Research Article

A Mathematical Design of Genetic Operators on $GL_n(\mathbb{Z}_2)$

Yourim Yoon¹ and Yong-Hyuk Kim²

¹ Department of Computer Engineering, College of Information Technology, Gachon University (Global Campus), 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do 461-701, Republic of Korea

² Department of Computer Science & Engineering, Kwangwoon University 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Republic of Korea

Correspondence should be addressed to Yong-Hyuk Kim; yhdflly@kw.ac.kr

Received 9 April 2013; Accepted 30 September 2013; Published 9 January 2014

Academic Editor: Yang Xu

Copyright © 2014 Y. Yoon and Y.-H. Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We study the space that consists of all nonsingular binary matrices, that is, $GL_n(\mathbb{Z}_2)$. The space is quite important in that it is used for the change of basis in binary representation, which is the encoding typically adopted in genetic algorithms. We analyze the properties of $GL_n(\mathbb{Z}_2)$ and theoretically design possible encodings and their corresponding recombination operators for evolutionary algorithms. We present approaches based on elementary matrices of linear algebra as well as typical two-dimensional ones.

1. Introduction

An $n \times n$ matrix A is called *binary* if $A \in M_{n \times n}(\mathbb{Z}_2)$ [1]. Binary matrices have been widely used to deal with the adjacency of a graph [2–5]. In particular, Anderson and Feil [2] transformed the *light bulb puzzle* into the problem of solving a linear system $A\mathbf{x} = \mathbf{b}$ from its graph structure, where $A \in M_{n \times n}(\mathbb{Z}_2)$ and $\mathbf{x}, \mathbf{b} \in \mathbb{Z}_2^n$. Then the solution could be obtained by computing the inverse of A , that is, $\mathbf{x} = A^{-1}\mathbf{b}$. Binary matrices are also useful in dealing with the *cut/cycle* subspace of a graph, which is a vector space over $\mathbb{Z}_2$ [3, 6]. They can also be used to represent a change of basis of a vector space over $\mathbb{Z}_2$ in combinatorial problems [7, 8]. A change of basis plays an important role in the performance of evolutionary algorithms [8–12]. It can be represented as a nonsingular matrix, and hence we need to consider the space composed of all nonsingular binary matrices. In this paper, we investigate the space of nonsingular binary matrices, $GL_n(\mathbb{Z}_2)$. Table 1 shows the sizes of spaces according to small n . The size of $M_{n \times n}(\mathbb{Z}_2)$ is 2^{n^2} . Those of $GL_n(\mathbb{Z}_2)$ in Table 1 are obtained by an exhaustive search over $M_{n \times n}(\mathbb{Z}_2)$. Each matrix in $M_{n \times n}(\mathbb{Z}_2)$ is verified whether it is singular or nonsingular by Gauss-Jordan elimination. $|GL_n(\mathbb{Z}_2)|$ grows exponentially as matrix size n increases.

In this paper, we focus on the mathematical designs of encodings and their corresponding recombination operators

in $GL_n(\mathbb{Z}_2)$ for evolutionary algorithms. The remainder of this paper is organized as follows. In Section 2, typical 2D encoding and recombination operators, in which we can intuitively and easily devise, are introduced. In advance of the typical 2D approaches, ones using elementary matrices of linear algebra are presented in Section 3. In Section 3.1, we introduce the elementary matrix of $M_{n \times n}(\mathbb{Z}_2)$ as a preliminary. If we represent a nonsingular binary matrix as a product of elementary matrices, the feasibility of offspring is naturally preserved. As representation based on the product of elementary matrices, we propose the representation by a variable-length linear string in Section 3.2 and that by directed acyclic graph in Section 3.3. When we consider a nonsingular matrix as a product of elementary matrices, the representation is not unique. Moreover, it is not so easy to know how many equivalent representations exist for a nonsingular binary matrix. In Section 3.4, nontrivial equivalences are investigated to help analyze the same genotypes for a phenotype. Finally, we give our conclusions in Section 4.

2. Approaches Based on 2D Encoding

The most intuitive and natural representation for a nonsingular binary matrix is 2D $n \times n$ binary encoding. Or more simply, we can use just length- n^2 binary linear string. When using linear encoding, we can adopt traditional genetic

TABLE 1: The sizes of spaces $M_{n \times n}(\mathbb{Z}_2)$ and $GL_n(\mathbb{Z}_2)$ according to matrix size n .

n	$ M_{n \times n}(\mathbb{Z}_2) $	$ GL_n(\mathbb{Z}_2) $
2	16 ($=2^4$)	6
3	512 ($=2^9$)	168
4	65,536 ($=2^{16}$)	20,160
5	33,554,432 ($=2^{25}$)	9,999,360

operators such as n -point crossover and uniform crossover for recombination operators.

