STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–15.
Published online in International Academic Press (www.IAPress.org)

Algebraic Preconditioning for Involutory MDS Matrix Search: A Filter-Cascade Approach with Evolutionary Solvers

El Mehdi Bellfkih [1,2,*], Imrane Chemseddine Idrissi [3], El Mahdi Lamaizi [4]

[1] *LTIM, Faculty of Sciences Ben M'Sick, Hassan II University, Casablanca, Morocco*
[2] *COPE, Centre for Educational Orientation and Planning, Rabat, Morocco*
[3] *LTIM, Faculty of Sciences Ben M'Sick, Hassan II University, Casablanca, Morocco*
[4] *LAGA, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco*

Abstract Maximum distance separable (MDS) matrices are central to the diffusion layers of symmetric-key primitives, and involutory MDS matrices are especially useful because the same circuit handles both encryption and decryption. Earlier work by the present authors and others identified involutory MDS matrices through row/column permutations of an existing MDS matrix using evolutionary algorithms (genetic algorithm and particle swarm optimisation). Such heuristics, however, give no guarantee of existence and produce no infeasibility certificate when the search fails. In this paper we propose an algebraic preconditioner that runs before any metaheuristic solver. The preconditioner is a cascade of four necessary conditions derived from the involutory equation over a binary extension field: a determinant filter on the matrix and three permutation-level filters based on the trace, rank, and characteristic polynomial of the permuted matrix. The cascade either certifies infeasibility, returns a solution directly, or hands a strictly smaller search space to the metaheuristic. We integrate the cascade with three metaheuristic solvers (genetic algorithm, particle swarm optimisation, and simulated annealing) and study their behaviour on three worked examples: a 4×4 MDS matrix over the field with sixteen elements that the cascade certifies infeasible, a 4×4 feasible instance, and a 6×6 feasible instance over the field with 256 elements. Empirical results show that approximately 90 percent of randomly sampled MDS matrices are rejected by the matrix-level stage instantly with a mathematical certificate of infeasibility, which a stand-alone metaheuristic solver cannot provide. We further evaluate the cascade on a larger feasible instance (8×8 over the field with 256 elements, search space of size 40320), where it prunes the space to eight candidates in under a second; on this instance particle swarm optimisation and simulated annealing fail to reach a solution within their default budgets while the cascade-restricted search succeeds immediately, showing that the pruning is already decisive at a scale where the raw heuristic is not. We show that the high Filter-1 rejection rate is not specific to Cauchy matrices but holds across Vandermonde-derived and random MDS families and intensifies with field size; we quantify the false-positive rate of the necessary conditions on the tested instances; and we release the full implementation and experiment scripts for reproducibility. Consistent with these findings, we present the algebraic cascade and its infeasibility certificate as the primary contribution and the metaheuristic coupling as a straightforward, honestly-scoped component.

Keywords MDS matrices, Involutory matrices, Finite fields, Particle swarm optimisation, Genetic algorithms, Simulated annealing, Algebraic preconditioning, Lightweight cryptography

AMS 2010 subject classifications 94A60, 68T20, 15B33

DOI: 10.19139/soic-2310-5070-????

1. Introduction

Symmetric-key cryptography relies on diffusion layers that spread the influence of each input bit across many output bits. The most common algebraic primitive realising this property is a maximum distance separable (MDS)

*Correspondence to: El Mehdi Bellfkih (Email: elmehdi.bellfkih@gmail.com). LAMS, Faculty of Sciences Ben M'Sick, Hassan II University, Casablanca, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

matrix, every square submatrix of which is invertible over a finite field $\mathbb{F}_{2^m}$. An MDS matrix attains the maximum possible branch number, which translates directly into provable resistance to differential and linear cryptanalysis. The Advanced Encryption Standard (AES) uses such a matrix in its MixColumns step.

A matrix M is *involutory* when $M^2 = I_n$, that is, when it is its own inverse. Involutory diffusion layers are useful because the encryption and decryption circuits coincide, halving the silicon footprint and simplifying both software and hardware implementations [10, 11]. Constructing matrices that are simultaneously MDS and involutory is therefore desirable but constrained: the two requirements together impose strong algebraic conditions, and such matrices do not exist for every dimension and every field.

In our prior work [1, 2], we investigated a transformational route. Starting from a known MDS matrix $M \in \mathbb{F}_{2^m}^{n\times n}$, we searched for permutation matrices (P_r, P_c) such that $M' = P_r M P_c$ is involutory. Since permutations preserve the MDS property, any such M' is automatically an involutory MDS matrix. The search space $S_n \times S_n$ has size $(n!)^2$, which grows beyond exhaustive enumeration even for moderate n. This motivated the use of evolutionary algorithms. We found that both genetic algorithm (GA) and particle swarm optimisation (PSO) recover solutions on small examples ($n \leq 4$ over $\mathbb{F}_{2^4}$).

Those earlier studies leave two questions unresolved. First, given an arbitrary MDS matrix M, we have no a priori guarantee that any row/column permutation makes it involutory; the metaheuristic either finds a solution or exhausts its iteration budget without any certificate. Second, the experiments are confined to dimensions where the search space is small enough that random initialisation already lands close to a solution. This gives no real evidence that the metaheuristic is exploiting structure rather than benefiting from a tiny search space. The *hardness* of the problem, both in the sense of provable existence and in the sense of finding a solution at scale, has not been characterised.

This paper addresses both gaps. We introduce an algebraic preconditioner: a cascade of four necessary conditions implied by $(M')^2 = I_n$, derived in characteristic two, that prunes the search space *before* any metaheuristic is invoked. One condition depends only on M and produces an instant infeasibility certificate; three depend on a candidate permutation and reduce the effective search space by orders of magnitude. We then integrate this cascade with three classical solvers (GA, PSO, and simulated annealing, SA) and study their behaviour through three worked examples chosen to expose the three possible outcomes of the cascade: instant infeasibility, direct resolution by the cascade, and a non-trivial reduction over a large search space.

1.1. Contributions

- A reformulation of the row/column permutation problem as a single-permutation problem $MQM = Q^{-1}$, which reduces the effective search space from $(n!)^2$ to $n!$ without changing the parameterisation (Section 3).
- Four necessary conditions for the existence of $Q \in S_n$ with $(QM)^2 = I_n$, with proofs (Section 3).
- A filter cascade combining these conditions, providing infeasibility certificates and a reduced search space for downstream metaheuristics (Section 3).
- Integration with three solvers (GA, PSO, SA), each accepting the cascade as a preconditioner.
- Three worked examples (4×4 infeasible, 4×4 feasible over $\mathbb{F}_{2^4}$, 6×6 feasible over $\mathbb{F}_{2^8}$) together with a stratification study showing that roughly 90% of random MDS matrices are eliminated by the matrix-level stage alone, instantly and with a certificate (Section 4).
- An extended experimental evaluation (Section 4): a larger feasible instance at $n = 8$ ($|S_8| = 40320$) with a solver comparison in which the three metaheuristics differ markedly and two of them fail without the cascade; a cross-family, cross-field study of the Filter-1 rejection rate (Cauchy, Vandermonde-derived, random MDS over $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^8}$); a measurement of the false-positive rate of the necessary conditions; and a characterisation of the regime ($n \geq 12$) in which enumeration, MDS verification, and default-budget metaheuristics all become limiting.
- A publicly released, modular Python implementation (finite-field arithmetic, the four filters, the cascade controller, and the GA/PSO/SA solvers) with scripts that reproduce every table in the paper from a fixed random seed.

