SIMULATED ANNEALING AND ADAPTIVE DYNAMIC VARIABLE BAND MUTATION FOR STRUCTURAL OPTIMIZATION BY GENETIC ALGORITHMS

A. Kaveh*1 and M. Shahrouzi2
1Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran 16, Iran
2Department of Structural Engineering, International Institute of Earthquake Engineering and Seismology, Tehran 19, Iran

ABSTRACT

Genetic Algorithm is known as a general tool for optimization in the field of discrete variables such as the case of structural problems. For such a search to be efficient and practical, the genetic parameters should be selected of proper type and should be fine tuned. For this purpose, a direct index coding is utilized in this paper and the corresponding mutation operator is improved in two ways. First, it is modified incorporating simulated annealing concepts to form a generation based genetic algorithm. A dynamic variable band mutation is also developed as the second attempt in order to properly adopt mutation operator with the treated problem during optimization. A number of structural examples are treated by the proposed method to investigate its effect on efficiency and convergence rate of the genetic search.

Keywords: Genetic algorithm, structural layout optimization, direct index coding, simulated annealing, minimal length chromosome

1. INTRODUCTION

Since 1975 when the Genetic Algorithms (GAs) were introduced to simulate strategy of survival of the fittest from the nature to numerical methods [1], several investigators have expanded GAs applications in various fields of optimization problems [2-7]. Genetic algorithms work on a discrete genotype space as an encoded map of the physical alternatives called phenotypes. Different types of variables can be coded in one genotype while the information exchange between the genotype and phenotype spaces takes place only in fitness evaluation of decoded solutions without any need to gradient evaluation. Thus, even multi-modal non-convex discontinuous problems can be treated by GA as a generalized

* Email address of the corresponding author: alikaveh@iust.ac.ir
method of stochastic optimization. Such a search is dominated by GA operators specially crossover, mutation and selection\cite{7,8}. Crossover provides exploration phase of GA taking merit of the direct information share between a set of coded solution trials known as chromosomes in order to evolve total desirability or fitness of such a set called population through its generations. The convergence proof of such operators is governed by schema theory and building block hypothesis\cite{1,8,9}. It is also necessary to provide GA an exploitation strategy in order to enable it to escape from local optima toward the global optimum. The feature is mainly dominated by the mutation operator. Using these operators together with selection of fitter individuals as parent set of the next generation, it is aimed to improve the fitness of the best individuals in the population in charge of expanding more computational effort as the generation number increases. However, it is crucial to provide fine balance between the exploration and exploitation operators in order to preserve GA convergence within reasonable number of generations rather than shifting it toward a pure random search. A number of investigators have studied the effect of genetic parameters on its convergence\cite{10-13} and implied that the efficiency of GA is mainly governed by the mutation parameter and population size\cite{14}. The authors have shown that the proper encoding scheme can also have considerable effect in reducing cardinality of the genotype search space leading to better convergence and efficiency of the genetic search\cite{15}.

In this paper improving mutation operator for direct index coding\cite{15,16} in discrete structural problems is concerned using two distinct approaches. First the variation of the direct mutation band is performed following a developed envelope function by the strategy incorporated in the simulated annealing approaches\cite{17-20}. In the second attempt, a dynamic adaptive strategy is developed to alter mutation band due to the problem specific conditions. Both methods are then applied to a number of literature examples and the results are compared through discussion.

2. DIRECT INDEX ENCODING FOR GENETIC ALGORITHM

In genetic notation, each genotype is denoted by a chromosome consisting of a number of genes. A pre-assigned set of character alphabet is considered to evaluate each gene value called an allele. The number of characters assignable to each gene is governed by the encoding method and indicates the number of its variants in the genotype space which may even differ from the other genes.

Binary encoding as a common method uses minimum number of gene variants; 2 between 0 and 1, in charge of leading to maximum chromosome length for any given set of non-binary design parameters. A number of problems are associated with this type of encoding when any of the design parameters has more than 2 variants.

