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

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

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

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 bino-
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 variable 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:

\[
\begin{align*}
\triangle f_i &= f(\ldots, x_i', \ldots, x_j, \ldots) - f(\ldots, x_i, \ldots, x_j, \ldots) \quad (1) \\
\triangle f_j &= f(\ldots, x_i, \ldots, x_j', \ldots) - f(\ldots, x_i, \ldots, x_j, \ldots) \quad (2) \\
\triangle f_{ij} &= f(\ldots, x_i', \ldots, x_j', \ldots) - f(\ldots, x_i, \ldots, x_j, \ldots) \quad (3)
\end{align*}
\]

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.

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

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:

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

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.

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad l_j \leq x_j \leq u_j, \quad j = 1, \ldots, D, \\
& \quad \text{where } x = (x_1, x_2, \ldots, x_D) \text{ is a } D \text{ dimensional vector and } f(x) \text{ is an objective function. The function } f \text{ is a nonlinear real-valued function. Values } l_j \text{ and } u_j \text{ are the lower bound and the upper bound of } x_j, \text{ respectively.}
\end{align*}
\]

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 \( x_i, i = 1, 2, \ldots, N \) 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 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
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].

where an outside element of the population is used as a base vector. A mutation vector is generated by current-to-pbest without an archive as follows:

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 re-generated if \( F_i \leq 0 \). The location \( \mu_{CR} \) and the mean \( \mu_{CR} \) are updated as follows:

\[
\mu_{CR} = (1-c) \mu_{CR} + c S_{F^2}/ S_F \quad (10)
\]

\[
\mu_{CR} = (1-c) \mu_{CR} + c S_{CR}/ S_N \quad (11)
\]

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

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

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

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

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

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:

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

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

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

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

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:

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

The $\rho^{strong}$ is a threshold value. In order to avoid too small threshold value, the minimum value of $\rho^{strong}$ 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 $CR_r$. The gene, or variable $x_{mv_k}$, which has the largest coefficient with each gene, or variable $x_k$ is defined as follows:

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

for ($j=1$; $j \leq D$; $j++$) $cross[j]=1$; // crossover flags
for ($i=1$; $i \leq D$; $i++$) { if($cross[i]=0$); // already in a group
else if($r_{j_{rand},j} > \rho^{strong}$) $cross[j]=1$;
else if($u(0,1) < CR_r$) $cross[j]=1$;
else $cross[j]=0$;
if($cross[m_{\rho_j},j]=1$) & & $\rho_{j,m_{\rho_j}} > \rho^{strong}$)
if($cross[m_{\rho_j}]=cross[j]$);
$j=(j+1)\%D$;
}

Step1 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.

Step2 Termination condition. If the number of function evaluations exceeds the maximum number of evaluations $FE_{\text{max}}$, 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 ($k = 0, 1$).

**Step 4** DE operation with adaptive parameters. GBX or BX is selected according to $R_{GBX}$. $K$ is set to 0 in case of BX and to 1 in case of GBX. The scaling factor $F_i$ is generated according to Cauchy distribution using $\mu^K_F$. The crossover rate $CR_i$ is generated according to the normal distribution using $\mu^K_{CR}$. GBX 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 the scaling factor. The means of the scaling factor $\mu^K_F$ and the means of crossover rate $\mu^K_{CR}$ 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 $|.|$ 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.

**Step 7** Go back to Step 2.

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 $z$ is converted as $x = Mz$ and $f(x)$ is minimized, where $M$ is a rotation matrix. In this study, Helmert matrix in Fig.5 is used as the rotation matrix [13, 14].

![Fig 4: The algorithm of proposed method](image-url)
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].