However, a 2D encoding/crossover pair can reflect more geographical linkages of genes than 1D encoding/crossover pairs [13]. Cohoon and Paris [14] proposed a 2D crossover which chooses a small rectangle from one parent and copies the genes in the rectangle into the offspring, with the rest of the genes copied from the other parent. Anderson et al. [15] suggested the block-uniform crossover on $n \times n$ grid. It divides the grid into $i \times j$ blocks at random; each block of one parent is interchanged randomly with the corresponding block of the other parent based on a preassigned percentage.

Although 2D encoding can preserve more geographical relationships among the genes, when traditional straight-line-based cutting strategies are used, the power of new-schema creation is far below that of the crossovers on linear encodings [16].

Geographic crossover was suggested to resolve this problem [13, 16, 17]. In the case of a 2D encoding, it chooses a number of monotonic lines, divides the chromosomal domain into two equivalence classes, and alternately copies the genes from the two parent chromosomes. Figure 1 shows an example using geographic crossover in 2D encoding of binary matrices. By combining 2D representation and geographic crossover, we can expect both reduced information loss in the stage of encoding and the power of new-schema creation.

However, an inevitable defect of this typical 2D approach is that it needs feasibility check and repair mechanism. We can see an example case from Figure 1(c). In this case, the offspring is not invertible. One of the possible repair mechanisms is finding maximal linearly independent rows (or columns)—this can be conducted by Gauss-Jordan elimination—and replacing linearly dependent rows/columns by other available basis elements (see Figure 1(d)). Algorithm 1 shows the repair mechanism. But it is too time consuming since Gauss-Jordan elimination takes $O(n^3)$ time.

3. Approaches Based on Elementary Matrices

In this section, we propose novel methods inspired by the fact that all the invertible matrices can be represented as products of elementary matrices [18].

3.1. Elementary Matrix in $M_{n \times n}(\mathbb{Z}_2)$. Before introducing the elementary matrix of $M_{n \times n}(\mathbb{Z}_2)$, we define elementary row operation.

Definition 1. Let $A \in M_{n \times n}(\mathbb{Z}_2)$. Any one of the following two operations on the rows of A is called an elementary row operation:

- (i) interchanging any two rows of A and
- (ii) adding a row of A to another row.

Elementary row operations are of *type 1* or *type 2* depending on whether they are obtained by (i) or (ii) of Definition 1. An $n \times n$ elementary matrix over $\mathbb{Z}_2$ field is a matrix obtained by performing an elementary row operation on the identity matrix I_n . The elementary matrix is said to be of *type 1* or *type 2* according to whether the elementary row operation performed on I_n is of type 1 or type 2, respectively. We can easily check the following property.

Proposition 2. For each elementary matrix $E \in M_{n \times n}(\mathbb{Z}_2)$, $E^2 = I_n$, where I_n is the identity matrix.

Proof. It is trivial for the cases of type 1. Since $1 + 1 = 0$ and $0 + 0 = 0$ in $\mathbb{Z}_2$, $\mathbf{m} + \mathbf{m} = \mathbf{0}$ for any row vector $\mathbf{m}$. Hence, the identity also holds in the cases of type 2. $\square$

We define S_n^{ij} as the elementary matrix of type 1 that interchanges the i th row and the j th one ($i \neq j$). We also define A_n^{ij} as the elementary matrix of type 2 that adds the i th row to the j th one ($i \neq j$). For example, consider 3×3 cases; that is, $n = 3$. There are three elementary matrices of type 1, S_3^{12} , S_3^{13} , and S_3^{23} :

$$\begin{aligned} S_3^{12} &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & S_3^{13} &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \\ S_3^{23} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \end{aligned} \quad (1)$$

And there are six elementary matrices of type 2, A_3^{12} , A_3^{13} , A_3^{21} , A_3^{23} , A_3^{31} , and A_3^{32} :

$$\begin{aligned} A_3^{12} &= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & A_3^{13} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \\ A_3^{21} &= \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & A_3^{23} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \\ A_3^{31} &= \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & A_3^{32} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \quad (2)$$

Proposition 3. The number of elementary matrices in $M_{n \times n}(\mathbb{Z}_2)$ is $3n(n-1)/2$. In detail, let S_n and A_n be the sets of all elementary matrices of type 1 and of type 2 in $M_{n \times n}(\mathbb{Z}_2)$, respectively. Then, $|S_n| = n(n-1)/2$ and $|A_n| = n(n-1)$.