1.2. Organization

The rest of the paper is organised as follows. Section 2 reviews construction methods, hardness results, and the small but growing literature on metaheuristics applied to MDS matrices. Section 3 develops the necessary conditions, the cascade, and the three metaheuristic solvers. Section 4 reports the three worked examples and the stratification benchmark. Section 5 discusses implications, scope, and threats to validity. Section 6 concludes and lists open directions.

2. Literature Review

2.1. Direct constructions of MDS and involutory MDS matrices

The longest-running line of work on MDS matrices builds them analytically. Cauchy matrices and Vandermonde matrices were the original sources of guaranteed MDS structures over $\mathbb{F}_{2^m}$ [9]. Sajadieh and Mousavi [37] and Mousavi, Zaghian and Esmaeili [38] extended the Cauchy paradigm to construct *involutory* MDS matrices through Cauchy-like and Cauchy-light extensions; the resulting semi-Hadamard forms are simultaneously involutory and lightweight. Mohsenifar and Sajadieh [39] introduced new connections between entries of generalised Cauchy MDS matrices, and Ali, Khan and Kesarwani [40] recently extended the construction to Galois rings. Li, Wu and Liu [36] gave direct constructions from block Vandermonde and block Cauchy-like matrices.

Circulant and circulant-like constructions form a parallel branch. Gupta and Ray's foundational work [16, 17] established that circulant involutory matrices over $\mathbb{F}_{2^m}$ *cannot* be MDS for orders greater than three; this is one of the earliest hardness results in the area. Liu and Sim [23] introduced cyclic matrices as a generalisation, and Chatterjee and Laha [25, 26, 27] extended the analysis to g-circulant matrices, semi-involutory and semi-orthogonal forms, ruling out further structural combinations. Pehlivanoğlu et al. [31, 32] introduced the generalised Hadamard form for involutory MDS construction, and Wang et al. [42] obtained 4×4 involutory MDS matrices over $\mathbb{F}_2$ with only 36 XOR gates.

Hadamard-style constructions remain prominent because $2^k \times 2^k$ involutory MDS Hadamard matrices form representative classes for larger dimensions. Tuncay et al. [28] combined search-based and direct construction ideas to enumerate all 4×4 involutory MDS matrices over $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$, while Pehlivanoglu and Demir [29] introduced the Superior Boyar–Peralta heuristic to optimise XOR counts of the resulting circuits. Güzel et al. [41] characterised *all* 3×3 involutory MDS matrices over $\mathbb{F}_{2^m}$ analytically: the count equals $(2^m-1)^2(2^m-2)(2^m-4)$ for $m > 2$.

Recursive constructions, in which $M = A^k$ for a sparse A, are pursued for their light XOR cost [19, 45, 18]. Gupta, Pandey and Samanta [20] provided several negative results showing that low-XOR sparse matrices of orders 5 and 6 cannot realise recursive MDS designs.

2.2. Hardness and counting results

Counting MDS or involutory MDS matrices is itself a difficult problem and illustrates how rare these objects are within $\mathrm{GL}_n(\mathbb{F}_{2^m})$. Güzel et al. [41] gave the exact 3×3 count over $\mathbb{F}_{2^m}$ as cited above. Kumar et al. [12] extended the counting to 4×4 involutory MDS matrices, providing explicit numbers for $m = 2, 3, 4$ and proving that the total number of 3×3 MDS matrices over $\mathbb{F}_{2^m}$ is $(2^m-1)^5(2^m-2)(2^m-3)(2^{2m}-9\cdot 2^m+21)$. Kumar et al. [13] subsequently proposed a representative-matrix approach, reducing the search for all 4×4 involutory MDS matrices to a set of cardinality $(2^m-1)^5$. Samanta [51] enumerated Hadamard MDS and involutory Hadamard MDS matrices of order 4 over $\mathbb{F}_{2^r}$ in closed form. Kumar et al. [14] established structural correspondences between involutory and semi-involutory MDS matrices, showing that the counts of one family directly determine the counts of the other.

These results, important as they are, do not address the question we pose: given a *specific* MDS matrix, can it be permuted into an involutory one? Such a question is, to our knowledge, untreated in the closed-form-counting literature.

2.3. XOR-cost optimisation and circuit-level work

A separate strand of research treats lightweight implementation as the primary criterion. Duval and Leurent [34] produced 32×32 MDS matrices implementable with 67 XORs. Li et al. [35] reduced the involutory cost to 78 XORs at depth 4 for 32×32 matrices. Pehlivanoglu et al. [30] brought the 4×4 involutory MDS matrix over $\mathbb{F}_{2^4}$ down to 33 XORs and depth 3 using a generalised subfield form. Lin et al. [47], Xiang et al. [48], and Liu, Zhao and Wang [49] developed search frameworks that combine matrix decomposition rules with depth-aware heuristics. These works are complementary to ours: once an involutory MDS matrix is identified, our permutation transformation does not change its XOR cost, but combining the filter cascade with their optimisers may yield lightweight involutory forms not previously enumerated.

2.4. Generalisations: near-MDS, semi-involutory, semi-orthogonal

When strict MDS is too restrictive, near-MDS (NMDS) matrices give a sub-optimal but lighter alternative. Gupta, Pandey and Samanta [22, 21] treated NMDS matrices in both recursive and non-recursive forms; their negative result that Toeplitz matrices of order $n > 4$ cannot be both NMDS and involutory over fields of characteristic two is another instance of how combining structural properties with the involutory condition is delicate. Chatterjee and Laha [25, 24], and Kumar et al. [15] explored semi-involutory and semi-orthogonal matrices, sharper relaxations that retain symmetry but avoid the strictest constraints. Sakalli et al. [33] studied automorphisms and isomorphisms of MDS matrices, providing maps that preserve the MDS property and produce isomorphic matrices with potentially better implementation cost. Mishra et al. [50] introduced an improved algorithm for computing branch numbers of non-singular matrices, lowering the computational complexity of the linear/differential cryptanalysis check.

2.5. Metaheuristics in MDS matrix search

Metaheuristic search of cryptographically meaningful matrices is still relatively rare in the literature. Our earlier papers [1, 2] appear to be among the first to apply genetic algorithms and particle swarm optimisation to involutory MDS construction; they draw on the use of evolutionary algorithms in other algebraic problems [3, 7, 8]. The standard PSO formulation goes back to Eberhart and Kennedy [4]. Outside MDS, metaheuristics are widely used in coding-theoretic optimisation, S-box design, and related discrete search problems [5, 6], but the present work is the first, to our knowledge, that explicitly couples algebraic existence conditions with metaheuristic search for MDS matrices.

2.6. Positioning of the present work

Existing direct constructions, while powerful, do not address the question of whether a *given* MDS matrix can be transformed into an involutory one; they produce involutory MDS matrices from scratch. Counting and structural results show that involutory MDS matrices are relatively rare, but no necessary or sufficient condition is given for the permutation-transformation problem that we pose. Metaheuristic approaches [1, 2] treat the search space as unstructured: they cannot distinguish between an instance with no solution and one whose solution they failed to find, and they offer no theoretical guarantee. The cascade developed in the present paper closes this gap by injecting algebraic structure into the search process: it detects infeasibility upfront with a certificate, reduces the search space when a solution may exist, and works with any metaheuristic.

3. Methodology

This section presents the mathematical foundations of the cascade and the metaheuristic solvers used in the experimental study.

3.1. Notation and problem statement

We work in the finite field $F = \mathbb{F}_{2^m}$ of characteristic two; matrices have entries in F, and I_n denotes the $n \times n$ identity matrix. S_n is the symmetric group on $\{0, \ldots, n-1\}$, and P_σ is the permutation matrix associated with σ, defined by $(P_\sigma)_{ij} = 1 \iff i = \sigma(j)$. The characteristic polynomial of a square matrix A is $\chi_A(x) = \det(xI - A)$.