First, the one-to-one corresponding between phenotypes and genotypes may be violated when the number of variants for a parameter in the phenotype space is not an exact power of 2. Consider the case that \( m \) bits are required to encode the j-th design parameter with \( n_j \) variants and \( n_j < 2^m \). Thus, some \( B = 2^k \) may arise as decimal equivalent of the generated bit-string during GA process such that \( B > n_j \). This out-of-bound allele gap should then be
compensated by correction formulas such as below [21] which may violate the one-to-one correspondence between phenotypes and genotypes as a base of the genetic search:

\[
D = D_{\text{min}} + \left( D_{\text{max}} - D_{\text{min}} \right) \frac{B}{2^m}
\]  

(1)

\(D_{\text{min}}\) and \(D_{\text{max}}\) denote lower and upper bounds for decimal parameter \(D\) and \(m\) is the minimum number of bits required for binary decoding. In another word, a hidden type of mutation will occur in binary coding of non-binary parameters due to such a gap, Figure 1.

The second is called \textit{variable Hamming distance problem}. It means a single bit-exchange in a binary expansion of a non-binary parameter results in variable changes in the value of that parameter according to the site of the exchanging bit in that bit-string. As a result some directions in phenotype space are undesirably preferred to the others and it disaffects the GA efficiency or convergence rate [16].

The third effect is associated with crossover imposed on the whole binary coded chromosome. When a site of binary crossover falls between the boundaries of a bit-string indicating a design parameter, it causes a hidden mutation in the value of that parameter, Figure 2. Finally, a fixed mutation rate in the binary coded chromosome does not mean the same mutation rate regarding the corresponding parameters because such a mutation is taking place in somewhat different number of bits in the bit-string representation of different parameters. Therefore, an uncontrolled type of variable mutation is really associated with binary encoding of non-binary parameters.

\begin{table}[h]
\centering
\begin{tabular}{cccccc}
  \hline
  & \(D_1=25\) & \(D_2=42\) & \(D_3=32\) & \(D_4=24\) & \(D_5=31\) \\
  \hline
  & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
\end{tabular}
\end{table}

(a)

\begin{table}[h]
\centering
\begin{tabular}{cccccc}
  \hline
  & \(D_1=27\) & \(D_2=58\) & \(D_3=0\) & \(D_4=24\) & \(D_5=62\) \\
  \hline
  & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
\end{tabular}
\end{table}

(b)

\begin{table}[h]
\centering
\begin{tabular}{cccccc}
  \hline
  & \(D_1=27\) & \(D_2=58\) & \(D_3=16\) & \(D_4=0\) & \(D_5=58\) \\
  \hline
  & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\end{tabular}
\end{table}

(c)

Figure 1. A direct index chromosome and its binary expansion a) before and b) after implementing a binary mutation. Mutation of only 1-bit in each of the first 3 parameters has resulted in different range of integer index mutation (variable Hamming distances). c) Hidden mutation due to out-of-bound correction of indices when maximum index is taken as 60
The above facts led to utilize a Direct Index Coding, DIC [15,16] by which the mutation band can be explicitly controlled as an important factor affecting the GA convergence. As every j-th parameter of the problem with \( n_j \)-variants can be minimally addressed by \( n_j \)-indices, a character alphabet from 1 to \( n_j \) is assumed for the corresponding direct indexed gene. The resulted direct indexed chromosome will then have the minimum required number of genes. Such an encoding method preserves the one-to-one correspondence between the coded genotypes and the problem phenotypes, while reducing the genetic search space to its minimum required avoiding out-of-bound alleles. The 1-point and uniform direct crossover operator can be defined in the same way as the case of binary encoding. The authors have developed a number of mutation operators for DIC among which the following types are reviewed in the following:

**Direct Index Mutation (DIM):** In this method the allele of every gene is altered to another index in its range of assignable indices, when a randomly generated number falls below the mutation rate, \( P_m \).

