Hindawi
Computational and Mathematical Methods
Volume 2023, Article ID 5951901, 8 pages
https://doi.org/10.1155/2023/5951901

WILEY | Hindawi

Research Article

Genetic Algorithm-Based Method for Discovering Involutory MDS Matrices

El Mehdi Bellfkih,[1] **Said Nouh**,[2] **Imrane Chems Eddine Idrissi**,[2] **Khalid Louartiti**,[3] **and Jamal Mouline**[1]

[1]*LAMS, Hassan II University, Casablanca 20700, Morocco*
[2]*LTIM, Hassan II University, Casablanca 20700, Morocco*
[3]*SMAD, Abdelmalek Essaadi University, Tetouan 93000, Morocco*

Correspondence should be addressed to El Mehdi Bellfkih; elmehdi.bellfkih@gmail.com

Received 22 September 2023; Revised 28 November 2023; Accepted 18 December 2023; Published 30 December 2023

Academic Editor: Pedro Alonso

In this paper, we present an innovative approach for the discovery of involutory maximum distance separable (MDS) matrices over finite fields $\mathbb{F}_{2^q}$, derived from MDS self-dual codes, by employing a technique based on genetic algorithms. The significance of involutory MDS matrices lies in their unique properties, making them valuable in various applications, particularly in coding theory and cryptography. We propose a genetic algorithm-based method that efficiently searches for involutory MDS matrices, ensuring their self-duality and maximization of distances between code words. By leveraging the genetic algorithm's ability to evolve solutions over generations, our approach automates the process of identifying optimal involutory MDS matrices. Through comprehensive experiments, we demonstrate the effectiveness of our method and also unveil essential insights into automorphism groups within MDS self-dual codes. These findings hold promise for practical applications and extend the horizons of knowledge in both coding theory and cryptographic systems.

1. Introduction and Preliminaries

Error correction is a critical aspect of various fields, including telecommunications, data storage, and digital communication. In these domains, ensuring the integrity and accuracy of transmitted or stored information is of paramount importance. Errors can occur due to noise, interference, or other factors, potentially leading to data corruption or loss. To address this challenge, error correction techniques are employed to detect and correct errors, enhancing the reliability and performance of systems.

One widely used approach for error correction is based on encoding data using polynomials and matrices over finite fields. Finite fields provide a mathematical framework for representing and manipulating data elements in a structured manner. The encoding process involves mapping the original data into a set of symbols, which are then transformed into polynomials or matrices. These encoded representations incorporate redundancy, enabling the detection and correction of errors during the decoding phase.

Finding suitable matrices is essential for error correction codes. Specifically, maximum distance separable (MDS) matrices play a crucial role in achieving maximum error correction capability [1–3]. MDS codes can correct the maximum number of errors possible for a given code length, making them highly desirable for error-prone environments [1]. Additionally, MDS matrices facilitate efficient error correction algorithms and reduce complexity in error correction procedures, thus enhancing overall system performance.

Moreover, special matrices, such as involutory matrices, have gained attention for their unique properties. Involutory matrices have their own inverses, simplifying the decoding process and reducing computational overhead. The search for MDS involutory matrices is particularly significant, as it allows for efficient error correction with fewer computational resources, making them valuable in resource-constrained

applications and enhancing the overall reliability and security of the system [4]. By leveraging special matrices, researchers can achieve a delicate balance between error correction capability, reliability, and performance, enabling the development of robust and efficient error correction techniques for various real-world applications.

Maximum distance separable (MDS) matrices not only find applications in coding theory but also play a crucial role in the design of block ciphers and hash functions [5, 6]. Their unique properties, including full rank and nonsingularity, make them essential for achieving error correction and data integrity. However, finding MDS matrices is a highly nontrivial task due to the stringent conditions they must satisfy. In recent years, various techniques have been explored to efficiently discover MDS matrices with desired properties, paving the way for enhanced reliability, security, and robustness of both communication and cryptographic systems [7–9]. The construction of MDS matrices, including involutory MDS matrices over $\mathbb{F}_{p^n}$, uses self-dual codes [10] or often involves utilizing specific matrices with desirable properties. Companion matrices, Hadamard matrices, Cauchy matrices, and Vandermonde matrices, along with the inverse of another Vandermonde matrix, are among the key matrices used for this purpose [5, 11–15]. Hadamard matrices possess orthogonal properties, making them valuable for constructing MDS matrices that aid in error correction and data integrity. The Cauchy matrices, on the other hand, are essential in constructing MDS matrices with a high degree of redundancy, contributing to enhanced fault tolerance. Additionally, the Vandermonde matrices and their inverses play a pivotal role in generating involutory MDS matrices, ensuring that these matrices maintain their properties even after squaring to the identity matrix. By leveraging these particular matrices, researchers have been able to develop efficient and reliable methods for constructing MDS and involutory MDS matrices.

Let $\mathbb{F}_p$ be the field of p elements and $\mathbb{F}_p[X]$ be the polynomial ring with coefficients in $\mathbb{F}_p$. We denote by $\mathcal{M}_{n,n}(\mathbb{F}_p)$ squared matrices with coefficients in $\mathbb{F}_p$.