Proof. Since $S_n^{ij} = S_n^{ji}$ for each (i, j) pair, $S_n = \{S_n^{ij} \mid 1 \leq i < j \leq n\}$. Since $A_n^{ij} \neq A_n^{ji}$ for each (i, j) pair, $A_n = \{A_n^{ij} \mid 1 \leq i \leq n, 1 \leq j \leq n, i \neq j\}$. $\square$

```

RepairBinaryMatrix( $M = (m_{ij})$ )
{
  // Let  $m_i$  be the  $i$ th row vector of  $M$ , that is,  $M^T = (m_1 \ m_2 \ \dots \ m_n)$ .
  // Let  $I$  be the identity matrix, where  $I^T = (e_1 \ e_2 \ \dots \ e_n)$ .
   $J \leftarrow \{1, 2, \dots, n\}$ ;
   $K \leftarrow \emptyset$ ;
   $R \leftarrow \emptyset$ ;
  for each  $m_i, i \in J$  // use the process of Gauss-Jordan elimination
    if  $m_i \neq 0$  and  $m_i$  is not spanned by  $R$  then
       $R \leftarrow R \cup \{m_i\}$ ;
       $K \leftarrow K \cup \{i\}$ ;
  // Then,  $R$  becomes the set of maximal linearly-independent row vectors.
   $S \leftarrow \emptyset$ ;
  for each  $e_i, i \in J$ 
    if  $e_i$  is not spanned by  $R$  then
       $S \leftarrow S \cup \{e_i\}$ ;
  for each  $i \in J - K$ 
     $r \leftarrow$  any element in  $S$ ;
     $S \leftarrow S - \{r\}$ ;
     $m_i \leftarrow r$ ; // repairing  $m_i$ 
  return  $M = (m_{ij})$ ;
}

```

ALGORITHM 1: An algorithm for repairing offspring generated by typical 2D approach.

FIGURE 1: An example of typical 2D approach. $H(P_1, P_2) = 5$, $H(P_1, O) = 6$, and $H(P_2, O) = 3$, where H is the Hamming distance on typical 2D binary encoding. An offspring that is quite different from parents is generated by recombination operator.

It is known that every invertible matrix is represented as a product of elementary matrices [18]. Hence, if we represent a solution in $GL_n(\mathbb{Z}_2)$ as a product of elementary matrices, we get a big advantage of preserving feasibility.

3.2. Variable-Length Linear String. In the case of using elementary matrices, we can consider the representation by variable-length linear string, of which each element is an elementary matrix. Matrix P_1 in Figure 1(a) is $S_4^{12} A_4^{34} S_4^{24} A_4^{13} S_4^{14}$ when represented as a product of elementary matrices. Similarly, matrix P_2 in Figure 1(b) becomes $A_4^{34} S_4^{12} A_4^{24} S_4^{14}$. So their new encodings are given in Figure 2(a).

As recombination operators for this encoding, any recombinations for variable-length string can be used. Among such recombination operators, *geometric crossover*

[19–24] by sequence alignment [25–27] is expected to be effectively performed. The string can be regarded as a sequence having the fixed number of elementary matrices. Geometric crossover by sequence alignment will be helpful for offspring to preserve features inherent in parents.

As a distance for variable-length string, there is *edit distance* d_e : the minimum number of insertion, deletion, and replacement of single elementary matrix to transform one string into the other. The geometric crossover associated with this distance is called the *homologous geometric crossover* [25]: two strings are aligned optimally before recombination. Alignment here means allowing parent strings to be *stretched* to match better with each other. Formally stretching strings means interleaving “—” anywhere and in any number in the strings to create two stretched strings of the same length

FIGURE 2: An example of variable-length string. $H(P_1, P_2) = 5$, $(P_1, O) = 3$, and $H(P_2, O) = 2$, where H is the Hamming distance on typical 2D binary encoding. $d_e(P_1, P_2) = 5$, $d_e(P_1, O) = 2$, and $d_e(P_2, O) = 3$, where d_e is the edit distance in that insertion, deletion, and replacement have weights of 1, 1, and 2, respectively.

that have minimal Hamming distance. In our example, when we recombine $S_4^{12} A_4^{34} S_4^{24} A_4^{13} S_4^{14}$ and $A_4^{34} S_4^{12} A_4^{24} S_4^{14}$, we need to align them optimally first: $S_4^{12} A_4^{34} S_4^{24} A_4^{13} S_4^{14}$ and $-A_4^{34} S_4^{12} A_4^{24} S_4^{14}$ (see Figure 2(b)). Notice that the Hamming distance between the aligned strings is less than that between the nonaligned ones.