**Limited Band Direct Index Mutation (LBDIM):** In this method, the assignable character set for each gene is limited to less than its maximum available with the center of the current
allele of that gene.

**Variable Band Direct Index Mutation (VBDIM):** This mutation is a type of LBDIM in which the length of character set for each gene is not fixed during the GA process.

The VBDIM is in fact a compromise to a class of variable length binary coded chromosomes or grouping methods offered in literature [4,22-24].

### 3. ANNEALED VARIABLE BAND MUTATION

In early stages of a genetic search, it is desired to access starting points in all regions of local optima to maximize probability of exploring the global optimum in later generations. As the generation number increases, the GA process evolves fitness of greater portion of individuals in the population. Thus in the last generations high mutation range is not desired because it can disturb the evolved population causing loss of the improved genetic material and consequent reduction of the convergence rate. The concept can be utilized by gradual decrease in the index limit of a LBDIM through generations.

In simulated annealing approach probability of jump acceptance to any neighbor region of local or global optima, is gradually decreased according to a cooling temperature function. The same strategy can be used in gradual decrease of mutation rate or band. In this article variation in the mutation band of a VBDIM is concerned using the following envelope function:

\[
f_c(k) = e^{-\frac{-\Delta E}{C_B T(k)}alphabet}
\]

In which \( C_B \) is the Boltzman coefficient and \( \Delta E \) as energy difference is taken constant of unity in the present study. The simulated temperature, \( T \), is decreased with generation number, \( k \), according to the developed cooling function as:

\[
T(k) = \begin{cases} 
\frac{N}{k-1}, & k > 1 \\
1, & k = 1 
\end{cases}
\]

\[
N \rightarrow \frac{1}{T} = 0, k = 1
\]

\( N \) is the maximum generation number in the GA search. The recent relation is similar to that referenced in *Very Fast Simulated Annealing* techniques [18,25]; except that this form preserves the envelope function of unity in the first generation. Such an envelope function is then used to compute the mutation band, \( R_j \), for any \( j \)-th direct indexed gene with maximum variants of \( n_j \):

\[
R_j(k) = \max(2, f_c(k) \times n_j)
\]
This method is hereinafter called *Annealed Variable Band Mutation* (AVBM). Figure 3 demonstrates temperature cooling envelope curves for some sample values of $C_B$ as the Boltzman coefficient in this study.

![Figure 3. Sample envelope function curves for various Boltzman coefficients.](image)

### 4. DYNAMIC VARIABLE BAND MUTATION

As mentioned before, proper convergence of the genetic algorithm is related to finding true balance between its exploration and exploitation agents. In the case of direct indexed encoding, the exploitation feature not only depends to mutation rate but also to the mutation index band employed for every gene. In this work such a proper balance is searched by adaptive problem-specific tuning the mutation band of each design parameter. The concept can be inspired using *Dynamic Variable Band Mutation* (DVBM), in the following algorithm:

1. Initiate the mutation band of every $j$-th gene with its maximum number of variants:

   $$ R_j(k) = n_j $$  \hspace{1cm} (5)

2. Select the mutation rate, $P_m$, with an estimate of its proper value small enough to start GA convergence with the initiated mutation band.

3. For later generations as soon as the elitist fitness improves, start decreasing the
mutation band due to the following relation:

\[ R_j(k) = \max(2, R_j(k-1) - 1) \]  \hspace{1cm} (6)

4. Continue such degradation until the current fitness improvement lag, \( D_F^{\text{Elitist}}(k) \), equals \( \alpha \) times the maximum experienced lag till that generation, \( L_F^{\text{Elitist}}(k) \). The coefficient \( \alpha \) is taken as unity in the present study.

