_3, f_5, f_{10} and f_{11} are non-separable functions.

In order to investigate the performance for non-separable problems, rotated problems are solved. The rotated problems are obtained by rotating the benchmark problems: A candidate solution $\boldsymbol{z}$ is converted as $\boldsymbol{x} = M\boldsymbol{z}$ and $f(\boldsymbol{x})$ is minimized, where M is a rotation matrix. In this study, Helmert matrix in Fig.5 is used as the rotation matrix [13, 14].

```
JADE/current-to-pbest/1/adaptive GBX+BX()
{
+ $\mu_F^k$=$\mu_{CR}^k$=0.5, $k$=0,1;
  $\sigma_F = \sigma_{CR}$=0.1;
+ $R_{GBX}$=0.5; $\triangle R$=0.01;
// Initialize a population
  $P$=$N$ individuals generated randomly in $\mathcal{S}$;
  $FE$=$FE$+$N$;
  for($t$=1; $FE < FE_{max}$; $t$++) {
+   $(r_{kj})$=Correlation matrix obtained by Eq.(14);
+   $(\rho_{kj})$ is obtained by Eq.(17);
+   $\rho^{strong}$ is obtained by Eq.(20);
+   $m_{\rho_k}$, $k$=1,$\cdots$,$D$ are obtained by Eq.(21);
+   $S^k$=$\emptyset$, $k$=0,1;
+   $n^k$=0, $k$=0,1;   // number of trials for BX and GBX
    for($i$=1; $i \le N$; $i$++) {
+     if($u(0,1) < R_{GBX}$) $K$=1;  // GBX
+     else $K$=0;                    // BX
+     $n^K$++;
+     $CR_i = \mu_{CR}^K + N(0, \sigma_{CR}^2)$;
      if($CR_i < 0$) $CR_i$=0;
      else if($CR_i > 1$) $CR_i$=1;
      do {
+       $F_i$=$\mu_F^K + C(0, \sigma_F)$;
      } while($F_i \le 0$);
      if($F_i > 1$) $F_i = 1$;
      $\boldsymbol{x}_{pbest}$ = Randomly selected from top 100$p$% in $P$;
      $\boldsymbol{x}_{r1}$ = Randomly selected from $P(r1 \notin \{i\})$;
      $\boldsymbol{x}_{r2}$ = Randomly selected from $P(r2 \notin \{i, r1\})$;
      $\boldsymbol{m}_i = \boldsymbol{x}_i$+$F_i(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i)$+$F_i(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2})$;
+     if($K$==1)
+       $\boldsymbol{x}_i^{\mathrm{child}}$=GBX between $\boldsymbol{x}_i$ and $\boldsymbol{m}_i$;
+     else
        $\boldsymbol{x}_i^{\mathrm{child}}$=binomial crossover between $\boldsymbol{x}_i$ and $\boldsymbol{m}_i$;
      $FE$=$FE$+1;
// Survivor selection
      if($f(\boldsymbol{x}_i^{child}) < f(\boldsymbol{x}_i)$) {
        $\boldsymbol{z}_i = \boldsymbol{x}_i^{child}$;
        $S^K = S^K \cup \{(F_i, CR_i)\}$;
        // a success case is added
      }
      else $\boldsymbol{z}_i = \boldsymbol{x}_i$;
    }
    $P = \{\boldsymbol{z}_i\}$;
+   for($k$=0; $k < 2$; $k$++)
+     if($|S^k| > 0$) {
+       $\mu_F^k = (1-c)\mu_F^k + c\sum_{F_i \in S^k} F_i^2 / \sum_{F_i \in S^k} F_i$;
+       $\mu_{CR}^k = (1-c)\mu_{CR}^k + c\sum_{CR_i \in S^k} CR_i / |S^k|$;
+     }
+   if($n^0 > 0$ && $n^1 > 0$) {
+     $s^0 = |S^0|/n^0$; // success rate of BX
+     $s^1 = |S^1|/n^1$; // success rate of GBX
+     if($s^1 > s^0$)
+       $R_{GBX} = R_{GBX} + \triangle R$;
+     else if($s^0 > s^1$)
+       $R_{GBX} = R_{GBX} - \triangle R$;
+     if($R_{GBX} > 0.95$) $R_{GBX}$=0.95;
+     else if($R_{GBX} < 0.05$) $R_{GBX}$=0.05;
+   }
  }
}
```