<table>
<thead>
<tr>
<th>Test functions</th>
<th>Search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x) = \sum_{i=1}^{D} x_i^2$</td>
<td>[-100, 100]^D</td>
</tr>
<tr>
<td>$f_2(x) = \sum_{i=1}^{D}</td>
<td>x_i</td>
</tr>
<tr>
<td>$f_3(x) = \max(</td>
<td>x_i</td>
</tr>
<tr>
<td>$f_4(x) = \sum_{i=1}^{D} [100(x_{i+1})^2 + (x_i - 1)^2]$</td>
<td>[-30, 30]^D</td>
</tr>
<tr>
<td>$f_5(x) = \sum_{i=1}^{D}</td>
<td>x_i + 0.5</td>
</tr>
<tr>
<td>$f_7(x) = \sum_{i=1}^{D} x_i^4 + \text{rand}(0,1)$</td>
<td>[-1.28, 1.28]^D</td>
</tr>
<tr>
<td>$f_8(x) = \sum_{i=1}^{D} -ix_i \sin \sqrt{</td>
<td>x_i</td>
</tr>
<tr>
<td>$f_9(x) = \sum_{i=1}^{D} x_i^2 - 10 \cos(2\pi x_i) + 10$</td>
<td>[-5.12, 5.12]^D</td>
</tr>
<tr>
<td>$f_{10}(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^{D} x_i^2 \right) + 20 + e$</td>
<td>[-32, 32]^D</td>
</tr>
<tr>
<td>$f_{11}(x) = \frac{1}{\prod_{i=1}^{D} x_i} - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right)$ + 1</td>
<td>[-600, 600]^D</td>
</tr>
<tr>
<td>$f_{12}(x) = \frac{1}{\prod_{i=1}^{D} x_i} - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right)$ + 1</td>
<td>[-50, 50]^D</td>
</tr>
<tr>
<td>$f_{13}(x) = -0.1 \sin \left( 3\pi x_1 \right) + \sum_{i=1}^{D} x_i^2 (x_{i+1}) + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$</td>
<td>[-50, 50]^D</td>
</tr>
</tbody>
</table>

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 $\mu_C = 0.5$, the best parameter $p=0.05$, and the learning parameter $c=0.1$. $S_r$ is selected from $\{0, 0.5, 1, 1.5\}$ and $\Delta R=0.01$ 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 ($S_r=1$ and 1.5) attained significantly better results than JADE in 12 functions except for $f_7$. ADEGBX ($S_r=0.5$) attained significantly better results than JADE in 11 functions except for $f_4$ and $f_7$. ADEGBX ($S_r=0$) 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 ($S_r=0$) attained the best median results in 6 functions $f_2$, $f_6$, $f_8$, $f_9$, $f_{12}$ and $f_{13}$. ADEGBX ($S_r=0.5$) attained the best median results in 4 functions $f_1$, $f_2$, $f_6$ and $f_{10}$. ADEGBX ($S_r=1$) attained the best median results in 2 functions $f_3$ and $f_5$. ADEGBX ($S_r=1$) attained the best median result in $f_7$. 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 ($S_r=1$) attained significantly better results than JADE in 11 functions except for $f_7$ and $f_8$. 