5. Increase the mutation band by the following relation but return to step-3 as soon as an improvement in the elitist fitness occurs.

\[ R_j(k) = \min(n_j, R_j(k-1) + 1) \]  \hspace{1cm} (7)

6. Repeat steps 3 to 5 until the GA stopping criterion is satisfied.

The stopping criterion for GA is usually taken a fixed number of generations; however, the following criterion may also be used as an alternative:

\[ \varsigma(k) = \varsigma_{\text{stop}} \]  \hspace{1cm} (8)

Whereas:

\[ \varsigma(k) = \frac{D_F^{\text{Elitist}}(k)}{L_F^{\text{Elitist}}(k)} \]  \hspace{1cm} (9)

\( \varsigma_{\text{stop}} \) is selected a pre-assigned value greater than 1. An elitist strategy \([26,27,28]\) is also employed in this work in which the fittest individual of every previous generation is put to the current mating pool in order not to lose the best genetic material.

5. ILLUSTRATIVE EXAMPLES

A number of structural examples are treated here to study effectiveness of the proposed methods via comparison with binary coded standard genetic algorithms. Sizing optimization with different sets of constraints and available sections for truss and frame members is considered in examples 1 and 3 as well as a layout optimization problem in example 2. The examples are treated for structural weight minimization by GA while the tournament selection is employed and penalty coefficient, \( K_p \), is taken as 10, Ref. [3]. Other parameters are given in the Tables.
5.1 Example 1: Size optimization for a 10-bar planar truss
As a well-known literature benchmark, the 10-bar 6-node truss of Figure 3 is selected here to search best structural profiles among the available section list $S_1$ in Table 1, Refs. [3,28-33]. Two point loads of Figure 3 are considered as a single loading condition. Material properties are taken as $E = 68.95\text{GN/m}^2$, $\rho = 27.1\text{kN/m}^3$ while the constraint on member stresses is $\sigma^m = \pm 172\text{MN/m}^2$ and nodal displacements are confined to $|\Delta^{x,y}| \leq 0.0508\text{m}$.

<table>
<thead>
<tr>
<th>Profile Index</th>
<th>List name: $S_1$</th>
<th>Section Area (cm$^2$)</th>
<th>$S_2$</th>
<th>Section Area (cm$^2$)</th>
<th>$S_3$</th>
<th>Section ID (AISC profiles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_1$</td>
<td>10.45</td>
<td></td>
<td>10.45</td>
<td></td>
<td>W6X25</td>
</tr>
<tr>
<td>2</td>
<td>$S_2$</td>
<td>11.61</td>
<td></td>
<td>11.61</td>
<td></td>
<td>W8X10</td>
</tr>
<tr>
<td>3</td>
<td>$S_3$</td>
<td>12.84</td>
<td></td>
<td>15.35</td>
<td></td>
<td>W8X13</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>13.74</td>
<td></td>
<td>16.90</td>
<td></td>
<td>W8X18</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>15.35</td>
<td></td>
<td>18.58</td>
<td></td>
<td>W8X21</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>16.90</td>
<td></td>
<td>19.94</td>
<td></td>
<td>W8X24</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>16.97</td>
<td></td>
<td>20.19</td>
<td></td>
<td>W8X28</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>18.58</td>
<td></td>
<td>21.81</td>
<td></td>
<td>W8X31</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>18.90</td>
<td></td>
<td>23.42</td>
<td></td>
<td>W10X15</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>19.93</td>
<td></td>
<td>24.77</td>
<td></td>
<td>W10X17</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>20.19</td>
<td></td>
<td>24.97</td>
<td></td>
<td>W10X19</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>21.81</td>
<td></td>
<td>26.97</td>
<td></td>
<td>W10X22</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>22.39</td>
<td></td>
<td>28.97</td>
<td></td>
<td>W10X26</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>22.90</td>
<td></td>
<td>30.97</td>
<td></td>
<td>W10X30</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>23.42</td>
<td></td>
<td>32.06</td>
<td></td>
<td>W12X14</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>24.77</td>
<td></td>
<td>33.03</td>
<td></td>
<td>W12X16</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>24.97</td>
<td></td>
<td>37.03</td>
<td></td>
<td>W12X19</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>25.03</td>
<td></td>
<td>46.58</td>
<td></td>
<td>W12X22</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>26.97</td>
<td></td>
<td>51.42</td>
<td></td>
<td>W12X26</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>27.23</td>
<td></td>
<td>74.19</td>
<td></td>
<td>W12X30</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>28.97</td>
<td></td>
<td>87.10</td>
<td></td>
<td>W14X22</td>
</tr>
</tbody>
</table>
Table 1 continued