A code $\mathscr{C}$ of dimension n and length $2n$ over $\mathbb{F}_p$ is a subspace $\mathbb{F}_p^{2n}$. Moreover, if $\mathscr{C}$ is a subspace of $\mathbb{F}_p^{2n}$, then the code $\mathscr{C}$ is said to be linear code; in this case, the Hamming distance between two vectors x and y in $\mathbb{F}_p^{2n}$ is defined as follows:

$$d_H(x, y) = |\{i | x_i \neq y_i\}|. \tag{1}$$

The minimum distance of a code $\mathscr{C}$ is defined as the smallest Hamming distance between any two distinct elements within the code $\mathscr{C}$, denoted as $d(\mathscr{C})$.

Let $\mathscr{C}$ be a $[n, k, d]$ linear code over $\mathbb{F}_p$. Then, $\mathscr{C}$ is MDS if its d meets the singleton bound:

$$d \leq n - k + 1. \tag{2}$$

The dual code of $\mathscr{C} \subset \mathbb{F}_p^n$ is defined by

$$\mathscr{C}^{\perp} = \left\{ \mathrm{x} \in \mathbb{F}_p^n | <\mathrm{x}, \mathrm{c}> = 0 \forall \mathrm{c} \in \mathscr{C} \right\}, \tag{3}$$

where $<x, c> = \sum_{i=1}^{n} x_i c_i$.

Definition 1. $\mathscr{C}$ is self-dual code if $\mathscr{C} = \mathscr{C}^{\perp}$.

Definition 2. Matrix $M \in \mathcal{M}_{n,n}(\mathbb{F}_p)$ is MDS if and only if all its minors are nonzero.

Definition 3. Another definition using the fact that $\mathscr{C}$ is an MDS codes is that matrix $M \in \mathcal{M}_{n,n}(\mathbb{F}_p)$ is MDS if and only if the generator matrix of $\mathscr{C}$ is equal to $(I|M)$.

Definition 4. Matrix $M \in \mathcal{M}_{n,n}(\mathbb{F}_p)$ is involutory MDS if M is MDS and $M^2 = I_n$.

Definition 5. $M \in \mathcal{M}_{\mathrm{n,n}}(\mathbb{F}_p)$ is orthogonal if $MM^{\intercal} = M^{\intercal}M = I_n$.

Proposition 6. *Let $\mathscr{C}$ be an MDS self-dual code over $\mathbb{F}_p$ of dimension n and length 2n and generator matrix $G = (I_n|M)$; then, M is an orthogonal MDS matrix.*

Proof. Since $G = (I_n|M)$ is a generator matrix of MDS self-dual code $\mathscr{C}$, then

$$\begin{aligned} GG^{\intercal} = 0 = [I_n \quad M]. \begin{bmatrix} I_n \\ M^{\intercal} \end{bmatrix} = [I_n + MM^{\intercal}] \\ = 0 \Longrightarrow MM^{\intercal} = M^{\intercal}M = I. \end{aligned} \tag{4}$$

So, M is an orthogonal matrix. Moreover, since $\mathscr{C}$ is MDS, then M is orthogonal MDS. □

For our purpose, we only consider square MDS matrices. These matrices are redundant part of MDS self-dual codes of dimension n and length $2n$ over $\mathbb{F}_p$ where $p = 2^m$.

Let $\sigma \in \mathcal{S}_n$ be a permutation; P_σ is a permutation matrix related to σ defined as follows:

$$P_\sigma = \sigma(I_n). \tag{5}$$

Also, $N_\sigma = DP_\sigma$ is monomial matrix, where P_σ is a permutation matrix and D is a matrix such that $D = \mathrm{diag}\,(\alpha_1, \alpha_2, \cdots, \alpha_n)$. As shown in [5], multiplying by a monomial matrix preserves the invariant properties of an MDS matrix. This means that if M is an MDS matrix, $N_{\sigma_i} i \in \{1, 2\}$ are monomial matrices. Then, $N_{\sigma_1} M N_{\sigma_2}$ is an MDS matrix.

2. Genetic Algorithm

The genetic algorithm is a computational search heuristic, drawing inspiration from Charles Darwin's theory of natural evolution [16]. This algorithm simulates the process of natural selection, wherein individuals exhibiting higher fitness levels are chosen for reproduction, thereby generating offspring for the subsequent generation. By mimicking the principles of natural selection, the genetic algorithm seeks to efficiently optimize solutions to complex problems through iterative improvement and selection mechanisms [17–19].

The genetic algorithm addresses the issue of permutation in combinatorial optimization problems by efficiently exploring the search space. It achieves this by employing selection, crossover, and mutation operators, which contribute to the generation of better chromosomes at minimal cost [11]. Empirical studies have demonstrated the efficacy of evolutionary algorithms, including GA, in tackling various combinatorial optimization challenges [17].

GA offers several advantages over traditional algorithms:

It relies solely on the objective function's evaluation, irrespective of its characteristics (e.g., continuity, and differentiability), providing greater flexibility and applicability across diverse problem domains [20, 21].

The generation process in GA operates in a parallel manner, allowing for simultaneous exploration of multiple points, in contrast to standard algorithms that typically involve single iterations.