After the optimal alignment, one does the normal crossover and produces a new *stretched* string. The offspring is obtained by removing “—”, so by unstretching the string (see Figure 2(c)). In the example of Figure 2, we can easily check $d_e(P_1, P_2) = d_e(P_1, O) + d_e(O, P_2)$.

Now we assume that insertion, deletion, and replacement have weights of 1, 1, and 2 in edit distance d_e , respectively. Then, when $E_1 \neq E_2 \in M_{n \times n}(\mathbb{Z}_2)$, $d_e(E_1, E_2) = 2$ and $d_e(E_1 E_2, E_2 E_1) = d_e(E_1 E_2, I_n) = 2$. In general, the following proposition holds.

Proposition 4. Let A, B, C be in $GL_n(\mathbb{Z}_2)$. Then $d_e(A, B) = d_e(AC, BC) = d_e(CA, CB)$; that is, d_e is translation-invariant.

Proof. Let E be an elementary matrix in $M_{n \times n}(\mathbb{Z}_2)$. By the definition of edit distance d_e , $d_e(AE, BE) \leq d_e(A, B)$. Here the equality holds when the last elementary matrices of AE and BE are aligned. Analogously, we can obtain the inequality of $d_e(AEE, BEE) \leq d_e(AE, BE)$. By Proposition 2, $d_e(AEE, BEE) = d_e(AI_n, BI_n) = d_e(A, B)$, and then $d_e(A, B) \leq d_e(AE, BE)$. Hence, $d_e(A, B) = d_e(AE, BE)$. Since nonsingular binary matrix C can be represented by a product of elementary matrices, by induction, $d_e(A, B) = d_e(AC, BC)$. Similarly, we can show that $d_e(A, B) = d_e(CA, CB)$. $\square$

3.3. Directed Acyclic Graph. When the representation by elementary matrices is considered as a sequence of elementary matrices, the representation for a matrix may not be unique.

For example, $S_4^{12} A_4^{34} = A_4^{34} S_4^{12}$ and $S_4^{24} A_4^{13} = A_4^{13} S_4^{24}$. Moreover, it is not easy to know how many equivalent representations exist for a nonsingular matrix. However, we could find an obvious equivalence as follows.

Proposition 5. Let $C_n(i, j)$ be $\{S_n^{ij}, A_n^{ij}, A_n^{ji}\}$. Let $E_1 \in C_n(i_1, j_1)$, $E_2 \in C_n(i_2, j_2)$, and $\{i_1, j_1\} \cap \{i_2, j_2\} = \emptyset$. Then, $E_1 E_2 = E_2 E_1$.

Proof. Let $M = (m_{ij})$ be an $n \times n$ binary matrix in which m_i is the i th row vector. It is enough to show that $E_1 E_2 M = E_2 E_1 M$. $E_1 M$ changes only two row vectors m_{i_1} and m_{j_1} of M , and $E_2 M$ changes only two row vectors m_{i_2} and m_{j_2} of M . Since $\{i_1, j_1\} \cap \{i_2, j_2\} = \emptyset$ by assumption, the i_1 th and j_1 th row vectors of $E_2 M$ are the same as m_{i_1} and m_{j_1} , respectively, and the i_2 th and j_2 th row vectors of $E_1 M$ are the same as m_{i_2} and m_{j_2} , respectively. Also, $E_1 E_2 M$ does not change the i_2 th and j_2 th row vectors of $E_2 M$. Analogously, $E_2 E_1 M$ does not change the i_1 th and j_1 th row vectors of $E_1 M$. The i_1 th and j_1 th row vectors of $E_1 E_2 M$ and $E_2 E_1 M$ are the same as those of $E_1 M$, respectively, and the i_2 th and j_2 th row vectors of $E_1 E_2 M$ and $E_2 E_1 M$ are the same as those of $E_2 M$, respectively. Hence, $E_1 E_2 M = E_2 E_1 M$. $\square$

Figure 3 shows the noninterchangeability among the sets $C_4(i, j)$ s. Each set $C_4(i, j)$ has only one interchangeable set among the total 6 sets. It seems that such interchangeable cases are rare. However, it is not true for large n . For arbitrary chosen elementary matrices E_1 and E_2 , the probability of $E_1 E_2 = E_2 E_1$ is as follows:

$$P(E_1 E_2 = E_2 E_1) = \begin{cases} \frac{(n-2)(n-3)}{n(n-1)}, & \text{if } n > 3, \\ 0, & \text{if } n \leq 3. \end{cases} \quad (3)$$

FIGURE 3: Noninterchangeability of 4×4 cases ($n = 4$). Each edge means noninterchangeability between both classes.