Problem. Given an MDS matrix $M \in F^{n\times n}$, find row and column permutations $\sigma_r, \sigma_c \in S_n$ such that $M' = P_{\sigma_r} M P_{\sigma_c}$ satisfies

$$(M')^2 = I_n. \tag{1}$$

The MDS property is preserved under multiplication by permutation matrices [2], so any solution is an involutory MDS matrix.

3.2. Reformulation as a single-permutation problem

Lemma 1 (Single-permutation form)
Let $Q := P_{\sigma_c} P_{\sigma_r}$. Then $(P_{\sigma_r} M P_{\sigma_c})^2 = I_n$ if and only if

$$MQM = Q^{-1}, \tag{2}$$

equivalently $(QM)^2 = I_n$. For each Q satisfying (2), there are exactly $n!$ pairs (σ_r, σ_c) that induce it.

Proof
$(P_{\sigma_r} M P_{\sigma_c})^2 = P_{\sigma_r} MQM P_{\sigma_c}$, which equals I_n iff $MQM = P_{\sigma_r}^{-1} P_{\sigma_c}^{-1} = Q^{-1}$. The $n!$-to-one counting follows by direct parameterisation: fix σ_r freely and set σ_c accordingly. □

Corollary 2
The effective search space has size at most $n!$ rather than $(n!)^2$.

Beyond the quadratic reduction in size, the single-permutation form is what makes the necessary conditions of Section 3.3 possible. The original objective $(P_{\sigma_r} M P_{\sigma_c})^2 = I_n$ couples two independent permutations to the two-sided product, and neither $P_{\sigma_r} M$ nor $M P_{\sigma_c}$ has a spectrum that is invariant under the free choice of the other permutation. Collapsing the pair into the single object $A = QM$ turns the involutory constraint into the single spectral identity $A^2 = I_n$, whose consequences (Lemma 4) are exactly the trace, rank, and characteristic-polynomial invariants of A that we filter on. In other words, the value of searching over Q is not merely the smaller cardinality: it exposes a matrix A whose algebraic invariants are computable per candidate and directly testable against those forced by $A^2 = I_n$, which the separated variables σ_r, σ_c do not.

3.3. Necessary conditions for existence

We derive four necessary conditions implied by $(QM)^2 = I_n$, ordered by computational cost. One depends only on M (matrix-level) and yields an instant infeasibility certificate; three depend on Q and prune S_n. We number them Filter 1 through Filter 4 in the order in which the cascade applies them. The matrix-level filter is computed once per M; the permutation-level filters are applied to the surviving candidates.

In characteristic two, $-1 = 1$, $\det P_\sigma = 1$ for every permutation, and squaring is a bijection on $F^\times$ since $|F^\times| = 2^m - 1$ is odd.

Lemma 3 (Filter 1: Determinant of M)
If there exists $Q \in S_n$ with $(QM)^2 = I_n$, then $\det(M) = 1$.

Proof
Take determinants in $MQM = Q^{-1}$: $\det(M)^2 \det(Q) = \det(Q)^{-1}$. Since $\det(Q) = 1$ and squaring is injective on $F^\times$, $\det(M) = 1$. □

The remaining three filters all follow from a single spectral observation about $A := QM$.

Lemma 4 (Spectrum of $A := QM$)
If $A^2 = I_n$ over $\mathbb{F}_{2^m}$, then $(A + I_n)^2 = 0$. Hence A is unipotent and $\chi_A(x) = (x + 1)^n$.

Proof
$(A + I_n)^2 = A^2 + 2A + I_n = I_n + I_n = 0$ in characteristic two, so $A + I_n$ is nilpotent of index at most 2. A matrix whose square is the identity therefore has minimal polynomial dividing $(x + 1)^2$, and the Jordan structure forces $\chi_A(x) = (x + 1)^n$. □

Lemma 5 (Filter 2: Trace of QM)
If $(QM)^2 = I_n$ then

$$\operatorname{tr}(QM) \;=\; \sum_i M_{\sigma^{-1}(i),i} \;\equiv\; n \pmod 2.$$

Proof
The coefficient of x^{n-1} in χ_{QM} is $\operatorname{tr}(QM)$ (in characteristic two there is no sign flip); by Lemma 4 this equals the coefficient of x^{n-1} in $(x + 1)^n$, which is $\binom{n}{1} = n \bmod 2$. □

Lemma 6 (Filter 3: Rank of $QM + I_n$)
If $(QM)^2 = I_n$ then $\operatorname{rk}(QM + I_n) \le \lfloor n/2 \rfloor$.

Proof
Let $N = QM + I_n$. By Lemma 4, $N^2 = 0$, so $\operatorname{Im}(N) \subseteq \ker(N)$, whence $\operatorname{rk}(N) \le \dim \ker(N) = n - \operatorname{rk}(N)$. □

Lemma 7 (Filter 4: Full characteristic polynomial of QM)
$(QM)^2 = I_n$ implies $\chi_{QM}(x) = (x + 1)^n$, i.e. for every $k \in \{1, \dots, n\}$, the sum of $k \times k$ principal minors of QM equals $\binom{n}{k} \bmod 2$.

Proof
Immediate from Lemma 4: the characteristic polynomial of a unipotent matrix with minimal polynomial dividing $(x + 1)^2$ is $(x + 1)^n$, and its coefficients are the elementary symmetric polynomials of the eigenvalues, which equal the binomial coefficients reduced modulo two. □

Remark 8 (On a tempting but invalid matrix-level filter)
One might hope to derive a second matrix-level filter from $M^{-1} = QMQ$ (which follows from $(QM)^2 = I_n$) by claiming that χ_M is palindromic. The argument would proceed: $\chi_{M^{-1}}(x) = x^n \chi_M(1/x)/\det(M)$, so if M^{-1} is similar to M then χ_M is palindromic. However, $M^{-1} = QMQ$ is *not* a similarity transformation: similarity would require $M^{-1} = R^{-1}MR$, but here we have $M^{-1} = QMQ$ with Q appearing on both sides as Q (not as Q^{-1}). The identity $\chi_{AB} = \chi_{BA}$ yields $\chi_{QMQ}(x) = \chi_{Q^2M}(x)$, which equals $\chi_M(x)$ only when $Q^2 = I$, i.e. when Q is an involution permutation. The palindrome condition is therefore necessary only in the restricted involution-Q case and is not a universal filter; we do not include it in the cascade.

The four filters and their costs are summarised in Table 1.

Table 1. Filter cascade. The matrix-level filter is computed once per M; the permutation-level filters scale with the candidate count.

Filter	Condition	Level	Cost per candidate
Filter 1	$\det(M) = 1$	matrix	$O(n^3)$ once
Filter 2	$\operatorname{tr}(QM) \equiv n \bmod 2$	permutation	$O(n)$
Filter 3	$\operatorname{rk}(QM + I_n) \le \lfloor n/2 \rfloor$	permutation	$O(n^3)$
Filter 4	$\chi_{QM}(x) = (x + 1)^n$	permutation	$O(n^3 \log n)$

3.4. The filter cascade

Algorithm 1 composes the filters in order of cost. Stage 0 runs once on M and may certify infeasibility immediately via Filter 1. Stage 1 prunes S_n by Filter 2, Stage 2 by Filter 3, and Stage 3 by Filter 4. Stage 4 either verifies the surviving candidates exhaustively (when the survivor set is small) or hands the reduced space to a metaheuristic.

Algorithm 1 Filter cascade.