Probabilistic transition rules, involving selection, crossover, and mutation probabilities, are employed in GA, offering stochastic and dynamic decision-making capabilities rather than deterministic approaches.

Overall, the utilization of GA and similar nature-inspired algorithms presents a promising avenue for efficiently addressing complex optimization problems with diverse applications. It may suffer from slow convergence and may not always find the global optimum due to their stochastic nature.

The algorithm begins with an initial population of potential solutions, where fitter individuals are selected based on a fitness function. These selected individuals then undergo crossover and mutation operations, which mimic the inheritance and variation mechanisms in natural evolution. The process iterates, generating new generations with increasingly fit individuals until a satisfactory solution is obtained. The five key phases of the GA include the following:

(1) The initial population setup

(2) Defining the fitness function

(3) Selection of fitter individuals

(4) Applying crossover to create offspring

(5) Introducing mutation for genetic diversity

By following these phases, the genetic algorithm effectively explores the search space to find optimal or near-optimal solutions to the given problem. In our specific case, the optimal solution corresponds to the exact solution (or solutions), and there are no near-optimal solutions as the objective is to precisely identify the best possible outcome within the given problem domain.

3. Proposed Method

In this section, we present a comprehensive explanation of the method employed in this paper, which is based on the genetic algorithm. Our aim is to provide a detailed account of the GA's underlying mechanisms and operations, thereby offering a clear understanding of how it functions to attain optimal solutions. Specifically, we will delve into the intricacies of its key operators, namely, the selection, crossover, and mutation operations. Through this detailed exposition, we seek to demonstrate the efficiency and efficacy of our GA-based approach in addressing the specific optimization problem under investigation, while emphasizing our focus on achieving exact optimal solutions which is the involutory MDS matrices.

3.1. The Search Space and the Fitness Function. The problem-solving process commences with the establishment of a population, consisting of a collection of individuals. Each individual represents a potential solution to the problem at hand and is defined by a unique set of parameters, referred to as genes (see Figure 1). To form a complete solution, these genes are combined into a string structure known as a chromosome. In the genetic algorithm framework, he Genes of an individual are typically represented using a list of genes, often employing binary values. However, in our case, they are represented as a list of integers ranging from 1 to the dimension of the code (the number of rows in a matrix). This process of encoding the genes within a chromosome allows for efficient handling and manipulation of the solutions during the evolutionary search, enabling the algorithm to explore and refine a diverse set of potential solutions over successive generations.

In our study, the search space is tied to the size n, representing the number of matrix M columns (or rows). The search space consists of all possible pairs of permutations, amounting to $(n!)^2$ pairs of permutations, involving the rows or columns of matrix M. However, the ultimate solutions are derived from the action of permutations on the arrangement of M's rows or columns. To navigate this extensive search space effectively, we employ a fitness function that serves to evaluate candidate permutations. The fitness function assesses the optimality of each pair of permutations based on specific optimization criteria. During the selection process, the fittest candidates, those with higher fitness scores based on fitness function value (see Equation (6)) are identified by their lower values according to the fitness function, which are favored to advance to the next generation of the genetic algorithm. In our approach, pairs of permutations (chromosomes) are represented as lists of integers from 1 to n for each permutation. These permutations are related to their permutation matrices, which in turn are randomly applied to the rows or columns of matrix M. This randomness ensures exploration of the search space to find potentially optimal solutions (pairs of permutations).

Let $M \in \mathscr{M}_{(n\times n)}(\mathbb{F}_{2^m})$ be an MDS matrix, $D = d_{ij} = M^2$.

$$f_M = \sum_{i=1}^{n} f_i + \sum_{j=1}^{n} \sum_{\substack{k=1 \\ k\neq j}}^{n} f_{i,j}, \tag{6}$$

where

$$f_i = \begin{cases} 1, & \text{if } d_{i,i} \neq 1, \\ 0, & \text{otherwise,} \end{cases}$$
$$f_{i,j} = \begin{cases} 1, & \text{if } d_{i,i} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

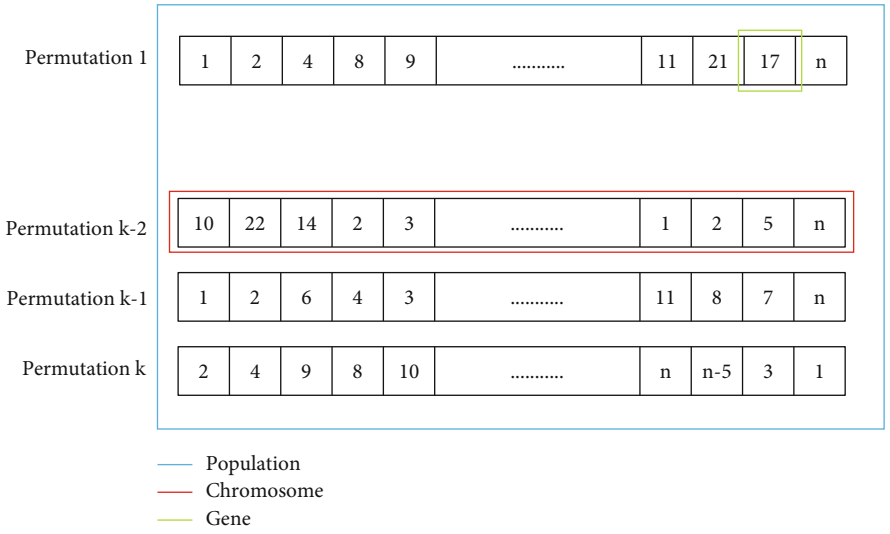

Figure 1: Population, chromosomes, and genes.