![Fig. 5: Helmert matrix](image-url)
### Table 2: Experimental results

<table>
<thead>
<tr>
<th>$F_{E_{\text{max}}}$</th>
<th>JADE</th>
<th>ADEGBX ($S_r = 0$)</th>
<th>ADEGBX ($S_r = 0.5$)</th>
<th>ADEGBX ($S_r = 1$)</th>
<th>ADEGBX ($S_r = 1.5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$</td>
<td>150,000</td>
<td>9.38e-59 ± 6.53e-58</td>
<td>3.32e-69 ± 1.62e-68</td>
<td>3.26e-69 ± 1.52e-68</td>
<td>2.70e-69 ± 1.36e-68</td>
</tr>
<tr>
<td>$2$</td>
<td>200,000</td>
<td>4.19e-31 ± 2.37e-30</td>
<td>4.73e-72 ± 1.26e-72</td>
<td>3.19e-72 ± 1.14e-72</td>
<td>2.80e-72 ± 1.04e-72</td>
</tr>
<tr>
<td>$3$</td>
<td>500,000</td>
<td>8.17e-60 ± 3.01e-61</td>
<td>7.99e-44 ± 5.27e-44</td>
<td>7.19e-44 ± 4.73e-44</td>
<td>5.80e-44 ± 4.34e-44</td>
</tr>
<tr>
<td>$5$</td>
<td>150,000</td>
<td>5.83e-01 ± 3.56e-00</td>
<td>1.59e-01 ± 0.78e-01</td>
<td>1.19e-01 ± 0.60e-01</td>
<td>8.01e-01 ± 0.45e-01</td>
</tr>
<tr>
<td>$6$</td>
<td>500,000</td>
<td>3.14e-01 ± 0.50e-01</td>
<td>3.80e-00 ± 0.50e-00</td>
<td>3.00e-00 ± 0.40e-00</td>
<td>2.75e-00 ± 0.35e-00</td>
</tr>
<tr>
<td>$7$</td>
<td>300,000</td>
<td>6.04e-04 ± 0.23e-04</td>
<td>6.34e-04 ± 0.21e-04</td>
<td>6.03e-04 ± 0.19e-04</td>
<td>5.84e-04 ± 0.17e-04</td>
</tr>
<tr>
<td>$8$</td>
<td>100,000</td>
<td>2.37e-00 ± 1.66e-01</td>
<td>2.38e-00 ± 1.66e-01</td>
<td>2.37e-00 ± 1.66e-01</td>
<td>2.36e-00 ± 1.66e-01</td>
</tr>
<tr>
<td>$9$</td>
<td>100,000</td>
<td>1.01e-04 ± 0.39e-01</td>
<td>1.07e-04 ± 0.39e-01</td>
<td>1.05e-04 ± 0.38e-01</td>
<td>1.04e-04 ± 0.37e-01</td>
</tr>
<tr>
<td>$10$</td>
<td>50,000</td>
<td>9.20e-10 ± 3.63e-10</td>
<td>4.32e-11 ± 2.74e-11</td>
<td>3.92e-11 ± 2.63e-11</td>
<td>3.53e-11 ± 2.54e-11</td>
</tr>
<tr>
<td>$11$</td>
<td>50,000</td>
<td>1.15e-08 ± 6.91e-09</td>
<td>3.45e-04 ± 1.71e-04</td>
<td>3.00e-04 ± 1.64e-04</td>
<td>2.56e-04 ± 1.57e-04</td>
</tr>
<tr>
<td>$12$</td>
<td>50,000</td>
<td>2.40e-16 ± 1.56e-16</td>
<td>6.14e-04 ± 3.15e-04</td>
<td>7.80e-04 ± 3.90e-04</td>
<td>1.02e-03 ± 3.82e-03</td>
</tr>
<tr>
<td>$13$</td>
<td>50,000</td>
<td>1.15e-16 ± 2.23e-16</td>
<td>2.56e-20 ± 0.00e-00</td>
<td>2.66e-20 ± 0.00e-00</td>
<td>3.11e-20 ± 0.00e-00</td>
</tr>
<tr>
<td>$+$</td>
<td></td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>$=$</td>
<td></td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$-$</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 3: Experimental results for rotated problems