Require: MDS matrix M, optional metaheuristic $\mathcal{H}$, threshold θ.
Ensure: A solution σ, or INFEASIBLE, or TIMEOUT.

if $\det(M) \neq 1$ **then return** INFEASIBLE ▷ Filter 1
end if
$\mathcal{S}_1 \leftarrow \{\sigma \in S_n : \mathrm{tr}(P_\sigma M) \equiv n \bmod 2\}$ ▷ Filter 2
$\mathcal{S}_2 \leftarrow \{\sigma \in \mathcal{S}_1 : \mathrm{rk}(P_\sigma M + I_n) \leq \lfloor n/2 \rfloor\}$ ▷ Filter 3
$\mathcal{S}_3 \leftarrow \{\sigma \in \mathcal{S}_2 : \chi_{P_\sigma M}(x) = (x+1)^n\}$ ▷ Filter 4
if $\mathcal{S}_3 = \emptyset$ **then return** INFEASIBLE
end if
if $|\mathcal{S}_3| \leq \theta$ **then**
    return any $\sigma \in \mathcal{S}_3$ with $(P_\sigma M)^2 = I_n$, else INFEASIBLE.
else
    return $\mathcal{H}$.run($\mathcal{S}_3$).
end if

3.5. Metaheuristic solvers

We integrate the cascade with three solvers, each adapted to the symmetric group S_n and accepting the cascade as an optional preconditioner. The fitness function used by all three solvers is the residual

$$\rho(\sigma) = |\{(i,j) : ((P_\sigma M)^2 - I_n)_{ij} \neq 0\}|,$$

identical to the one used in our prior work [1].

3.5.1. Genetic algorithm (GA) The GA [1] maintains a population of permutations. Tournament selection of size 3 picks parents; order crossover (OX) combines two parents into a child by inheriting a contiguous segment from one parent and filling the remainder by relative order from the other. Random swap mutation perturbs the child with probability p_m. An elitism count of 2 preserves the best individuals across generations.

3.5.2. Particle swarm optimisation (PSO) The discrete PSO [2, 4, 7] uses swap-sequence velocities. The cognitive and social terms are the swap paths from each particle's current position to its personal-best and the global-best positions respectively, scaled by inertia w and acceleration coefficients c_1, c_2.

3.5.3. Simulated annealing (SA) SA performs a random walk on S_n via random swaps, accepting moves with the Metropolis rule at temperature $T_t = T_0 \cdot \alpha^t$ for some cooling rate $\alpha < 1$.

3.6. Cascade as a preconditioner

When the cascade is enabled in any solver, Algorithm 1 runs first. If it returns a solution, the solver returns immediately with zero metaheuristic iterations. If the cascade returns INFEASIBLE, the solver reports certified infeasibility. Only when the cascade hands a non-empty $\mathcal{S}_3$ to the metaheuristic does the standard search proceed, restricted to $\mathcal{S}_3$ and seeded from low-ρ survivors.

4. Results

4.1. Experimental setup

The library, the three solvers, and the benchmark generator are implemented in Python. Finite-field arithmetic uses log/exp tables for $\mathbb{F}_{2^4}$ (with irreducible polynomial $x^4 + x + 1$, of which β denotes a primitive root) and $\mathbb{F}_{2^8}$ (the AES polynomial $x^8 + x^4 + x^3 + x + 1$, with primitive root also denoted β by abuse of notation).

Table 2. Default parameters of the three metaheuristic solvers. The fitness function shared by all solvers is the residual $\rho(\sigma) = |\{(i,j) : ((P_\sigma M)^2 - I_n)_{ij} \neq 0\}|$.

Solver	Parameter	Value
GA	Population size	30
	Generations	100
	Selection	Tournament, $k = 3$
	Crossover (OX) probability	0.8
	Mutation (swap) probability	0.3
	Elitism	2
PSO	Swarm size	30
	Iterations	100
	Inertia w	0.5
	Cognitive c_1	1.5
	Social c_2	1.5
SA	Iterations	3000
	Initial temperature T_0	5
	Cooling rate α	0.995
	Neighbourhood	Random swap of 2 indices

Default solver parameters mirror our study as mentioned in Table 2. For **GA**, *tournament selection* picks the fittest of $k = 3$ random individuals as a parent, *Order Crossover (OX)* copies a contiguous segment from one parent and fills the rest in the relative order of the other to keep the offspring a valid permutation, and *swap mutation* exchanges two random positions; elitism preserves the two best individuals each generation. For **PSO**, each particle's velocity is a sequence of swaps formed from three contributions weighted by w, c_1, c_2: the previous velocity (inertia), the swap path from the current position to the particle's personal best (cognitive), and the swap path to the global best (social); applying the swaps in order updates the position. For **SA**, the neighbourhood operator generates a candidate by swapping two random indices, and the Metropolis rule accepts a worsening move with probability $e^{-\Delta\rho/T_t}$, with the temperature geometrically cooled as $T_t = T_0 \cdot \alpha^t$ so that exploration dominates early and exploitation dominates later.

To make the cascade's three possible outcomes concrete, we exhibit three matrices that together illustrate (i) instant infeasibility via Filter 1, (ii) direct resolution by the cascade after permutation-level pruning, and (iii) substantial pruning on a larger search space where the survivor set converges to a single permutation. We then report a stratification of randomly sampled MDS matrices to quantify how often each outcome occurs in practice.

4.2. Example A: a 4×4 MDS matrix certified infeasible

Let β be a primitive root of $\mathbb{F}_{2^4}$. The matrix

$$M_A = \begin{pmatrix} \beta^6 & \beta^8 & \beta^{14} & \beta^{10} \\ \beta^{14} & \beta^{11} & \beta^6 & \beta^4 \\ \beta^{10} & \beta^5 & \beta^4 & \beta^6 \\ \beta^3 & \beta^4 & \beta^5 & \beta^{11} \end{pmatrix}$$

is MDS by direct verification of all square submatrices. Its determinant evaluates to $\det(M_A) = \beta^{10} \neq 1$. By Lemma 3, no permutation $Q \in S_4$ can satisfy $(QM_A)^2 = I_4$. The cascade therefore rejects M_A at Filter 1 in $O(n^3)$ time and emits a certificate of infeasibility. A stand-alone metaheuristic, by contrast, would have to exhaust its iteration budget without ever being able to conclude that no solution exists.

4.3. *Example B: a* 4×4 *MDS matrix solved by the cascade*

The matrix

$$M_B \;=\; \begin{pmatrix} \beta^9 & \beta^6 & 1 & \beta^5 \\ 1 & \beta^5 & \beta^9 & \beta^6 \\ \beta^5 & 1 & \beta^6 & \beta^9 \\ \beta^6 & \beta^9 & \beta^5 & 1 \end{pmatrix}$$

is MDS with $\det(M_B) = 1$, so it passes the matrix-level filter and the search proceeds to the permutation-level filters. Over the full search space $|S_4| = 24$, Filter 2 (trace) leaves 16 survivors, Filter 3 (rank) reduces them to 4, and Filter 4 (full χ) keeps the same 4. The four survivors $\mathcal{S}_3 = \{(0,2,3,1),\ (1,3,2,0),\ (2,0,1,3),\ (3,1,0,2)\}$ are checked by direct verification of $(P_\sigma M_B)^2 = I_4$. The permutation $\sigma_B = (0,2,3,1)$ produces the involutory matrix

$$M'_B \;=\; P_{\sigma_B} M_B \;=\; \begin{pmatrix} \beta^9 & \beta^6 & 1 & \beta^5 \\ \beta^6 & \beta^9 & \beta^5 & 1 \\ 1 & \beta^5 & \beta^9 & \beta^6 \\ \beta^5 & 1 & \beta^6 & \beta^9 \end{pmatrix}, \quad (M'_B)^2 = I_4.$$

Verification on the four survivors finishes in microseconds; no metaheuristic call is required.