We are adopting the Hamming distance as the measure of dissimilarity between any two elements.

3.2. The Selection, the Crossover, and the Mutation. The proposed genetic algorithm-based method is depicted in the diagram (Figure 2), taking several inputs such as the number of generations, initial population size, crossover rate, mutation rate, and the fitness function. The selection process employs elitism, where individuals are chosen based on their fitness values as determined by equation (6). Crossover and mutation operations are illustrated in Figure 3, detailing their implementation steps within the algorithm.

Figure 3 presents a comprehensive schema illustrating the crossover and mutation operators employed in the genetic algorithm. For the crossover operation, a single parent is selected, and subsequently, three positions are randomly chosen within the permutation set, denoted as acting on the left of the matrix. Similarly, the same process is applied to the permutation set that acts on the right of the matrix. This procedure facilitates the exploration of pairs of permutations that lead us to minimize the fitness function, thereby ensuring a thorough examination of potential solutions. Following the crossover, the mutation operator is implemented, involving the swapping of two genes within each permutation. This step introduces further diversity and exploration in the search process, enabling the algorithm to converge towards more optimal solutions. The inherent property of MDS matrices, where permuting rows or columns does not compromise their MDS characteristics, extends to our algorithm. Consequently, operations such as crossover and mutation (involving the swapping of two genes within the permutation) also uphold the MDS property. The combination of these operators contributes to the efficacy and robustness of the genetic algorithm in efficiently navigating through the search space and identifying MDS involutory matrices with improved characteristics.

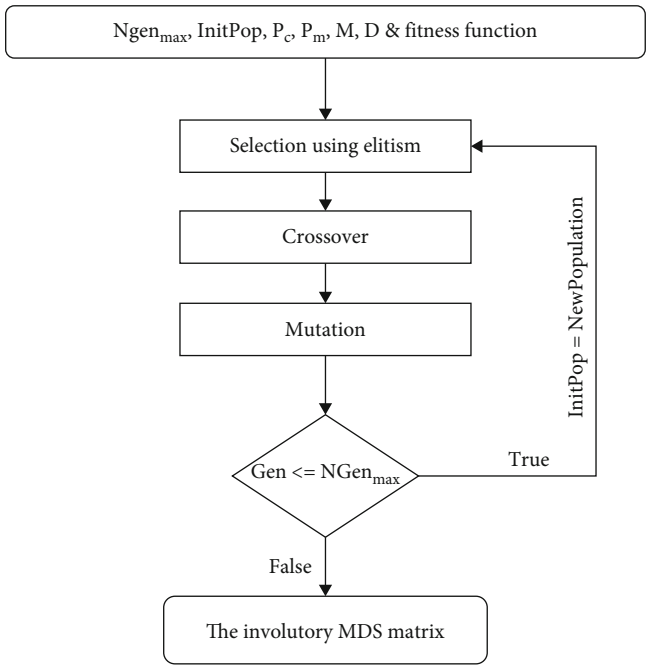

Figure 2: Genetic algorithm-based method.

4. Results and Discussion

The pursuit of self-dual MDS codes presents a challenging task, given the inherent complexities associated with their construction. Furthermore, the creation of MDS matrices over finite fields of characteristic 2 is no straightforward endeavor. Therefore, our approach primarily focuses on smaller code sets, where we endeavor to extract specific properties. These properties serve as foundational elements for the utilization of our algorithm in the identification of involutory MDS matrices. The application of our method

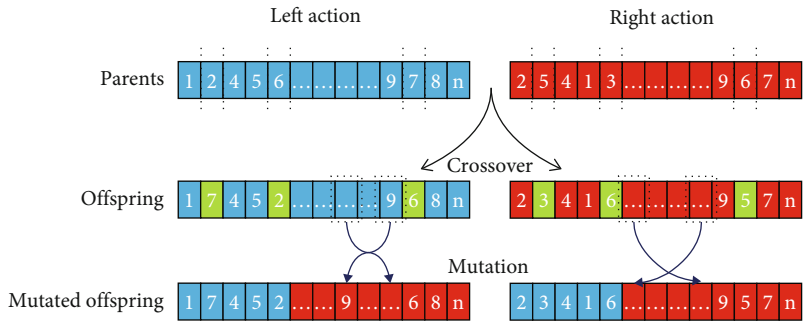

Figure 3: Crossover and mutation operators.

holds significant value, as it enables the derivation of crucial matrices. These matrices, in turn, contribute to the establishment of an essential automorphism group. This development opens doors to important applications that extend beyond the realm of error correction coding, encompassing broader domains where automorphisms play a pivotal role. To execute the method, the default parameters mentioned in Table 1 are used.