FIGURE 4: $P(E_1E_2 = E_2E_1)$ according to matrix size n .

Figure 4 shows the probabilities of $E_1E_2 = E_2E_1$ according to matrix size n . We can see that $P(E_1E_2 = E_2E_1)$ is close to 1 for sufficiently large n . It means that the cases of $E_1E_2 = E_2E_1$ have very frequently appeared for large n .

Based on the equivalence of Proposition 5, we can represent an invertible matrix by using a directed acyclic graph (DAG) with elementary matrices as its vertices. Each edge means noninterchangeability between two elementary matrices. Consider $A = E_1E_2 \cdots E_m \in GL_n(\mathbb{Z}_2)$. We assume that the DAG structure of A is as follows:

Then, $E_1E_2 \cdots E_m$ is a topological order of the above DAG. The following is also another topological order of the above DAG

When we use this DAG representation for nonsingular binary matrices, we can apply known DAG recombination such as [28] or design geometric operators based on the similarity of DAGs [29].

(a) DAG representation of parents given in Figure 2(a)

(b) An offspring O generated by a simple DAG recombination

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

(c) Matrix form of the offspring O

FIGURE 5: An example of directed acyclic graph. $H(P_1, P_2) = 5$, $H(P_1, O) = 3$, and $H(P_2, O) = 3$, where H is the Hamming distance on typical 2D binary encoding.

Figure 5 shows the DAG representation of parents given in Figure 2 and an offspring generated by a simple DAG recombination. We can see that DAG representation reflects the similarity of parents more than that based on variable-length string.

3.4. Nontrivial Equivalences. We could also find more sophisticated equivalences. One class of them is about the exchange of two elementary matrices.

Proposition 6 (exchange rules). *For each i, j, k such that $i \neq j$, $j \neq k$, and $k \neq i$, the following five exchange rules hold. (i) $A_n^{ik} A_n^{jk} = A_n^{jk} A_n^{ik}$, (ii) $A_n^{ij} A_n^{ik} = A_n^{ik} A_n^{ij}$, (iii) $S_n^{ij} A_n^{ik} = A_n^{jk} S_n^{ij}$, (iv) $S_n^{ij} A_n^{ki} = A_n^{kj} S_n^{ij}$, and (v) $S_n^{ij} S_n^{jk} = S_n^{jk} S_n^{ij} = S_n^{ik} S_n^{ij}$.*

Proof. Let $M = (m_{ij})$ be an $n \times n$ binary matrix in which m_i is the i th row vector; that is, $M^T = (m_1 \ m_2 \ \cdots \ m_n)$.

- (i) It is enough to show that the k th row vector of $A_n^{ik} A_n^{jk} M$ is the same as that of $A_n^{jk} A_n^{ik} M$. First, consider the left side: $(A_n^{ik} M)^T = (m_1 \ \cdots \ m_{k-1} \ m_j \ + \ m_k \ m_{k+1} \ \cdots \ m_n)$. $(A_n^{ik} A_n^{jk} M)^T = (m_1 \ \cdots \ m_{k-1} \ m_i + m_j + m_k \ m_{k+1} \ \cdots \ m_n)$. Now, consider the right side: $(A_n^{jk} M)^T = (m_1 \ \cdots \ m_{k-1} \ m_i + m_k \ m_{k+1} \ \cdots \ m_n)$. $(A_n^{jk} A_n^{ik} M)^T = (m_1 \ \cdots \ m_{k-1} \ m_i + m_j + m_k \ m_{k+1} \ \cdots \ m_n)$.