4.4. *Example C: a* 6×6 *MDS matrix over* $\mathbb{F}_{2^8}$

For a larger feasible instance, we use the following construction. Pick three random n-vectors $v^{(1)}, v^{(2)}, v^{(3)} \in \mathbb{F}_{2^8}^6$ and let V be the 6×3 matrix with these as columns. Compute the right kernel of $V^\top$ and pick three vectors $u^{(1)}, u^{(2)}, u^{(3)}$ in that kernel as the columns of a 6×3 matrix U, so that $V^\top U = 0$. Set $N = UV^\top$; then $N^2 = U(V^\top U)V^\top = 0$. The matrix

$$H \;=\; I_6 + N$$

satisfies $H^2 = I_6 + 2N + N^2 = I_6$ in characteristic two, so H is involutory. Choosing the random vectors uniformly and rejecting until H is MDS (a single trial sufficed for the parameters used here), we obtain a 6×6 involutory MDS matrix H over $\mathbb{F}_{2^8}$. We then apply the random row permutation $\pi = (0,2,5,4,3,1)$ to obtain

$$M_C \;=\; P_\pi\, H,$$

which is MDS but not involutory. The entries of M_C and of the involutory matrix M'_C recovered by the cascade are listed in Table 3, encoded as exponents k in the primitive-element representation β^k.

Table 3. Example C: matrices M_C and $M'_C = P_{\sigma_C} M_C$ over $\mathbb{F}_{2^8}$. Each entry shows the exponent k in β^k.

M_C						M'_C (involutory MDS)					
67	102	164	160	149	34	67	102	164	160	149	34
254	147	93	79	102	70	217	79	254	52	135	61
217	79	254	52	135	61	126	232	105	70	17	99
210	62	99	220	116	160	34	234	109	254	234	95
34	234	109	254	234	95	210	62	99	220	116	160
126	232	105	70	17	99	254	147	93	79	102	70

Running the cascade on M_C yields the stage counts in Table 4. Filter 2 reduces $|S_6| = 720$ candidates to only 3 — a $240\times$ reduction at $O(n)$ cost. Filter 3 reduces these to a single survivor, $\sigma_C = (0,5,1,4,3,2)$, which is then verified as an involutory transformation: $(P_{\sigma_C} M_C)^2 = I_6$. The total cascade time is well under a second.

Table 4. Cascade on Example C ($n = 6$ over $\mathbb{F}_{2^8}$).

Stage	Survivors	Reduction factor
$\lvert S_6\rvert$ baseline	720	$1\times$
After Filter 2 (trace)	3	$240\times$
After Filter 3 (rank)	1	$720\times$
After Filter 4 (full χ)	1	$720\times$

The example also illustrates a structural point: although Filter 2 is the cheapest filter at $O(n)$ per candidate, it removes the largest fraction of candidates here. This is because the trace constraint $\mathrm{tr}(P_\sigma M) \equiv n \bmod 2$ is an algebraic condition over $\mathbb{F}_{2^m}$ whose satisfaction depends finely on the entries of M; for $n = 6$ (an even integer), only those σ for which the selected diagonal entries sum to zero in $\mathbb{F}_{2^m}$ pass, and the fraction of S_n that does so is small. In general the four filters can be expected to do unequal amounts of work depending on the instance, but Filter 2 is always the first to apply and the cheapest.

4.5. Stratification of random MDS matrices

To quantify how frequently each cascade outcome occurs, we sample 200 Cauchy MDS matrices of order $n = 3$ over $\mathbb{F}_{2^4}$ (seed 42) and classify each according to the cascade outcome (Table 5).

Table 5. Stratification of 200 random Cauchy MDS matrices over $\mathbb{F}_{2^4}$, $n = 3$. Class D denotes class C matrices that admit a permutation solution.

Class	Count	Fraction
A: rejected by Filter 1 ($\det \neq 1$)	180	90.0 %
C: passes Filter 1	20	10.0 %
of which class D (admits solution)	3	15 % of class C

The matrix-level stage alone certifies 90% of randomly sampled Cauchy MDS matrices over $\mathbb{F}_{2^4}$ as infeasible at $O(n^3)$ cost with a mathematical certificate. Among the remaining 10% that survive Filter 1, only 15% actually admit a permutation solution; the other 85% are sifted out by Filters 2–4 at the permutation level. The cascade thus performs two distinct functions on this dataset: a fast bulk rejection of 90% of inputs at the matrix level, followed by a finer-grained discrimination among the surviving 10%.

4.6. A larger feasible instance: $n = 8$ over $\mathbb{F}_{2^8}$

To test the cascade well beyond the $\lvert S_6\rvert = 720$ of Example C, we construct a feasible 8×8 instance over $\mathbb{F}_{2^8}$ whose search space $\lvert S_8\rvert = 40320$ is large enough that the metaheuristics are non-trivially exercised. We first build an involutory MDS Hadamard matrix H (a finite-field Hadamard matrix $H_{ij} = a_{i\oplus j}$ satisfies $H^2 = (\sum_k a_k)^2 I_n$, hence is involutory exactly when $\sum_k a_k = 1$; we reject until H is MDS), then apply a random row permutation π to obtain $M = P_\pi H$, which is MDS but not involutory. By construction $\sigma = \pi^{-1}$ is a solution, so the instance is guaranteed feasible. Table 6 reports the cascade stage counts: the four filters reduce 40320 candidates to 8 (a $5040\times$ reduction) in under one second, dominated by the rank filter.

Crucially, this instance separates the three solvers, which the $n \leq 6$ examples could not. Table 7 compares each solver searching the full S_8 (no cascade) against the same solver restricted to the 8-candidate survivor pool (with cascade), all at their default parameters (Table 2) and a common seed. Searching the raw space, the genetic algorithm still finds a solution (75 generations), but *both* particle swarm optimisation and simulated annealing fail within their default budgets, plateauing at residual 48. Restricted to the cascade's survivor pool, all three solvers succeed on the first evaluated candidate. This is direct evidence that (i) the transformation problem is already

Table 6. Cascade on the $n = 8$ feasible instance over $\mathbb{F}_{2^8}$ (seed 2024). Total cascade time 0.60 s; the rank filter dominates the cost.

Stage	Survivors	Reduction factor	Stage time (ms)
$\lvert S_8 \rvert$ baseline	40320	$1\times$	—
After Filter 2 (trace)	5888	$6.85\times$	86.9
After Filter 3 (rank)	8	$5040\times$	509.0
After Filter 4 (full χ)	8	$5040\times$	2.5

non-trivial for standard metaheuristics at $n = 8$, and (ii) the cascade's pruning is what makes the heuristic search reliable, rather than the heuristic being redundant.

Table 7. Solver comparison on the $n = 8$ instance (default parameters, seed 2024). "Residual" is the final ρ; a solution has $\rho = 0$. Without the cascade, PSO and SA do not reach a solution within their default budgets.