Our initial population is comprised of permutations of length n, where the product of $n \times n$ represents the dimension of the MDS matrix. These permutations act upon the rows (left action) and columns (right action) of the MDS matrix iteratively, with the objective of identifying a pair of permutations that satisfy a specific condition, leading to the transformation of the MDS matrix into an involutory MDS matrix.

The size of our initial population is contingent upon the search space, which is equal to $(n!)^2$). In our particular case, given the relatively small size of the matrix, we have opted for an initial population size of 12 individuals. Also, a high crossover rate promotes diversity and accelerates convergence in our algorithm, while a low mutation rate maintains diversity and prevents local solution trapping. As a selection method, we employ elitism by selecting the top 6 chromosomes. This balance optimizes solution exploration. The solution involves finding pairs of permutations that lead to an involutory property in MDS matrix. By employing these predefined parameter values, the genetic algorithm-based method can efficiently explore the search space, find potential solutions, and converge towards an optimal the given problem.

In the context of MDS codes, the generator matrix G may not always be in systematic form initially. However, through the application of the Gauss-Jordan elimination process, it can be transformed into systematic form. In our specific case, we focus on working with generator matrices that are already in systematic form. This form facilitates the representation of the code with easily identifiable systematic components.

Example 1. Let α be a root of $p(x) = x^4 + x + 1 \in \mathbb{F}_2[X]$, where α is a primitive element in $\mathbb{F}_{2^4}$, and let G_1 be the following generator matrix of an MDS code of dimension 3 and length 6 over $\mathbb{F}_{2^4}$ [8]:

Table 1: Default parameters.

Parameter	Value
Initial population size	12
Selection	Elitism
Crossover rate	0.97
Mutation rate	0.02
Number of generations	8

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & \alpha^3 & 1+\alpha^2 & \alpha^2+\alpha^3 \\ 0 & 1 & 0 & \alpha & \alpha^3 & 1+\alpha+\alpha^3 \\ 0 & 0 & 1 & 1+\alpha+\alpha^3 & \alpha^2+\alpha^3 & \alpha+\alpha^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \alpha^3 & \alpha^8 & \alpha^6 \\ 0 & 1 & 0 & \alpha & \alpha^3 & \alpha^7 \\ 0 & 0 & 1 & \alpha^7 & \alpha^6 & \alpha^5 \end{pmatrix} = (I_3|M). \tag{8}$$

Our MDS matrix M is

$$M = \begin{pmatrix} \alpha^3 & \alpha^8 & \alpha^6 \\ \alpha & \alpha^3 & \alpha^7 \\ \alpha^7 & \alpha^6 & \alpha^5 \end{pmatrix}. \tag{9}$$

We can easily check that our MDS matrix M is an orthogonal MDS matrix; also, M is not an involutory MDS matrix.

After using the algorithm, we get the following matrix using the pair of permutations ((),(1,2)):

$$M' = \begin{pmatrix} \alpha^8 & \alpha^3 & \alpha^6 \\ \alpha^3 & \alpha & \alpha^7 \\ \alpha^6 & \alpha^7 & \alpha^5 \end{pmatrix}. \tag{10}$$

M' is the involutory MDS matrix.

Example 2. Let α be a root of $p(x) = x^4 + x + 1 \in \mathbb{F}_2[X]$, where α is a primitive element in $\mathbb{F}_{2^4}$, and let G_2 be the following

generator matrix of an MDS code of dimension 4 and length 8 over $\mathbb{F}_{2^4}$:

$$G_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & \alpha+\alpha^3 & \alpha^2+\alpha^3 & 1 & \alpha+\alpha^2 \\ 0 & 1 & 0 & 0 & 1 & \alpha+\alpha^2 & \alpha+\alpha^3 & \alpha^2+\alpha^3 \\ 0 & 0 & 1 & 0 & \alpha+\alpha^2 & 1 & \alpha^2+\alpha^3 & \alpha+\alpha^3 \\ 0 & 0 & 0 & 1 & \alpha^2+\alpha^3 & \alpha+\alpha^3 & \alpha+\alpha^2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \alpha^9 & \alpha^6 & 1 & \alpha^5 \\ 0 & 1 & 0 & 0 & 1 & \alpha^5 & \alpha^9 & \alpha^6 \\ 0 & 0 & 1 & 0 & \alpha^5 & 1 & \alpha^6 & \alpha^9 \\ 0 & 0 & 0 & 1 & \alpha^6 & \alpha^9 & \alpha^5 & 1 \end{pmatrix} = (I_4|M). \tag{11}$$

For this example, our MDS matrix M is

$$M = \begin{pmatrix} \alpha^9 & \alpha^6 & 1 & \alpha^5 \\ 1 & \alpha^5 & \alpha^9 & \alpha^6 \\ \alpha^5 & 1 & \alpha^6 & \alpha^9 \\ \alpha^6 & \alpha^9 & \alpha^5 & \alpha^9 \end{pmatrix}. \tag{12}$$

After using the same algorithm, we get the following matrix using the pair of permutations ((24), (34)):

$$M' = \begin{pmatrix} \alpha^9 & \alpha^6 & \alpha^5 & 1 \\ \alpha^6 & \alpha^9 & 1 & \alpha^5 \\ \alpha^5 & 1 & \alpha^9 & \alpha^6 \\ 1 & \alpha^5 & \alpha^6 & \alpha^9 \end{pmatrix}. \tag{13}$$

M' is the involutory MDS matrix.