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>31</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>33</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>35</td>
</tr>
<tr>
<td>36</td>
</tr>
<tr>
<td>37</td>
</tr>
<tr>
<td>38</td>
</tr>
<tr>
<td>39</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>41</td>
</tr>
<tr>
<td>42</td>
</tr>
</tbody>
</table>

The problem is formulated as below while the equality constraint refers to the static...
A. Kaveh and M. Shahrouzi

equilibrium relation:

\begin{align*}
\text{Minimize} & \quad w = \rho \sum_{i=1}^{\text{M}} A_i L_i \\
\text{Subject to:} & \quad KU - P = 0 \\
& \quad g_j^d = \left[ \frac{\Delta_j^l}{\Delta_j^a} \right] - 1 \leq 0 \\
& \quad g_i^s = \left[ \frac{\sigma_i}{\sigma_a^i} \right] - 1 \leq 0
\end{align*}

For \( i = 1, \ldots, \text{ne}, \quad j = 1, \ldots, \text{nDOF} \)

Where \( K, U \) and \( P \) are the stiffness matrix, nodal displacement and loading vectors, respectively. \( L_i, A_i \) denote length and cross-section area of the \( i \)-th member, while \( \rho \) is the material density. \( \Delta_j^l \) and \( \sigma_i \) indicate the nodal displacement and axial member stress response, while the number of elements and degrees of freedom in the problem are denoted by \( \text{ne} \) and \( \text{nDOF} \), respectively.

![Figure 4. Boundary conditions, dimensions and loading of the 6-node truss example: \( L=9.14m, \quad P_1=P_2=445.4 \text{ KN} \) ](image)

Since the number of available sections in the list \( S_1; 42 \) is not an exact power of 2, the hidden mutation due to out-of-bound correction is associated with the binary coding for this
example. Here, the binary encoding scheme requires a genotype space of length $2^{60}$ to be searched using a 60-bit chromosome. Using DIC as an alternative encoding, the direct indexed chromosome has only 10 genes to address the $42^{10}$ genotypes; that is considerably smaller than $2^{60}$.

For comparison purposes, the 1-point crossover is applied to the binary coding as well as to the direct index coding for DVBM and AVBM. As a stopping criterion the maximum number of generations is taken as 500 in which the global optimum of this problem [28] has been achieved in the Present Work 3. Related controlling parameters of this example in the present works are given in Table 2. Table 3 demonstrates sample achieved section indices by various methods applied to this example. As can be realized the efficiency of VBDIM methods has been more than traditional binary coded simple GA meanwhile the AVBM has led to the best result within the same number of generations. Note that AVBM is a generation-based GA despite the two other methods.

Table 2. Genetic parameters for various methods applied to Example 1

<table>
<thead>
<tr>
<th>Parameter Method</th>
<th>Population Size</th>
<th>Number of generations</th>
<th>Encoding Type</th>
<th>Mutation Type</th>
<th>Mutation Probability</th>
<th>Crossover Type</th>
<th>Crossover Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Work 1</td>
<td>50</td>
<td>500</td>
<td>Binary</td>
<td>Binary</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
<tr>
<td>Present Work 2</td>
<td>50</td>
<td>500</td>
<td>DIC</td>
<td>DVBM</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
<tr>
<td>Present Work 3</td>
<td>50</td>
<td>500</td>
<td>DIC</td>
<td>AVBM</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 3. The fittest achieved layouts for Example 1.