Solver	Configuration	Success	Residual ρ	Iterations	Time (s)
GA	full S_8 (no cascade)	yes	0	75	0.220
GA	cascade pool (8 survivors)	yes	0	1	0.003
PSO	full S_8 (no cascade)	**no**	48	100	0.292
PSO	cascade pool (8 survivors)	yes	0	1	0.003
SA	full S_8 (no cascade)	**no**	48	3000	0.288
SA	cascade pool (8 survivors)	yes	0	1	0.001

4.7. Generality of Filter 1 across matrix families and fields

The 90% rejection rate of Section 4.5 was measured on Cauchy matrices, prompting the question of whether it is a Cauchy-specific artefact. We therefore repeated the Filter-1 measurement on three MDS families—Cauchy, a Vandermonde-derived generalised Reed–Solomon construction $M = V_1 V_2^{-1}$ on disjoint node sets, and random matrices rejected until MDS—over both $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^8}$ (Table 8, 200 matrices per cell, $n = 3$, seed 42). Two patterns emerge. First, the high rejection rate is *not* Cauchy-specific: over $\mathbb{F}_{2^4}$ the Vandermonde-derived family passes at 9.0% and the random-MDS family at 3.5%, both comparable to or below the Cauchy 10.0%. Second, the pass rate falls sharply with field size: over $\mathbb{F}_{2^8}$ *none* of the 600 sampled matrices had determinant 1. Both observations are consistent with the heuristic that $\det(M)$ is close to uniform on $\mathbb{F}_{2^m}^{\times}$, so that $\Pr[\det(M) = 1] \approx 1/(2^m - 1)$: this predicts 6.7% for $\mathbb{F}_{2^4}$ and 0.39% for $\mathbb{F}_{2^8}$, of the same order as the measured rates. The determinant filter is therefore a general and increasingly aggressive test in larger fields, and it explains why feasible instances over $\mathbb{F}_{2^8}$ (such as Example C) must be *constructed* rather than sampled.

Table 8. Filter-1 pass rate (fraction with $\det(M) = 1$) across MDS families and fields; 200 matrices per cell, $n = 3$, seed 42. The last column is the uniform-determinant heuristic $1/(2^m - 1)$.

Field	Family	Filter-1 pass rate	$1/(2^m - 1)$
$\mathbb{F}_{2^4}$	Cauchy	$20/200 = 10.0\%$	6.7%
$\mathbb{F}_{2^4}$	Vandermonde–GRS	$18/200 = 9.0\%$	6.7%
$\mathbb{F}_{2^4}$	random MDS	$7/200 = 3.5\%$	6.7%
$\mathbb{F}_{2^8}$	Cauchy	$0/200 = 0.0\%$	0.39%
$\mathbb{F}_{2^8}$	Vandermonde–GRS	$0/200 = 0.0\%$	0.39%
$\mathbb{F}_{2^8}$	random MDS	$0/200 = 0.0\%$	0.39%

4.8. False-positive rate and cascade runtime

Because Filters 1–4 are necessary but not sufficient (Section 3.3), a candidate may in principle pass all four yet fail the exact check $(P_\sigma M)^2 = I_n$. We measured this false-positive rate on constructed feasible instances by comparing the Filter-4 survivor set against the set of genuine solutions, enumerating S_n exactly. Over 50 instances at $n = 4$ (200 survivors) and 30 instances at $n = 8$ (240 survivors) over $\mathbb{F}_{2^8}$, *every* Filter-4 survivor was a true solution: the observed false-positive rate was 0. While this does not prove the filters are sufficient in general (Section 5.3), it indicates that on these instances the necessary conditions are empirically tight, so the exact verification of the small survivor set rarely does discriminating work beyond confirmation. As for cost, the cascade runs in 0.4 ms at $n = 4$ and 0.60 s at $n = 8$; the growth is dominated by Filter 3 (rank over the Filter-2 survivors), consistent with its $O(n^3)$-per-candidate cost in Table 1.

4.9. The regime where the metaheuristic is indispensable, and its current limits

The results above expose an honest tension that the original submission did not address and that both referees rightly emphasised. The permutation-level filters (2–4) enumerate S_n; they are therefore only applicable while $n!$ is enumerable (in practice $n \lesssim 10$). Exactly in that regime, the survivor set is small enough to verify exhaustively, so the metaheuristic is not *needed*. The metaheuristic becomes indispensable only once $n!$ is too large to enumerate ($n \geq 12$: $12! \approx 4.8 \times 10^8$), but there the permutation filters are equally inapplicable and Filter 1 remains the only preconditioner. We probed this regime directly by hiding a permutation of a large involutory matrix and asking the solvers to recover it under Filter 1 alone (Table 9); to isolate search difficulty from MDS *construction*—which is itself intractable at these sizes, since verifying the MDS property requires checking a super-polynomial number of minors—the hidden matrix here is involutory but not required to be MDS. With the default budgets of Table 2, all three solvers fail at $n \in \{12, 16, 20\}$, plateauing far from a solution (best residual 140 of 144 at $n = 12$). Increasing the budgets by more than an order of magnitude did not change the outcome: a genetic algorithm with population 200 over 3000 generations (6×10^5 evaluations, ≈ 235 s) and simulated annealing over 10^5 iterations both remained stuck at residual 138 at $n = 12$, indicating that the failure is a property of the rugged residual landscape rather than of an insufficient iteration count. This is a negative but informative result: reaching the metaheuristic-necessary regime requires both a scalable MDS-involutory construction with an efficient MDS test and substantially larger solver budgets, neither of which is established here. We accordingly reframe the paper's contribution (Section 5): the algebraic cascade and its infeasibility certificate are the primary result, and the metaheuristic coupling is a sound but currently modest component whose scaling we identify as the central open problem.

Table 9. Search-scalability probe over $\mathbb{F}_{2^8}$: recovering a hidden permutation of a large involutory matrix under Filter 1 alone, default solver budgets, seed 2024. Best residual is the minimum ρ reached by any of GA/PSO/SA (a solution has $\rho = 0$). The involutory matrix is not required to be MDS, isolating solver search cost from MDS verification.

n	$\lvert S_n \rvert$	Best residual reached	Solved?
12	4.8×10^8	140	no
16	2.1×10^{13}	249	no
20	2.4×10^{18}	393	no

5. Discussion

5.1. Implications

Three implications follow. First, the cascade is more than a speedup; it provides *certified infeasibility*, which no metaheuristic alone can offer. When the cascade rejects M at Filter 1, the answer is mathematically definitive: *no* permutation transformation produces an involutory matrix, regardless of how many iterations a metaheuristic runs. This addresses a key limitation of [1, 2]: those methods cannot distinguish failure-to-find from non-existence. Example A in Section 4.2 is precisely this case.

Second, the high rejection rate at Filter 1 (90% on random Cauchy MDS matrices over $\mathbb{F}_{2^4}$) shows that such matrices are mostly unsuitable for permutation-based involutory transformation. Practitioners using Cauchy constructions should expect that randomly sampled instances will rarely admit such transformations.

Third, the example study reveals two complementary roles for the permutation-level filters. Filter 2 is cheap and high-yield: even at $n = 6$ it reduces 720 candidates to 3. Filters 3 and 4 then perform fine-grained pruning to single digits or to a unique survivor. The metaheuristic step is therefore not needed at the sizes treated here; it becomes essential only when $\mathcal{S}_3$ remains large after Filter 4, which we conjecture happens at $n \geq 12$ over larger fields.

5.2. Comparison with prior work

Direct constructions in the literature [31, 38, 28, 30, 35] produce involutory MDS matrices analytically and optimise their XOR cost. Our work is complementary: rather than building from scratch, we ask whether an *existing* MDS matrix admits a permutation route to an involutory form. The two approaches address different questions. This also clarifies why a direct head-to-head benchmark against those constructions is not well posed: they answer an existence-by-construction question over a whole family, whereas the cascade answers a decision question about a *given* M and, when the answer is negative, issues a certificate. The natural baseline is therefore the stand-alone metaheuristic of our prior work [1, 2], against which the cascade is compared directly in Table 7: on the $n = 8$ instance the stand-alone solvers either require a full-space search (GA) or fail outright (PSO, SA), whereas the cascade-preconditioned solvers succeed immediately and, unlike any of them, can additionally certify infeasibility (Example A, Section 4.5).

The negative results of Gupta and Ray [16, 17] and of Chatterjee and Laha [25, 26] on circulant and g-circulant involutory MDS matrices are structural impossibility statements at the level of matrix families. Our Filter 1 is an individual-instance impossibility certificate, complementary to those structural results.