Example 3. Let α be a root of $p(x) = x^3 + x + 1 \in \mathbb{F}_2[X]$, where α is a primitive element in $\mathbb{F}_{2^3}$, and let G_3 be the following generator matrix of an MDS code of dimension 3 and length 6 over $\mathbb{F}_{2^3}$ [14]:

$$G_3 = \begin{pmatrix} 1 & 0 & 0 & 1+\alpha+\alpha^2 & 1+\alpha^2 & 1+\alpha \\ 0 & 1 & 0 & 1+\alpha^2 & 1+\alpha & 1+\alpha+\alpha^2 \\ 0 & 0 & 1 & 1+\alpha & 1+\alpha+\alpha^2 & 1+\alpha^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \alpha^5 & \alpha^6 & \alpha^3 \\ 0 & 1 & 0 & \alpha^6 & \alpha^3 & \alpha^5 \\ 0 & 0 & 1 & \alpha^3 & \alpha^5 & \alpha^6 \end{pmatrix} = (I_3|M). \tag{14}$$

For this example, our MDS matrix M is

$$M = \begin{pmatrix} \alpha^5 & \alpha^6 & \alpha^3 \\ \alpha^6 & \alpha^3 & \alpha^5 \\ \alpha^3 & \alpha^5 & \alpha^6 \end{pmatrix}. \tag{15}$$

After using the same algorithm, we get the following matrix using the pair of permutations ((23), (23)):

$$M' = \begin{pmatrix} \alpha^5 & \alpha^3 & \alpha^6 \\ \alpha^3 & \alpha^6 & \alpha^5 \\ \alpha^6 & \alpha^5 & \alpha^3 \end{pmatrix}. \tag{16}$$

M' is the involutory MDS matrix.

It is important to note that the results obtained from the search for MDS involutory matrices using generator matrices of self-dual MDS codes and genetic algorithm-based methods are not unique. The nature of the genetic algorithm introduces an element of randomness in the selection, crossover, and mutation processes, leading to different solutions in each run. As a result, multiple valid involutory MDS matrices may be discovered, all satisfying the desired criteria. The non-uniqueness of solutions highlights the diversity and flexibility of the genetic algorithm in exploring the search space and presenting a variety of feasible solutions for the given problem, which may be advantageous in practical applications.

5. Automorphism Group of Some MDS Codes

By employing the Jordan-Gauss elimination technique, we transform the generator matrix into a systematic form $(I|M')$, with M' being an MDS matrix. Subsequently, our genetic algorithms facilitate the derivation of an involutory MDS matrix M from an existing MDS matrix. Let $\mathscr{C}$ be $[2n, n]$ an MDS code of generator matrix G in systematic form and $\mathrm{GL}_n(2^q)$ the general linear group of size n over $\mathbb{F}_{2^q}$; we consider this mapping ψ which is also a group homomorphism defined by

$$\begin{aligned} \psi &: \mathscr{A}ut(\mathscr{C}) \longrightarrow \mathrm{GL}_n(2^q), \\ \sigma &\longrightarrow D \text{ such that } DG = GP_\sigma, \end{aligned} \tag{17}$$

where P_σ is the permutation matrix associated with σ.

Theorem 7. *$\mathscr{A}ut(\mathscr{C})$ is not trivial.*

Proof. Let $\sigma_0 \in S_{2n}$; its corresponding permutation matrix $P_{\sigma_0} = \sigma_0(I_{2n})$, where

$$\sigma_0 = (1, n+1)(2, n+2)\cdots(n, 2n),$$

$$\begin{aligned}\sigma_0 \in \mathscr{A}ut(\mathscr{C}) &\Leftrightarrow \exists D \in GL_n(2^q) \text{ such that } DG = GP_{\sigma_0}\\ &\Leftrightarrow DI_n + DM = MI_n + I_n^2\\ &\Leftrightarrow DI_n = MI_n \text{ and } DM = I_n^2\\ &\Leftrightarrow M^2 = I_n.\end{aligned} \tag{18}$$

The automorphism group $\mathscr{A}ut(\mathscr{C})$ always contains at least one nontrivial automorphism, distinct from the identity automorphism. □

Corollary 8. *Let* $\sigma_0 = (1, n+1)(2, n+2)\cdots(n, 2n)$.

$$\sigma_0 \in \mathscr{A}ut(\mathscr{C}) \; M \text{ is involutory MDS matrix}. \tag{19}$$

Proof. The proof is an outcome of the preceding proof. □

Proposition 9. *Let* $\pi, \sigma \in \mathscr{S}_{2n}$ *such that* $P_\pi = (U \oplus L)$, *where* U *and* L *are two permutation matrices of permutations in* $\mathscr{S}_n$:

$$\begin{aligned}&P_\sigma = P_{\sigma_0} P_\pi,\\ &\pi \in \mathscr{A}ut(\mathscr{C}) \Leftrightarrow UM = ML,\\ &\sigma \in \mathscr{A}ut(\mathscr{C}) \Leftrightarrow MUM = L.\end{aligned} \tag{20}$$