Fig. 4: The algorithm of proposed method

$$M = \begin{pmatrix} \frac{1}{\sqrt{D}} & \frac{1}{\sqrt{D}} & \frac{1}{\sqrt{D}} & \cdots & \frac{1}{\sqrt{D}} & \frac{1}{\sqrt{D}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & \cdots & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{1}{\sqrt{(D-2)+(D-2)^2}} & \frac{1}{\sqrt{(D-2)+(D-2)^2}} & \frac{1}{\sqrt{(D-2)+(D-2)^2}} & \cdots & \frac{-(D-2)}{\sqrt{(D-2)+(D-2)^2}} & 0 \\ \frac{1}{\sqrt{(D-1)+(D-1)^2}} & \frac{1}{\sqrt{(D-1)+(D-1)^2}} & \frac{1}{\sqrt{(D-1)+(D-1)^2}} & \cdots & \frac{1}{\sqrt{(D-1)+(D-1)^2}} & \frac{-(D-1)}{\sqrt{(D-1)+(D-1)^2}} \end{pmatrix}$$

Fig. 5: Helmert matrix

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[12].

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 $u(x_i, a, k, m) =$ $\begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^D$
$f_{13}(x) = 0.1[\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2$ $\{1 + \sin^2(3\pi x_{i+1})\} + (x_D - 1)^2$ $\{1 + \sin^2(2\pi x_D)\}] + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$

5.2 Conditions of Experiments

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 $\{0, 0.5, 1, 1.5\}$ and $\triangle R$=0.01 for ADEGBX.

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.

5.3 Experimental Results

Table 2 shows the experimental results on JADE and ADEGBX in case of S_r=0, 0.5, 1 and 1.5. The results of JADE can be obtained by ADEGBX with fixing R_{GBX}=0. The maximum number of function evaluations is selected for each function and is shown in column labeled $FE_{\max}$. The mean value and the standard deviation of best objective values in 50 runs are shown for each function. The median value is also shown under the mean value. The best result among algorithms is highlighted using bold face fonts. Also, Wilcoxon signed rank test is performed and the result for each function is shown under the mean value. Symbols '+', '−' and '=' are shown when ADEGBX is significantly better than JADE, is significantly worse than JADE, and is not significantly different from JADE, respectively. Symbols '++' and '−−' are shown when the significance level is 1% and '+' and '−' are shown when the significance level is 5%.

ADEGBX (Sr=1 and 1.5) attained significantly better results than JADE in 12 functions except for f_7. ADEGBX (Sr=0.5) attained significantly better results than JADE in 11 functions except for f_4 and f_7. ADEGBX (Sr=0) attained significantly better results than JADE in 9 functions except for f_3, f_4, f_7 and f_{11}. JADE did not attained significantly better results than ADEGBX.

The median results excluding f_{11} are considered because the results of f_{11} are same in all methods. ADEGBX (Sr=0) attained the best median results in 6 functions f_2, f_6, f_8, f_9, f_{12} and f_{13}. ADEGBX (Sr=0.5) attained the best median results in 4 functions f_1, f_2, f_6 and f_{10}. ADEGBX (Sr=1.5) attained the best median results in 2 functions f_3 and f_5. ADEGBX (Sr=1) attained the best median result in f_4. JADE attained the best median result in f_7.

Table 3 shows the experimental results for rotated problems on JADE and ADEGBX in case of S_r=0, 0.5, 1 and 1.5.

ADEGBX (Sr=1) attained significantly better results than JADE in 11 functions except for f_7 and f_8.