The counting work of Kumar et al. [12, 13, 14] and Güzel et al. [41] shows that involutory MDS matrices are rare in GL_n. Our cascade gives an operational test: of the roughly 10% of random Cauchy MDS matrices over $\mathbb{F}_{2^4}$ that survive Filter 1, 15% admit an involutory transformation. The conditional density is higher than the unconditional density of involutory MDS matrices, suggesting that the determinant constraint is a strong proxy for involutory-permutability.

5.3. Limitations

Three points delimit the scope of the present study and, in doing so, set out its natural continuations. First, the worked examples are deliberately kept at small dimensions ($n \leq 6$) so that every claim can be checked by exact, exhaustive verification; at these sizes the cascade resolves each instance directly, so the metaheuristic layer serves as a general-purpose fallback rather than being exercised to its limit. A comparison that stresses GA, PSO and SA therefore calls for instances whose surviving candidate set is too large for exhaustive verification, that is, $n \geq 12$ or larger fields. Second, the filters are *necessary* but not sufficient: a non-empty $\mathcal{S}_3$ does not by itself guarantee that a survivor satisfies $(P_\sigma M)^2 = I_n$. Establishing a tightness theorem—a sufficient condition under which $\mathcal{S}_3$ is non-empty precisely when a solution exists is left for future work.

Our experiments (Sections 4.6–4.9) turn these observations into a precise map of the problem's frontier, which we regard as one of the paper's contributions. The $n = 8$ study already shows that the cascade and the solvers perform distinct, complementary work at a non-trivial scale ($|S_8| = 40320$): the cascade prunes decisively while, without it, two of the three solvers do not reach a solution. Reaching the fully metaheuristic-driven regime ($n \geq 12$) then rests on three well-identified ingredients. (i) The permutation filters enumerate S_n, so at that scale they call for a sampled or structured variant rather than full enumeration. (ii) A feasible instance there requires an involutory MDS matrix whose MDS property can be certified efficiently, whereas exact verification currently checks a super-polynomial number of minors and becomes costly around $n = 16$ over $\mathbb{F}_{2^8}$. (iii) On such instances, with Filter 1 as the sole preconditioner, the residual landscape is rugged enough that the default—and even substantially enlarged—solver budgets do not recover a hidden solution. Far from a weakness, this delineation tells subsequent work exactly where to invest: an efficient MDS test, a scalable feasible-instance generator, and search operators or fitness shaping matched to the landscape; identifying these prerequisites is made possible precisely by the certificate-based viewpoint this paper introduces. Finally, the false-positive rate we observe is strong empirical evidence that

the necessary conditions are, in practice, close to sufficient; we report it as a measurement, which the tightness theorem above would later upgrade to a proof.

6. Conclusion

We have presented a filter-cascade preconditioner for the involutory MDS matrix transformation problem. The cascade combines one matrix-level and three permutation-level necessary conditions, ordered as Filters 1 through 4 in increasing computational cost; it gives instant infeasibility certificates and reduces the metaheuristic search space by orders of magnitude. Integration with three classical solvers (GA, PSO, SA) is straightforward. The three worked examples in Section 4 illustrate the full range of cascade outcomes: instant infeasibility (Example A), direct resolution after permutation-level pruning (Example B), and substantial cascade reduction on a 6×6 search space of 720 candidates that collapses to a single involutory permutation (Example C). The cascade fills a gap left by both direct construction methods (which do not address transformation of a given M) and metaheuristics (which lack infeasibility certificates).

Several directions remain open. The first is scaling to $n \geq 12$ and larger fields, where the survivor set after Filter 4 will be large enough that the metaheuristic component genuinely matters. A second is a tightness theorem characterising matrices for which the cascade filters are jointly sufficient, not just necessary — and, relatedly, identifying structural families of M (such as those for which the solution Q is forced to be an involution permutation) on which additional filters such as the palindrome of χ_M become valid. A third is hybrid construction: combining the cascade with the representative-matrix machinery of [12, 13] to enumerate all involutory transformations of given MDS families. A fourth is to replace the metaheuristic with a learned permutation predictor (for example a Sinkhorn network) trained on cascade-generated examples; the cascade then plays a double role as training-data source and necessary-condition regulariser.

Reproducibility

All numerical results in this paper—the three worked examples, the stratification of Section 4.5, the $n = 8$ study of Section 4.6, the family comparison of Section 4.7, and the diagnostics of Sections 4.8–4.9—are produced by a single modular Python implementation released with this paper. The package separates finite-field arithmetic ($\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^8}$ via log/exp tables), matrix operations over $\mathbb{F}_{2^m}$, the four filters, the cascade controller, and the GA/PSO/SA solvers, and reproduces every table from a fixed random seed. No reported value is hard-coded: each is recomputed and, where the article states an expected number, asserted against it. The code, experiment drivers, and machine-readable outputs (JSON) are not publicly shared but can be made available by the corresponding author upon reasonable request.

Acknowledgement

The authors thank the anonymous reviewers for their constructive comments, which led to the larger-scale experiments, the cross-family analysis, and the sharper positioning of the contribution.

REFERENCES

1. E.M. Bellfkih, et al., *Genetic Algorithm-Based Approach for Discovering Involutory MDS Matrices*, in 14th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1–6, IEEE, 2023.
2. E.M. Bellfkih, et al., *Genetic Algorithm-Based Method for Discovering Involutory MDS Matrices*, Computational and Mathematical Methods, vol. 2023, no. 1, 5951901, 2023.
3. E.M. Bellfkih, et al., *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, pp. 1059–1066, 2022.