Proof.

$$\begin{aligned}\pi \in \mathscr{A}ut(\mathscr{C}) &\Leftrightarrow \exists D \in GL_n(2^q) \text{ such that } DG = GP_\pi\\ &\Leftrightarrow DI_n + DM = I_nU + ML\\ &\Leftrightarrow DI_n = I_nU \text{ and } DM = ML\\ &\Leftrightarrow UM = ML,\end{aligned}$$

$$\begin{aligned}\sigma \in \mathscr{A}ut(\mathscr{C}) &\Leftrightarrow \exists D \in GL_n(2^q) \text{ such that } DG = GP_\sigma\\ &\Leftrightarrow \exists D \in GL_n(2^q) \text{ such that } DG = GP_{\sigma_0}P_\tau\\ &\Leftrightarrow \exists D \in GL_n(2^q) \text{ such that } D[I_nM] = [I_nM]\begin{bmatrix} 0 & L\\ U & 0\end{bmatrix}\\ &\Leftrightarrow DI_n + DM \text{ and } MU + I_nL\\ &\Leftrightarrow DI_n = MU \text{ and } DM = I_nL\\ &\Leftrightarrow MUM = L.\end{aligned} \tag{21}$$

□

Proposition 10. *Let* $M \in GL_n(2^q)$ *with* M *as an involutory matrix and* $\mathscr{P}_m$ *a group of* $n \times n$ *permutation matrices; then,* $\sigma \in \mathscr{A}ut(M) \Leftrightarrow MP_\sigma M \in \mathscr{A}ut(M)$.

Proof.

$$\begin{aligned}\sigma \in \mathscr{A}ut(M) &\Leftrightarrow \exists D \in \mathscr{P}_m \text{ such that } DMP_\sigma = M\\ &\Leftrightarrow \exists D \in \mathscr{P}_m \text{ such that } P_\sigma^\intercal M^\intercal D^\intercal = M^\intercal\\ &\Leftrightarrow \exists D \in \mathscr{P}_m \text{ such that } P_\sigma^{-1} M^\intercal D^{-1} = M^\intercal\\ &\Leftrightarrow D^{-1} \in \mathscr{A}ut(M^\intercal)\\ &\Leftrightarrow D^{-1} \in \mathscr{A}ut(M)\\ &\Leftrightarrow D \in \mathscr{A}ut(M)\\ &\Leftrightarrow M(MP_\sigma)^{-1} \in \mathscr{A}ut(M)\\ &\Leftrightarrow MP_\sigma^{-1}M^{-1} \in \mathscr{A}ut(M)\\ &\Leftrightarrow MP_\sigma M^{-1} \in \mathscr{A}ut(M)\\ &\Leftrightarrow MP_\sigma M \in \mathscr{A}ut(M).\end{aligned} \tag{22}$$

□

Theorem 11. *Let* M *be an involutory matrix; then,*

(1) $S_1 = \{(MP_\sigma M \oplus P_\sigma) : \sigma \in \mathscr{A}ut(M)\}$ *is an automorphism group of* $\mathscr{C}$

(2) $S_2 = \{P_{\sigma_0}(MP_\sigma M \oplus P_\sigma) : \sigma \in \mathscr{P}_m\}$ *is an automorphism set of* $\mathscr{C}$

Proof. Let U and L be two permutation matrices.

$$\begin{aligned}P_\tau = (U \oplus L) \in S_1 &\Leftrightarrow UM\\ &= ML \Leftrightarrow U\\ &= MLM^{-1} \text{ and } L \in \mathscr{A}ut(M) \Leftrightarrow U\\ &= MLM \text{ and } L \in \mathscr{A}ut(M) \Leftrightarrow P_\tau\\ &= (MLM \oplus L) \text{ and } L \in \mathscr{A}ut(M),\end{aligned}$$

$$\begin{aligned}P_\sigma = P_{\sigma_0}(U \oplus L) \in S_2 &\Leftrightarrow MUM\\ &= L \Leftrightarrow P_\sigma\\ &= P_{\sigma_0}(U \oplus MUM) \text{ and } U\\ &= M^{-1}LM^{-1} \Leftrightarrow P_\sigma\\ &= P_{\sigma_0}(U \oplus MUM) \text{ and } U\\ &= MLM \Leftrightarrow P_\sigma\\ &= P_{\sigma_0}(U \oplus MUM) \text{ and } U \in \mathscr{P}_m.\end{aligned} \tag{23}$$

□

Corollary 12. *Let* M *be an involutory matrix; then,* $S_3 = \langle P_{\sigma_0}, (MP_\sigma M \oplus P_\sigma) : \sigma \in \mathscr{A}ut(M)\rangle\}$ *is an automorphism group of* $\mathscr{C}$.

Proof. Since $\forall \sigma \in \mathscr{M}$, from Theorem 11, $(MP_\sigma M \oplus P_\sigma) = (P_\sigma \oplus MP_\sigma M)$ and $(P_\sigma \oplus MP_\sigma M)P_{\sigma_0}$ are automorphisms of the code $\mathscr{C}$.