- (ii) It is enough to show that the j th and k th row vectors of $A_n^{ij}A_n^{ik}M$ are the same as those of $A_n^{ik}A_n^{ij}M$. Without loss of generality, we assume that $j < k$. First, consider the left side: $(A_n^{ik}M)^T = (m_1 \cdots m_{k-1} m_i + m_k m_{k+1} \cdots m_n)$. $(A_n^{ij}A_n^{ik}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_{k-1} m_i + m_k m_{k+1} \cdots m_n)$. Now, consider the right side: $(A_n^{ij}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_n)$. $(A_n^{ik}A_n^{ij}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_{k-1} m_i + m_k m_{k+1} \cdots m_n)$.
- (iii) It is enough to show that the i th, j th, and k th row vectors of $S_n^{ij}A_n^{ik}M$ are the same as those of $A_n^{ik}S_n^{ij}M$. Without loss of generality, we assume that $i < j < k$. First, consider the left side: $(A_n^{ik}M)^T = (m_1 \cdots m_{k-1} m_i + m_k m_{k+1} \cdots m_n)$. $(S_n^{ij}A_n^{ik}M)^T = (m_1 \cdots m_{i-1} m_j m_{i+1} \cdots m_{j-1} m_i m_{j+1} \cdots m_{k-1} m_i + m_k m_{k+1} \cdots m_n)$. Now, consider the right side: $(S_n^{ij}M)^T = (m_1 \cdots m_{i-1} m_j m_{i+1} \cdots m_{j-1} m_i m_{j+1} \cdots m_n)$. $(A_n^{ik}S_n^{ij}M)^T = (m_1 \cdots m_{i-1} m_j m_{i+1} \cdots m_{j-1} m_i m_{j+1} \cdots m_{k-1} m_i + m_k m_{k+1} \cdots m_n)$.
- (iv) It is enough to show that the i th, j th, and k th row vectors of $S_n^{ij}A_n^{ki}M$ are the same as those of $A_n^{kj}S_n^{ij}M$. Without loss of generality, we assume that $i < j < k$. First, consider the left side: $(A_n^{ki}M)^T = (m_1 \cdots m_{i-1} m_i + m_k m_{i+1} \cdots m_n)$. $(S_n^{ij}A_n^{ki}M)^T = (m_1 \cdots m_{i-1} m_j m_{i+1} \cdots m_{j-1} m_i + m_k m_{j+1} \cdots m_n)$. Now, consider the right side: $(S_n^{ij}M)^T = (m_1 \cdots m_{i-1} m_j m_{i+1} \cdots m_{j-1} m_i m_{j+1} \cdots m_n)$. $(A_n^{kj}S_n^{ij}M)^T = (m_1 \cdots m_{i-1} m_i m_{j+1} \cdots m_{j-1} m_i + m_k m_{j+1} \cdots m_n)$.
- (v) It is enough to check the i th, j th, and k th row vectors of $S_n^{ij}S_n^{jk}M$, $S_n^{jk}S_n^{ik}M$, and $S_n^{ik}S_n^{ij}M$. We can easily check that the i th, j th, and k th row vectors of $S_n^{ij}S_n^{jk}M$ are m_k, m_i , and m_j , respectively. Similarly, the i th, j th, and k th row vectors of $S_n^{jk}S_n^{ik}M$ and $S_n^{ik}S_n^{ij}M$ are the same as those of $S_n^{ij}S_n^{jk}M$.

□

Another class is about the compaction of three elementary matrices into two ones.

Proposition 7 (compaction rules). *For each i, j, k such that $i \neq j, j \neq k$, and $k \neq i$, the following two compaction rules hold.* (i) $A_n^{ik}A_n^{jk}A_n^{ij} = A_n^{ij}A_n^{jk}$ and (ii) $A_n^{kj}A_n^{ki}A_n^{ji} = A_n^{ji}A_n^{kj}$.

Proof. Let $M = (m_{ij})$ be an $n \times n$ binary matrix in which m_i is the i th row vector; that is, $M^T = (m_1 \ m_2 \ \cdots \ m_n)$.

- (i) It is enough to show that the i th, j th, and k th row vectors of $A_n^{ik}A_n^{jk}A_n^{ij}M$ are the same as those of $A_n^{ij}A_n^{jk}M$. Without loss of generality, we assume that $i < j < k$. First, consider the left side: $(A_n^{ij}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_n)$.

$(A_n^{jk}A_n^{ij}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_{k-1} m_i + m_j + m_k m_{k+1} \cdots m_n)$. Since $m_i + m_i = 0$, $(A_n^{ik}A_n^{jk}A_n^{ij}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_{k-1} m_j + m_k m_{k+1} \cdots m_n)$. Now, consider the right side: $(A_n^{jk}M)^T = (m_1 \cdots m_{k-1} m_j + m_k m_{k+1} \cdots m_n)$. $(A_n^{ij}A_n^{jk}M)^T = (m_1 \cdots m_{j-1} m_i + m_j m_{j+1} \cdots m_{k-1} m_j + m_k m_{k+1} \cdots m_n)$.

- (ii) It is enough to show that the i th, j th, and k th row vectors of $A_n^{kj}A_n^{ki}A_n^{ji}M$ are the same as those of $A_n^{ji}A_n^{kj}M$. First, consider the left side: $(A_n^{ji}M)^T = (m_1 \cdots m_{i-1} m_i + m_j m_{i+1} \cdots m_n)$. $(A_n^{ki}A_n^{ji}M)^T = (m_1 \cdots m_{i-1} m_i + m_j + m_k m_{i+1} \cdots m_n)$. $(A_n^{kj}A_n^{ki}A_n^{ji}M)^T = (m_1 \cdots m_{i-1} m_i + m_j + m_k m_{i+1} \cdots m_{j-1} m_j + m_k m_{j+1} \cdots m_n)$. Now, consider the right side: $(A_n^{kj}M)^T = (m_1 \cdots m_{j-1} m_j + m_k m_{j+1} \cdots m_n)$. $(A_n^{ji}A_n^{kj}M)^T = (m_1 \cdots m_{i-1} m_i + m_j + m_k m_{i+1} m_{j-1} m_j + m_k m_{j+1} \cdots m_n)$.