Table 2: Experimental results

	FE_{max}	JADE	ADEGBX ($S_r = 0$)	ADEGBX ($S_r = 0.5$)	ADEGBX ($S_r = 1$)	ADEGBX ($S_r = 1.5$)
f_1	150,000	9.38e-59 ± 6.53e-58	6.25e-69 ± 3.94e-68	**3.32e-69 ± 1.62e-68**	1.26e-59 ± 8.84e-59	5.91e-64 ± 3.51e-63
		4.71e-66	8.42e-72 (++)	**4.73e-72** (++)	4.44e-69 (++)	1.96e-68 (++)
f_2	200,000	4.19e-31 ± 2.37e-30	1.40e-42 ± 8.86e-42	**9.79e-44 ± 5.27e-43**	9.21e-42 ± 4.07e-41	2.61e-34 ± 1.82e-33
		1.96e-37	**1.74e-47** (++)	**1.74e-47** (++)	9.84e-46 (++)	1.22e-43 (++)
f_3	500,000	8.17e-62 ± 3.01e-61	7.78e-60 ± 4.92e-59	2.37e-62 ± 1.25e-61	7.13e-65 ± 2.31e-64	**5.49e-65 ± 2.54e-64**
		2.30e-63	1.32e-64 (=)	9.38e-66 (++)	1.35e-66 (++)	**8.32e-68** (++)
f_4	500,000	2.01e-23 ± 9.83e-23	1.49e-23 ± 8.89e-23	4.95e-25 ± 1.09e-24	1.52e-24 ± 6.07e-24	**6.05e-26 ± 1.16e-25**
		9.27e-26	4.05e-26 (=)	1.27e-26 (=)	**7.15e-27** (++)	1.06e-26 (++)
f_5	150,000	5.83e-01 ± 3.56e+00	**1.59e-01 ± 7.81e-01**	3.19e-01 ± 1.08e+00	**1.59e-01 ± 7.81e-01**	2.39e-01 ± 9.47e-01
		3.04e-09	4.75e-13 (++)	1.16e-13 (++)	9.19e-13 (++)	**3.75e-14** (++)
f_6	10,000	3.02e+00 ± 1.26e+00	4.20e-01 ± 5.69e-01	**3.80e-01 ± 5.62e-01**	1.20e+00 ± 9.59e-01	1.68e+00 ± 1.12e+00
		3.00e+00	**0.00e+00** (++)	**0.00e+00** (++)	1.00e+00 (++)	2.00e+00 (++)
f_7	300,000	6.04e-04 ± 2.38e-04	**6.34e-04 ± 2.15e-04**	7.03e-04 ± 3.31e-04	6.83e-04 ± 2.60e-04	6.44e-04 ± 2.27e-04
		5.78e-04	6.54e-04 (=)	6.51e-04 (=)	6.61e-04 (=)	6.25e-04 (=)
f_8	100,000	**2.37e+00 ± 1.66e+01**	4.74e+00 ± 2.32e+01	4.74e+00 ± 2.32e+01	4.74e+00 ± 2.32e+01	**2.37e+00 ± 1.66e+01**
		2.87e-05	**2.38e-10** (++)	5.36e-10 (++)	3.34e-09 (++)	2.36e-07 (++)
f_9	100,000	1.01e-04 ± 3.91e-05	2.71e-06 ± 5.04e-06	**2.53e-06 ± 4.40e-06**	1.02e-05 ± 1.66e-05	2.26e-04 ± 1.38e-03
		9.19e-05	**1.07e-06** (++)	1.11e-06 (++)	2.55e-06 (++)	1.88e-05 (++)
f_{10}	50,000	9.20e-10 ± 6.43e-10	**4.32e-11 ± 2.74e-11**	4.85e-11 ± 4.00e-11	1.33e-10 ± 9.38e-11	3.28e-10 ± 3.86e-10
		7.15e-10	3.82e-11 (++)	**2.90e-11** (++)	1.14e-10 (++)	2.26e-10 (++)
f_{11}	50,000	1.15e-08 ± 6.91e-08	3.45e-04 ± 1.71e-03	3.94e-04 ± 1.99e-03	1.97e-04 ± 1.38e-03	**1.63e-13 ± 1.06e-12**
		0.00e+00	0.00e+00 (=)	0.00e+00 (+)	0.00e+00 (+)	0.00e+00 (++)
f_{12}	50,000	2.40e-16 ± 1.56e-15	**3.72e-21 ± 8.74e-21**	5.15e-21 ± 9.79e-21	4.02e-20 ± 1.01e-19	1.20e-18 ± 4.09e-18
		2.27e-18	**1.26e-21 (++)**	1.44e-21 (++)	1.18e-20 (++)	1.02e-19 (++)
f_{13}	50,000	1.15e-16 ± 2.23e-16	1.25e-19 ± 3.28e-19	**1.02e-19 ± 2.25e-19**	6.22e-19 ± 9.30e-19	1.48e-17 ± 5.49e-17
		2.69e-17	**2.56e-20** (++)	2.66e-20 (++)	3.11e-19 (++)	2.25e-18 (++)
+		—	9	11	12	12
=		—	4	2	1	1
−		—	0	0	0	0