The identification of the automorphism group is straightforward when the permutation on the left side is the identity

permutation. This is deduced through the application of the equivalence property, as exemplified in Example 1. □

If $|S_3| = |\mathcal{A}ut(\mathcal{C})|$, we say that S_3 is the full automorphism group of the code $\mathcal{C}$. Also, the number of distinct codes that are equivalent to $\mathcal{C}$ is $(2n)!/|\mathcal{A}ut(C)|$.

6. Conclusion

In this paper, we present an innovative methodology for the identification of MDS matrices within finite fields $\mathbb{F}_2^q$. This approach leverages genetic algorithms and draws from MDS self-dual codes, demonstrating its effectiveness through a practical example. The discovered matrices offer valuable applications, notably in establishing essential automorphism groups that play a pivotal role in decoding algorithms, further enhancing our understanding of code structures. Looking forward, our future endeavors will emphasize the exploration of larger matrices, expanding the scope of practical applications. Additionally, our focus will extend to the deliberate construction of matrices with specific and noteworthy properties, contributing to the advancement of coding theory and cryptographic systems. These efforts collectively deepen our understanding and open new avenues for research in this critical domain.

Data Availability

All data that were analyzed or generated are encompassed within this published article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. M. Corbella and R. Pellikaan, "A characterization of MDS codes that have an error correcting pair," 2015, https://arxiv.org/abs/1508.02187.

[2] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge university press, 2010.

[3] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes Vol. 16*, Elsevier, 1977.

[4] X. Zhou and T. Cong, *Construction of Generalized-Involutory MDS Matrices*, Cryptology ePrint Archive, 2022.

[5] K. Gupta, S. Kumar Pandey, I. Ghosh Ray, and S. Samanta, "Cryptographically significant MDS matrices over finite fields: a brief survey and some generalized results," *Advances in Mathematics of Communications*, vol. 13, no. 4, pp. 779–843, 2019.

[6] S. Samanta, "On the Counting of Involutory MDS Matrices," 2023, https://arxiv.org/abs/2310.00090.

[7] S. Ball, *A Course in Algebraic Error-Correcting Codes. In Compact Textbooks in Mathematics*, Springer International Publishing, Cham, 2020.

[8] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, Cambridge, 2012.

[9] S. M. Sim, K. Khoo, F. Oggier, and T. Peyrin, "Lightweight MDS involution matrices," in *Fast Software Encryption*, G. Leander, Ed., vol. 9054 of Lecture Notes in Computer Science, pp. 471–493, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[10] I. M. Alamsyah and F. Yuliawan, "A construction of MDS involutory matrices using MDS self-dual codes: a preliminary result," *Journal of Physics: Conference Series*, vol. 1722, no. 1, 2021.

[11] K. Gupta, S. Kumar Pandey, and I. Ghosh Ray, "Applications of design theory for the constructions of MDS matrices for lightweight cryptography," *Journal of Mathematical Cryptology*, vol. 11, 2017.

[12] M. Sajadieh, M. Dakhilalian, H. Mala, and B. Omoomi, "On construction of involutory MDS matrices from Vandermonde matrices in GF(2q)," *Designs, Codes and Cryptography*, vol. 64, no. 3, pp. 287–308, 2012.

[13] K. Chand Gupta and I. Ghosh Ray, "On constructions of involutory MDS matrices," in *International Conference on Cryptology in Africa*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[14] M. K. Pehlivanoğlu, M. T. Sakallı, S. Akleylek, N. Duru, and V. Rijmen, "Generalisation of Hadamard matrix to generate involutory MDS matrices for lightweight cryptography," *IET Information Security*, vol. 12, no. 4, pp. 348–355, 2018.

[15] Q. Li, B. Wu, and Z. Liu, "Direct constructions of (involutory) MDS matrices from block Vandermonde and Cauchy-like matrices," in *Arithmetic of Finite Fields*, L. Budaghyan and F. Rodríguez-Henríquez, Eds., vol. 11321 of Lecture Notes in Computer Science, pp. 275–290, Springer International Publishing, Cham, 2018.

[16] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

[17] E. M. Bellfkih, S. Nouh, I. C. Idrissi, A. Ettaoufik, K. Louartiti, and J. Mouline, "On the computation of the automorphisms group of low density parity check codes using genetic algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, 2022.

[18] C. Papadimitriou and K. Steiglitz, "Combinatorial optimization: algorithms and complexity," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, 1982.

[19] M. P. Cuéllar, J. Gómez-Torrecillas, F. J. Lobillo, and G. Navarro, "Genetic algorithms with permutation-based representation for computing the distance of linear codes," *Swarm and Evolutionary Computation*, vol. 60, article 100797, 2021.

[20] S. Nouh, I. Chana, and M. Belkasmi, "Decoding of block codes by using genetic algorithms and permutations set," *International Journal of Communication Networks and Information Security*, vol. 5, no. 3, 2013.

[21] E. M. Bellfkih, S. Nouh, I. C. Idrissi, A. K. Louartiti, and J. Mouline, "On the computation of the automorphisms group of some optimal codes using genetic algorithm," *Journal of Hunan University Natural Sciences*, vol. 49, no. 1, 2022, http://jonuns.com/index.php/journal/article/view/945.