<table>
<thead>
<tr>
<th>Member ID Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Weight (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Work 1</td>
<td>42</td>
<td>9</td>
<td>41</td>
<td>32</td>
<td>1</td>
<td>1</td>
<td>31</td>
<td>33</td>
<td>38</td>
<td>2</td>
<td>25483</td>
</tr>
<tr>
<td>Present Work 2</td>
<td>42</td>
<td>2</td>
<td>39</td>
<td>33</td>
<td>1</td>
<td>2</td>
<td>28</td>
<td>39</td>
<td>38</td>
<td>1</td>
<td>24457</td>
</tr>
<tr>
<td>Present Work 3</td>
<td>42</td>
<td>1</td>
<td>39</td>
<td>32</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td>39</td>
<td>38</td>
<td>1</td>
<td>24428</td>
</tr>
</tbody>
</table>
5.2 Example 2: Layout optimization for the 6-node planar truss

The 6-node truss of the previous example is treated here as a layout optimization benchmark [15,24]. The problem formulation and loading condition is the same as in the Example 1 but member sizes are to be selected from the section list \( S_2 \) in Table 1. Adding a zero index to denote elimination of a member, both topology and sizing is efficiently integrated in such a layout optimization problem [15]. Table 4 shows the imposed genetic controlling parameters.

Considering the results obtained in Table 5, it can be noted that the binary coding is again less efficient than both the other applied methods but this time both DVBM and AVBM have achieved the global optimum in 100 generations [15]. For structural layout optimization, a 0-index for any current chromosome allele plays the role of topological Boolean elimination bit. When such an allele is not close to the 1st index, a vast mutation band may disturb the population by entering more topologically unstable individuals. Hence, it is desirable to adaptively limit the mutation band as soon as possible, as in the case of DVBM. Note that using \( C_b \) of 0.5 in this example, resulted in an even faster convergence.

It is interesting to note that the eliminated members in this layout optimization example had been assigned the weakest section index, 1, in the previous sizing problem for the same truss example. The 0-index indicating member elimination is adjacent to the index 1 so the smooth index shifting among the sorted list of sections in this example can result in exploring the global optimum [15]. It particularly confirms the effect of mutation band confinement in the efficiency and convergence rate of the algorithm in such cases that the objective function (total structural weight) has a direct relation with a property of the design parameter (cross section area).

### Table 4. Genetic parameters for various methods applied to Example 2.

<table>
<thead>
<tr>
<th>Method</th>
<th>Population Size</th>
<th>Number of generations</th>
<th>Encoding Type</th>
<th>Mutation Type</th>
<th>Mutation Probability</th>
<th>Crossover Type</th>
<th>Crossover Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Work 4</td>
<td>100</td>
<td>100</td>
<td>Binary</td>
<td>Binary</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
<tr>
<td>Present Work 5</td>
<td>100</td>
<td>100</td>
<td>DIC</td>
<td>DVBM</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
<tr>
<td>Present Work 6</td>
<td>100</td>
<td>100</td>
<td>DIC</td>
<td>AVBM( C_b=0.5 )</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
</tbody>
</table>

### Table 5. The fittest achieved layouts for Example 2.

<table>
<thead>
<tr>
<th>Member ID</th>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Weight (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.3 Example 3: Size optimization for a 2-bay 6-story planar frame
The 2-bay 6-story frame of this example is previously treated in literature [34]; however, the section list S_3 in Table 1 is selected for the present work. Gravitational floor loads are 14.59kN/m combined with a lateral wind load of 40.05kN at each story level. Material properties are taken as $E = 207$GN/m², $\rho = 76.8$kN/m³. The allowable story drift is given by $|\Delta^s| \leq 0.0508$m and the yield stress is $F_y = 248.2$MN/m². Problem formulation for this frame example is then selected as:

Minimize

$$w = \rho \sum_{i=1}^{M} A_i L_i$$  \hspace{1cm} (14)