Table 3: Experimental results for rotated problems

	FE_{max}	JADE	ADEGBX ($S_r = 0$)	ADEGBX ($S_r = 0.5$)	ADEGBX ($S_r = 1$)	ADEGBX ($S_r = 1.5$)
f_1	150,000	9.38e-59 ± 6.53e-58	6.25e-69 ± 3.94e-68	**3.32e-69 ± 1.62e-68**	1.26e-59 ± 8.84e-59	5.91e-64 ± 3.51e-63
		4.71e-66	8.42e-72 (++)	**4.73e-72 (++)**	4.44e-69 (++)	1.96e-68 (++)
f_2	200,000	**2.42e-26 ± 1.70e-25**	9.68e-17 ± 4.47e-16	1.19e-06 ± 8.24e-06	1.39e-05 ± 9.76e-05	4.17e-07 ± 2.92e-06
		8.02e-40	2.19e-44 (=)	**2.31e-45** (=)	2.33e-44 (+)	3.02e-42 (=)
f_3	500,000	2.29e-77 ± 6.76e-77	2.78e-78 ± 1.36e-77	**2.89e-80 ± 1.97e-79**	4.00e-80 ± 2.79e-79	3.33e-83 ± 1.75e-82
		3.02e-79	4.10e-82 (++)	3.95e-84 (++)	1.73e-84 (++)	**2.56e-85** (++)
f_4	500,000	1.99e-12 ± 7.88e-12	8.84e-24 ± 4.12e-23	2.66e-26 ± 1.45e-25	**2.47e-27 ± 7.15e-27**	6.82e-27 ± 3.47e-26
		4.13e-26	1.85e-27 (++)	9.90e-29 (++)	3.71e-29 (++)	**7.41e-30** (++)
f_5	150,000	5.75e+00 ± 3.77e+00	**3.00e+00 ± 2.69e+00**	4.15e+00 ± 3.09e+00	3.57e+00 ± 2.68e+00	4.04e+00 ± 3.02e+00
		4.54e+00	**2.61e+00** (++)	4.10e+00 (+)	3.09e+00 (++)	3.54e+00 (+)
f_6	10,000	3.48e+00 ± 1.22e+00	**5.00e-01 ± 6.08e-01**	6.00e-01 ± 6.63e-01	1.24e+00 ± 9.91e-01	2.28e+00 ± 1.04e+00
		3.00e+00	**0.00e+00** (++)	5.00e-01 (++)	1.00e+00 (++)	2.00e+00 (++)
f_7	300,000	7.15e-04 ± 3.19e-04	7.53e-04 ± 3.51e-04	7.51e-04 ± 3.10e-04	7.50e-04 ± 3.21e-04	**6.17e-04 ± 2.43e-04**
		6.75e-04	6.83e-04 (=)	6.79e-04 (=)	6.79e-04 (=)	**5.50e-04** (=)
f_8	100,000	2.04e+03 ± 4.08e+02	2.13e+03 ± 3.35e+02	**2.00e+03 ± 4.39e+02**	2.07e+03 ± 4.43e+02	2.12e+03 ± 4.59e+02
		2.09e+03	2.14e+03 (=)	2.08e+03 (=)	**2.06e+03** (=)	2.24e+03 (=)
f_9	100,000	3.68e+01 ± 4.63e+00	3.19e+01 ± 4.42e+00	3.21e+01 ± 4.69e+00	3.01e+01 ± 4.67e+00	**3.00e+01 ± 4.74e+00**
		3.70e+01	3.28e+01 (++)	3.25e+01 (++)	3.03e+01 (++)	**2.96e+01** (++)
f_{10}	50,000	7.62e-10 ± 5.20e-10	**4.81e-11 ± 3.18e-11**	5.00e-11 ± 4.33e-11	1.56e-10 ± 1.54e-10	2.46e-10 ± 2.37e-10
		6.87e-10	4.12e-11 (++)	**3.82e-11** (++)	1.20e-10 (++)	1.69e-10 (++)
f_{11}	50,000	1.50e-04 ± 1.04e-03	1.97e-04 ± 1.38e-03	**6.17e-12 ± 3.03e-11**	2.23e-04 ± 1.39e-03	1.48e-04 ± 1.04e-03
		3.60e-14	**5.55e-17** (++)	5.55e-16 (++)	1.94e-15 (+)	6.99e-15 (++)
f_{12}	50,000	9.13e-13 ± 3.22e-12	**1.75e-15 ± 2.97e-15**	1.34e-14 ± 8.27e-14	7.17e-15 ± 9.62e-15	3.15e-14 ± 7.74e-14
		4.85e-14	8.16e-16 (++)	**7.87e-16** (++)	3.31e-15 (++)	5.67e-15 (++)
f_{13}	50,000	6.85e-11 ± 4.00e-10	**8.11e-15 ± 1.80e-14**	1.06e-14 ± 2.04e-14	5.37e-14 ± 1.22e-13	5.20e-13 ± 1.07e-12
		6.14e-13	**2.01e-15** (++)	3.53e-15 (++)	1.89e-14 (++)	1.05e-13 (++)
+		—	10	10	11	10
=		—	3	3	2	3
−		—	0	0	0	0