□

The encodings of matrices P_1 and P_2 given in Figure 2 can be reduced to other forms by using the presented propositions as follows:

$$\begin{aligned}
 P_1 &= S_4^{12} A_4^{34} S_4^{24} A_4^{13} S_4^{14} \\
 &= \underline{A_4^{34} S_4^{12}} S_4^{24} A_4^{13} S_4^{14} \quad (\because \text{Proposition 5}) \\
 &= A_4^{34} S_4^{12} \underline{A_4^{13} S_4^{24}} S_4^{14} \quad (\because \text{Proposition 5}) \\
 &= A_4^{34} \underline{A_4^{23} S_4^{12}} S_4^{24} S_4^{14} \quad (\because \text{Proposition 6 (iii)}) \\
 &= A_4^{34} A_4^{23} \underline{S_4^{24} S_4^{14}} S_4^{14} \quad (\because \text{Proposition 6 (v)}) \\
 &= A_4^{34} A_4^{23} S_4^{24} \underline{E} \quad (\because \text{Proposition 2}) \\
 &= A_4^{34} A_4^{23} S_4^{24},
 \end{aligned} \tag{4}$$

$$\begin{aligned}
 P_2 &= A_4^{34} S_4^{12} A_4^{24} S_4^{14} \\
 &= A_4^{34} S_4^{12} \underline{S_4^{14} A_4^{21}} \quad (\because \text{Proposition 6 (iv)}) \\
 &= A_4^{34} \underline{S_4^{14} S_4^{24}} A_4^{21} \quad (\because \text{Proposition 6 (v)}) \\
 &= A_4^{34} S_4^{14} \underline{S_4^{24}} A_4^{21} \quad (\because S_n^{ij} = S_n^{ji} \text{ by the definition of } S_n^{ij}) \\
 &= A_4^{34} S_4^{14} S_4^{24} A_4^{21}.
 \end{aligned} \tag{5}$$

From the reduction, the edit distance between both parents decreases by 2. This reduction process can be understood as a kind of normalization [23]. Figure 6 shows a recombination example with the reduced encodings of parents. It shows that reduced encodings can produce more similar offspring to parents.

FIGURE 6: Another example of variable-length string. $H(P_1, P_2) = 5$, $H(P_1, O) = 1$, and $H(P_2, O) = 4$, where H is the Hamming distance on typical 2D binary encoding $d_e(P_1, P_2) = 3$, $d_e(P_1, O) = 1$, and $d_e(P_2, O) = 2$, where d_e is the edit distance in that insertion, deletion, and replacement have weights of 1, 1, and 2, respectively.

4. Conclusions

We presented several ideas to deal with the space of nonsingular binary matrices, $GL_n(\mathbb{Z}_2)$, in evolutionary algorithms. This space is important in that it is used for the change of basis of standard binary encoding. As typical approaches, we represented a matrix in $GL_n(\mathbb{Z}_2)$ just as an array of matrix elements. Two-dimensional $n \times n$ binary encoding or length- n^2 binary linear string can be used in these approaches. However, these approaches need the feasibility check of offspring together with their repair mechanism. We illustrated this nature by an example.

We presented some novel methods using elementary matrices, which have an advantage of preserving feasibility. An invertible matrix can be represented as a product of elementary matrices. Variable-length linear string and DAG (directed acyclic graph) representation are proposed to represent each nonsingular binary matrix that could be understood as a product of elementary matrices. Different products of elementary matrices often represent the same matrix, so we tried to get more information about the relation between representations with the same phenotype by finding some nontrivial equivalences.

Future work will be directed to apply these results to the problem of finding good bases of binary encoding in evolutionary algorithms.