Subject to:
\[ KU - P = 0 \]  
\[ g^d_j = \left| \frac{\Delta_j}{\Delta_{ja}} \right| - 1 \leq 0 \]  
\[ g^s_i = \left| \frac{f_a + f_b}{F_a} \right| - 1 \leq 0, \frac{f_a}{F_a} \leq 0.15 \]  
otherwise consider maximum of
\[ g^s_i = \left| \frac{f_a + f_b}{0.6F_y} \right| - 1 \leq 0 \]  
\[ g^s_i = \left| \frac{f_a + \frac{C_m f_b}{(1 - f_a / F_a)F_b}}{F_a} \right| - 1 \leq 0 \]  
\[ \text{for } i = 1, \ldots, n_e, j = 1, \ldots, n_{DOF} \]

in which \( f_a, f_b \) are the axial and flexural member stresses. The corresponding allowable stresses \( F_a, F_b \) and other related coefficients in relations (17) to (19) are calculated based on the design code provisions [35]. In order to estimate slenderness buckling ratio, \( k \), the following relation is employed [36]:

\[ k = \max\{0.9, (1 - \frac{G_m}{20})\} \times \sqrt{1 + G_m} \]  
\[ G_m = \min\left(\frac{\sum E I_c / L_c}{\sum E I_g / L_g} \right)_{n1}, \left(\frac{\sum E I_c / L_c}{\sum E I_g / L_g} \right)_{n2} \]  

Where as \( n_1, n_2 \) denote the end nodes of each frame member and \( I_c, I_g \) are moment of inertia for each intersecting column and girder to that joint. The modified objective function and the fitness function in the mentioned examples are taken as:

\[ \varphi = w * (1 + K_p, V) \]  
\[ V = \sum_{nLc} (\max(g^d,0) + \max(g^s,0)) \]
\[ F = F_0 - \varphi \] (24)

In this example, 29 sections are available as design parameters for 14 member groups, so the search space cardinality will be \(2^{29}\) for binary coding much greater than \(29^{14}\) for direct index coding. As depicted in Table 7, the obtained result by binary coded GA is still heavier than the other applied works. Also note that in this example AVBM with the Cb of 0.5 has been less efficient but with Cb of 1 more efficient than the DVBM. Considering the envelope function curves in Figure 3, it is declared that Cb of 1 leads to more gradual decrease in the mutation band during generations.

Histories of the greatest fitness after 1st feasible elitist occurrence for this example are shown in Figures. 6-9. As can be realized the VBM methods have resulted in smoother fitness history curves than the binary coded method. The fact addresses desired effect of smartly controlling mutation band in VBM compared with undesired hidden uncontrolled mutation in binary encoding of non-binary parameters as discussed before.

### Table 6. Genetic parameters for various methods applied to Example 3

<table>
<thead>
<tr>
<th>Method</th>
<th>Population Size</th>
<th>Number of Generations</th>
<th>Encoding Type</th>
<th>Mutation Type</th>
<th>Mutation Probability</th>
<th>Crossover Type</th>
<th>Crossover Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present Work 7</td>
<td>20</td>
<td>200</td>
<td>Binary</td>
<td>Binary</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
<tr>
<td>Present Work 8</td>
<td>20</td>
<td>200</td>
<td>DIC</td>
<td>DVBM</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
<tr>
<td>Present Work 9</td>
<td>20</td>
<td>200</td>
<td>DIC</td>
<td>AVBM Cb=0.5</td>
<td>0.1</td>
<td>1-point</td>
<td>0.9</td>
</tr>
</tbody>
</table>