<table>
<thead>
<tr>
<th>$F_{E_{\text{max}}}$</th>
<th>JADE</th>
<th>ADEGBX ($S_r = 0$)</th>
<th>ADEGBX ($S_r = 0.5$)</th>
<th>ADEGBX ($S_r = 1$)</th>
<th>ADEGBX ($S_r = 1.5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1$</td>
<td>150,000</td>
<td>9.38e-59 ± 6.53e-58</td>
<td>3.32e-69 ± 1.62e-68</td>
<td>3.20e-69 ± 1.51e-68</td>
<td>2.75e-69 ± 1.45e-68</td>
</tr>
<tr>
<td>$2$</td>
<td>200,000</td>
<td>4.19e-31 ± 2.37e-30</td>
<td>4.73e-72 ± 1.26e-72</td>
<td>3.19e-72 ± 1.14e-72</td>
<td>2.80e-72 ± 1.04e-72</td>
</tr>
<tr>
<td>$3$</td>
<td>500,000</td>
<td>8.17e-60 ± 3.01e-61</td>
<td>7.99e-44 ± 5.27e-44</td>
<td>7.19e-44 ± 4.73e-44</td>
<td>5.80e-44 ± 4.34e-44</td>
</tr>
<tr>
<td>$5$</td>
<td>150,000</td>
<td>5.83e-01 ± 3.56e-00</td>
<td>1.59e-01 ± 0.78e-01</td>
<td>1.19e-01 ± 0.60e-01</td>
<td>8.01e-01 ± 0.45e-01</td>
</tr>
<tr>
<td>$6$</td>
<td>500,000</td>
<td>3.14e-01 ± 0.50e-01</td>
<td>3.80e-00 ± 0.50e-00</td>
<td>3.00e-00 ± 0.40e-00</td>
<td>2.75e-00 ± 0.35e-00</td>
</tr>
<tr>
<td>$7$</td>
<td>300,000</td>
<td>6.04e-04 ± 0.23e-04</td>
<td>6.34e-04 ± 0.21e-04</td>
<td>6.03e-04 ± 0.19e-04</td>
<td>5.84e-04 ± 0.17e-04</td>
</tr>
<tr>
<td>$8$</td>
<td>100,000</td>
<td>2.37e-00 ± 1.66e-01</td>
<td>2.38e-00 ± 1.66e-01</td>
<td>2.37e-00 ± 1.66e-01</td>
<td>2.36e-00 ± 1.66e-01</td>
</tr>
<tr>
<td>$9$</td>
<td>100,000</td>
<td>1.01e-04 ± 0.39e-01</td>
<td>1.07e-04 ± 0.39e-01</td>
<td>1.05e-04 ± 0.38e-01</td>
<td>1.04e-04 ± 0.37e-01</td>
</tr>
<tr>
<td>$10$</td>
<td>50,000</td>
<td>9.20e-10 ± 3.63e-10</td>
<td>4.32e-11 ± 2.74e-11</td>
<td>3.92e-11 ± 2.63e-11</td>
<td>3.53e-11 ± 2.54e-11</td>
</tr>
<tr>
<td>$11$</td>
<td>50,000</td>
<td>1.15e-08 ± 6.91e-09</td>
<td>3.45e-04 ± 1.71e-04</td>
<td>3.00e-04 ± 1.64e-04</td>
<td>2.56e-04 ± 1.57e-04</td>
</tr>
<tr>
<td>$12$</td>
<td>50,000</td>
<td>2.40e-16 ± 1.56e-16</td>
<td>6.14e-04 ± 3.15e-04</td>
<td>7.80e-04 ± 3.90e-04</td>
<td>1.02e-03 ± 3.82e-03</td>
</tr>
<tr>
<td>$13$</td>
<td>50,000</td>
<td>1.15e-16 ± 2.23e-16</td>
<td>2.56e-20 ± 0.00e-00</td>
<td>2.66e-20 ± 0.00e-00</td>
<td>3.11e-20 ± 0.00e-00</td>
</tr>
<tr>
<td>$+$</td>
<td></td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>$=$</td>
<td></td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$-$</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Experimental results for rotated problems

Table 3: Experimental results for rotated problems
ADEGBX ($S_r=0, 0.5, 1.5$) 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 ($S_r=0$) attained the best median results in 4 functions $f_5$, $f_6$, $f_{11}$ and $f_{13}$. ADEGBX ($S_r=0.5$) attained the best median results in 4 functions $f_1$, $f_2$, $f_{10}$ and $f_{12}$. ADEGBX ($S_r=1.5$) 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 ($S_r=1$) 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 ($S_r=1$) 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