Disclosure

A preliminary version of this paper appeared in the Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 855–858, 2009.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] Y.-H. Kim and K. Seo, "Two congruence classes for symmetric binary matrices over $\mathbb{F}_2$," *WSEAS Transactions on Mathematics*, vol. 7, no. 6, pp. 339–343, 2008.
- [2] M. Anderson and T. Feil, "Turning lights out with linear algebra," *Mathematics Magazine*, vol. 71, no. 4, pp. 300–303, 1998.
- [3] N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, Cambridge, UK, 2nd edition, 1994.
- [4] R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, Cambridge, UK, 1991.
- [5] Y.-H. Kim and B.-R. Moon, "New topologies for genetic search space," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1393–1399, New York, NY, USA, June 2005.
- [6] Y. Yoon, Y.-H. Kim, and B.-R. Moon, "A note on edge-based graph partitioning and its linear algebraic structure," *Journal of Mathematical Modelling and Algorithms*, vol. 10, no. 3, pp. 269–276, 2011.
- [7] Y.-H. Kim, "Linear transformation in pseudo-boolean functions," in *Proceedings of the 10th Annual Genetic and Evolutionary Computation Conference*, pp. 1117–1118, July 2008.
- [8] Y.-H. Kim and Y. Yoon, "Effect of changing the basis in genetic algorithms using binary encoding," *KSII Transactions on Internet and Information Systems*, vol. 2, no. 4, pp. 184–193, 2008.
- [9] I. Hwang, Y.-H. Kim, and B.-R. Moon, "Multi-attractor gene reordering for graph bisection," in *Proceedings of the 8th Annual*

- Genetic and Evolutionary Computation Conference*, pp. 1209–1215, July 2006.
- [10] Y.-H. Kim, Y.-K. Kwon, and B.-R. Moon, "Problem-independent schema synthesis for genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1112–1122, 2003.
 - [11] K. Seo, S. Hyun, and Y.-H. Kim, "A spanning tree-based encoding of the max cut problem for evolutionary search," in *Proceedings of the International Conference on Parallel Problem Solving From Nature*, vol. 7491 of *Lecture Notes in Computer Science*, pp. 510–518, 2012.
 - [12] D. Wyatt and H. Lipson, "Finding building blocks through eigenstructure adaptation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1518–1529, 2003.
 - [13] B.-R. Moon, Y.-S. Lee, and C.-K. Kim, "GEORG: VLSI circuit partitioner with a new genetic algorithm framework," *Journal of Intelligent Manufacturing*, vol. 9, no. 5, pp. 401–412, 1998.
 - [14] J. P. Cohoon and W. Paris, "Genetic placement," in *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 422–425, 1986.
 - [15] C. A. Anderson, K. F. Jones, and J. Ryan, "A two-dimensional genetic algorithm for the Ising problem," *Complex Systems*, vol. 5, pp. 327–333, 1991.
 - [16] A. B. Kahng and B.-R. Moon, "Toward more powerful recombinations," in *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 96–103, 1995.
 - [17] E.-J. Park, Y.-H. Kim, and B.-R. Moon, "Genetic search for fixed channel assignment problem with limited bandwidth," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1172–1179, 2002.
 - [18] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 1997.
 - [19] A. Moraglio, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Geometric crossovers for multiway graph partitioning," *Evolutionary Computation*, vol. 15, no. 4, pp. 445–474, 2007.
 - [20] A. Moraglio and R. Poli, "Topological interpretation of crossover," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1377–1388, 2004.
 - [21] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, 2013.
 - [22] Y. Yoon and Y.-H. Kim, "Geometricity of genetic operators for real-coded representation," *Applied Mathematics and Computation*, vol. 219, no. 23, pp. 10915–10927, 2013.
 - [23] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "Quotient geometric crossovers and redundant encodings," *Theoretical Computer Science*, vol. 425, pp. 4–16, 2012.
 - [24] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "A theoretical and empirical study on unbiased boundary-extended crossover for real-valued representation," *Information Sciences*, vol. 183, no. 1, pp. 48–65, 2012.
 - [25] A. Moraglio, R. Poli, and R. Seehuus, "Geometric crossover for biological sequences," in *Proceedings of the European Conference on Genetic Programming*, pp. 121–132, 2006.
 - [26] Y. Yoon, Y.-H. Kim, A. Moraglio, and B. R. Moon, "Mathematical interpretation between genotype and phenotype spaces and induced geometric crossovers," Technical Report Abs/0907.3202, CoRR, Ithaca, NY, USA, 2009.
 - [27] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "A mathematical unification of geometric crossovers defined on phenotype space," Technical Report abs/0907.3200, CoRR, Ithaca, NY, USA, 2009.
 - [28] C. Cotta and J. M. Troya, "Analyzing directed acyclic graph recombination," in *Proceedings of the 7th International Conference on Fuzzy Days on Computational Intelligence, Theory and Applications*, pp. 739–748, 2001.
 - [29] J. Jin, *Similarity of weighted directed acyclic graphs [M.S. thesis]*, The University of New Brunswick, New Brunswick, Canada, 2006.