### Table 7. The fittest achieved layouts for Example 3

<table>
<thead>
<tr>
<th>Group</th>
<th>Member</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>Weight (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Present Work 7</td>
<td>7</td>
<td>11</td>
<td>21</td>
<td>24</td>
<td>27</td>
<td>11</td>
<td>18</td>
<td>10</td>
<td>25</td>
<td>11</td>
<td>20</td>
<td>25</td>
<td>26</td>
<td>28</td>
<td>28732</td>
</tr>
<tr>
<td></td>
<td>Present Work 8</td>
<td>9</td>
<td>17</td>
<td>18</td>
<td>24</td>
<td>23</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>17</td>
<td>21</td>
<td>27</td>
<td>23</td>
<td>10</td>
<td>24909</td>
</tr>
<tr>
<td></td>
<td>Present Work 9</td>
<td>3</td>
<td>17</td>
<td>21</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>15</td>
<td>15</td>
<td>10</td>
<td>16</td>
<td>24</td>
<td>27</td>
<td>25</td>
<td>17</td>
<td>25111</td>
</tr>
<tr>
<td></td>
<td>Present Work 10</td>
<td>9</td>
<td>17</td>
<td>21</td>
<td>19</td>
<td>23</td>
<td>13</td>
<td>16</td>
<td>2</td>
<td>21</td>
<td>10</td>
<td>21</td>
<td>24</td>
<td>25</td>
<td>29</td>
<td>23450</td>
</tr>
</tbody>
</table>
Figure 6. History of the highest fitness for the Present Work 7

Figure 7. History of the highest fitness for the Present Work 8
Figure 8. History of the highest fitness for the Present Work

Figure 9. History of the highest fitness for the Present Work
6. DISCUSSION AND CONCLUSIONS

The present study concerned modifying encoding method and mutation operator features aiming to provide improvement for genetic search in discrete spaces with respect to standard GA. It was discussed that although the binary encoding provides the largest number of schema resolution to decode non-binary phenotypes, a number of hidden and uncontrolled mutation types are associated with it such as:

1. Crossover operator on the binary coded string may also result in a hidden mutation when the bit-exchange sites are not preserved at boundaries of bit-string expansion of each parameter
2. Binary encoding often results in variable Hamming distance problem
3. Required out-of-bound correction results in an additional non-rated mutation
4. Variable mutation rate in the phenotypes landscape when the number of mutated bits in the binary expansion of various parameters is not fixed.

These reasons may undesirably disturb the balance between exploration and exploitation operators in the GA search. Thus DIC as a method of integer coding were utilized to preserve encoding correspondence and enable mutation band control. The developed strategies take advantage of gradual fitness evolving in GA as a guideline to smartly alter limit of mutation jumps and consequently achieve more efficiency in search. It was done in such a way to preserve the exploitation memory of local search regions by gradual and adaptive confinement of the mutation band around the currently evolved search points. Taking the advantage of SA strategies to derive a generation-based envelope function for mutation band decrease, the AVBM method was introduced. The results obtained in sample treated benchmarks showed positive effect of AVBM in homogenizing the fitness improvement lag meanwhile avoiding over jump of any local regions in the genotype space to avoid missing the global optimum region. To this purpose the initial population was randomly generated but smartly searched in gradual-narrowing mutation band as the GA progress. The second attempt aimed to adopt the mutation band with the problem specific features of the search space in a dynamic manner. Results showed that DVAM can reasonably converge toward the optimum even for the case of integrated topology and size optimization as an example of more sensitive and complicated search space. Although the developed algorithm of DVBM is designed for rapid convergence due to immediate mutation band reduction as soon as an improvement in the elitist fitness occur, other delayed version of it can be used to compete tuned AVBM in searching the global optimum. Both AVBM and DVBM are specialized types of VBDIM applicable to direct indexed encoding. The proposed methods deal with a fixed length but variable band chromosome by pre-assigned or adaptive band variations preserving the one-to-one encoding correspondence. From this point of view, they are competitive to some other improvement techniques such as a class of variable length chromosomes or structural section grouping applied to binary encoding. It is not only due to their simplicity and correspondence accuracy but also because of their associated dynamic access to the entire index list especially for the case of DVBM.

REFERENCES