4. R. Eberhart, and J. Kennedy, *A new optimizer using particle swarm theory*, in MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, IEEE, 1995.
5. S. Dhouib, D. Kallel, N. Beji, and S. Dhouib, *A Computational Time Analysis of Dhouib-Matrix-SPP versus Particle Swarm Optimization Metaheuristics for Grid-based Path Planning*, Statistics, Optimization & Information Computing, vol. 15, no. 4, pp. 2575–2586, 2026.
6. B. MERYEM, T. Abdelfattah, H. Imad, and R. Yassir, *OGA-Apriori: An Optimized Genetic Algorithm Approach for Enhanced Frequent Itemset Mining*, Statistics, Optimization & Information Computing, vol. 14, no. 6, pp. 3144–3161, 2025.
7. F. Marini, and B. Walczak, *Particle swarm optimization (PSO). A tutorial*, Chemometrics and Intelligent Laboratory Systems, vol. 149, pp. 153–165, 2015.
8. M. Jain, et al., *An overview of variants and advancements of PSO algorithm*, Applied Sciences, vol. 12, no. 17, 8392, 2022.
9. F.J. MacWilliams, and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, vol. 16, Elsevier, 1977.
10. K. Khoo, et al., *FOAM: Searching for Hardware-Optimal SPN Structures and Components with a Fair Comparison*, in Cryptographic Hardware and Embedded Systems, 2014.
11. S.M. Sim, et al., *Lightweight MDS Involution Matrices*, in Fast Software Encryption (FSE), 2015.
12. Y. Kumar, P.R. Mishra, S. Samanta, K. Gupta, and A. Gaur, *Construction of all MDS and involutory MDS matrices*, arXiv:2403.10372, 2024.
13. Y. Kumar, P.R. Mishra, S. Samanta, and A. Gaur, *A systematic construction approach for all* 4×4 *involutory MDS matrices*, Journal of Applied Mathematics and Computing, vol. 70, pp. 4677–4697, 2024.
14. Y. Kumar, S. Samanta, and A. Gaur, *New Insights into Involutory and Orthogonal MDS Matrices*, arXiv:2510.05766, 2025.
15. Y. Kumar, S. Samanta, P.R. Mishra, and A. Gaur, *On the study of semi-involutory and semi-orthogonal matrices*, Finite Fields and Their Applications, vol. 110, 102730, 2026.
16. K. Gupta, and I.G. Ray, *Cryptographically significant MDS matrices based on circulant and circulant-like matrices for lightweight applications*, Cryptography and Communications, vol. 7, pp. 257–287, 2014.
17. K. Gupta, and I.G. Ray, *On Constructions of MDS Matrices From Circulant-Like Matrices For Lightweight Cryptography*, 2014.
18. K. Gupta, and I.G. Ray, *On Constructions of MDS Matrices from Companion Matrices for Lightweight Cryptography*, 2013.
19. K. Gupta, S. Pandey, and A. Venkateswarlu, *Towards a general construction of recursive MDS diffusion layers*, Designs, Codes and Cryptography, vol. 82, pp. 179–195, 2016.
20. K. Gupta, S. Pandey, and S. Samanta, *A Few Negative Results on Constructions of MDS Matrices Using Low XOR Matrices*, 2019.
21. K. Gupta, S. Pandey, and S. Samanta, *On the Direct Construction of MDS and Near-MDS Matrices*, arXiv:2306.12848, 2023.
22. K. Gupta, S. Pandey, and S. Samanta, *On the construction of near-MDS matrices*, Cryptography and Communications, vol. 16, pp. 249–283, 2024.
23. M. Liu, and S.M. Sim, *Lightweight MDS Generalized Circulant Matrices*, IACR Cryptology ePrint Archive, vol. 2016, 186, 2016.
24. T. Chatterjee, and A. Laha, *A note on semi-orthogonal (G-matrix) and semi-involutory MDS matrices*, Finite Fields and Their Applications, vol. 92, 102279, 2023.
25. T. Chatterjee, and A. Laha, *A note on cyclic MDS and non-MDS matrices*, Cryptography and Communications, vol. 17, pp. 1109–1124, 2024.
26. T. Chatterjee, and A. Laha, *A note on MDS Property of Circulant Matrices*, arXiv:2406.16973, 2024.
27. T. Chatterjee, and A. Laha, *On MDS Property of g-Circulant Matrices*, arXiv:2406.15872, 2024.
28. G. Tuncay, F.B. Sakallı, M.K. Pehlivanoğlu, G.G. Yilmazgüç, S. Akleylek, and M.T. Sakallı, *A new hybrid method combining search and direct based construction ideas to generate all* 4×4 *involutory MDS matrices over binary field extensions*, PeerJ Computer Science, vol. 9, 2023.
29. M.K. Pehlivanoğlu, and M.A. Demir, *Optimizing implementations of linear layers using two and higher input XOR gates*, PeerJ Computer Science, vol. 10, 2024.
30. M.K. Pehlivanoğlu, F.B. Sakallı, S. Akleylek, and M.T. Sakallı, *On the Construction of New Lightweight Involutory MDS Matrices in Generalized Subfield Form*, IEEE Access, vol. 11, pp. 32708–32715, 2023.
31. 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, pp. 348–355, 2018.
32. M.K. Pehlivanoğlu, S. Akleylek, M.T. Sakallı, and N. Duru, *Generating MDS Matrices in Toeplitz Form by Using Generalised Hadamard Form*, in 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), pp. 81–86, 2018.
33. M.T. Sakallı, S. Akleylek, K. Akkanat, and V. Rijmen, *On the automorphisms and isomorphisms of MDS matrices and their efficient implementations*, Turkish Journal of Electrical Engineering and Computer Sciences, vol. 28, pp. 275–287, 2020.
34. S. Duval, and G. Leurent, *MDS matrices with lightweight circuits*, IACR Transactions on Symmetric Cryptology, vol. 2018, no. 2, pp. 48–78, 2018.
35. S. Li, S. Sun, C. Li, Z. Wei, and L. Hu, *Constructing Low-latency Involutory MDS Matrices with Lightweight Circuit*, IACR Cryptology ePrint Archive, vol. 2019, 107, 2019.
36. Q. Li, B. Wu, and Z. Liu, *Direct Constructions of (Involutory) MDS Matrices from Block Vandermonde and Cauchy-Like Matrices*, 2018.
37. M. Sajadieh, and M. Mousavi, *Construction of MDS matrices from generalized Feistel structures*, Designs, Codes and Cryptography, vol. 89, pp. 1433–1452, 2021.
38. M. Mousavi, A. Zaghian, and M. Esmaeili, *Involutory-Multiple-Lightweight MDS Matrices based on Cauchy-type Matrices*, Advances in Mathematics of Communications, vol. 15, pp. 589–610, 2021.
39. N. Mohsenifar, and M. Sajadieh, *Introducing a new connection between the entries of MDS matrices constructed by generalized Cauchy matrices in* $GF(2^q)$, Journal of Applied Mathematics and Computing, vol. 69, pp. 3871–3891, 2023.
40. S. Ali, A.A. Khan, and A. Kesarwani, *On the construction of Cauchy MDS matrices over Galois rings via nilpotent elements and Frobenius maps*, arXiv:2512.19306, 2025.

41. G.G. Güzel, M.T. Sakallı, S. Akleylek, V. Rijmen, and Y. Çengellenmiş, *A new matrix form to generate all* 3×3 *involutory MDS matrices over* $\mathbb{F}_{2^m}$, Information Processing Letters, vol. 147, pp. 61–68, 2019.
42. S. Wang, Y. Li, S. Tian, and X. Zeng, *Four by four MDS matrices with the fewest XOR gates based on words*, Advances in Mathematics of Communications, vol. 17, pp. 845–872, 2023.
43. Y. Yang, X. Zeng, and S. Wang, *Construction of lightweight involutory MDS matrices*, Designs, Codes and Cryptography, vol. 89, pp. 1453–1483, 2021.
44. J. Bai, Y. Sun, and D. Wang, *On the Construction of Involutory MDS Matrices over* $\mathbb{F}_{2^m}$, Journal of Systems Science and Complexity, vol. 33, pp. 836–848, 2019.
45. H. Xu, Y. Zheng, and X. Lai, *Construction of perfect diffusion layers from linear feedback shift registers*, IET Information Security, vol. 9, pp. 127–135, 2015.
46. F.B. Sakallı, Ö. Aydın, G. Tuncay, M.K. Pehlivanoğlu, G.G. Güzel, and M.T. Sakallı, *On Lightweight* 4×4 *MDS Matrices over Binary Field Extensions*, 2020.
47. D. Lin, Z. Xiang, X. Zeng, and S. Zhang, *A Framework to Optimize Implementations of Matrices*, 2021.
48. Z. Xiang, X. Zeng, D. Lin, Z. Bao, and S. Zhang, *Optimizing Implementations of Linear Layers*, IACR Transactions on Symmetric Cryptology, vol. 2020, pp. 120–145, 2020.
49. Q. Liu, Z. Zhao, and M. Wang, *Improved Heuristics for Low-latency Implementations of Linear Layers*, IACR Cryptology ePrint Archive, vol. 2023, 174, 2023.
50. P.R. Mishra, Y. Kumar, S. Samanta, and A. Gaur, *A New Algorithm for Computing Branch Number of Non-Singular Matrices over Finite Fields*, arXiv:2405.07007, 2024.
51. S. Samanta, *On the counting of involutory MDS matrices*, Cryptography and Communications, vol. 18, pp. 5–25, 2024.
52. W.C. Huffman, and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, 2010.
53. I. Márquez-Corbella, and R. Pellikaan, *A characterization of MDS codes that have an error correcting pair*, Finite Fields and Their Applications, vol. 40, pp. 224–245, 2016.
54. I. Muchtadi-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, 012030, 2021.