ADEGBX (Sr=0, 0.5, 1.5) attained significantly better results than JADE in 10 functions except for f_2, f_7 and f_8. JADE did not attained significantly better results than ADEGBX.

ADEGBX (Sr=0) attained the best median results in 4 functions f_5, f_6, f_{11} and f_{13}. ADEGBX (Sr=0.5) attained the best median results in 4 functions f_1, f_2, f_{10} and f_{12}. ADEGBX (Sr=1.5) attained the best median results in 4 functions f_3, f_4, f_7 and f_9. ADEGBX (S_r=1) attained the best median result in f_8. JADE attained no best median result.

It is thought that from the viewpoint of statistical test ADEGBX (Sr=1) is the most stable method although the median values are not good compared with other ADEGBX methods.

6 Conclusions

We proposed a crossover operation GBX which groups genes according to correlation coefficients and the genes in the same group are crossed or not crossed simultaneously. The groups are decided so that highly correlated variables are inherited at the same time. Also, adaptive control of the probability of applying GBX is proposed. From numerical experiments, it is shown that ADEGBX (Sr=1) attained significantly better results and better median 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.

References

[1] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[2] K. Price, R. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

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

[4] S. Das and P. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[5] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] Y. Akimoto, Y. Nagata, J. Sakuma, I. Ono, and S. Kobayashi, "Proposal and evaluation of adaptive real-coded crossover arex," *Trans. of the Japanese Society of Artificial Intelligence*, vol. 24, no. 6, pp. 446–458, 2009, in Japanese.

[14] S. Sakai and T. Takahama, "A study on selecting an oblique coordinate system for rotation-invariant blend crossover in a real-coded genetic algorithm," in *Recent Studies in Economic Sciences: Information Systems, Project Managements, Economics, OR and Mathematics*, A.Kadoya and H.Teramoto, Eds. Kyushu University Press, Mar. 2018, pp. 65–87.