

Article

Not peer-reviewed version

PSO-Guided Construction of MRD Codes for Rank Metrics

Behnam Dehghani ^{*,†} and [Amineh Sakhaie](#) [†]

Posted Date: 22 August 2025

doi: 10.20944/preprints202504.0883.v3

Keywords: Maximum Rank-Distance Codes; Particle Swarm Optimization; Rank Metrics; Error-Correcting Codes; Combinatorial Optimization



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

PSO-Guided Construction of MRD Codes for Rank Metrics

Behnam Dehghani ^{1,*} and Amineh Sakhaie ^{2,†}¹ Minho University² Lisbon University

* Correspondence: pg41104@alunos.uminho.pt or behnam.dehghani@gmail.com; Tel. +351932297030

† These authors contributed equally to this work.

Abstract

Maximum Rank-Distance (MRD) codes are a class of optimal error-correcting codes that achieve the Singleton-like bound for rank metric, making them invaluable in applications such as network coding, cryptography, and distributed storage. While algebraic constructions of MRD codes (e.g., Gabidulin codes) are well-studied for specific parameters, a comprehensive theory for their existence and structure over arbitrary finite fields remains an open challenge. Recent advances have expanded MRD research to include twisted, scattered, convolutional, and machine-learning-aided approaches, yet many parameter regimes remain unexplored. This paper introduces a computational optimization framework for constructing MRD codes using Particle Swarm Optimization (PSO), a bio-inspired metaheuristic algorithm adept at navigating high-dimensional, non-linear, and discrete search spaces. Unlike traditional algebraic methods, our approach does not rely on prescribed algebraic structures; instead, it systematically explores the space of possible generator matrices to identify MRD configurations, particularly in cases where theoretical constructions are unknown. Key contributions include: (1) a tailored finite-field PSO formulation that encodes rank-metric constraints into the optimization process, with explicit parameter control to address convergence speed and global optimality; (2) a theoretical analysis of the adaptability of PSO to MRD construction in complex search landscapes, supported by experiments demonstrating its ability to find codes beyond classical families; and (3) an open-source Python toolkit for MRD code discovery, enabling full reproducibility and extension to other rank-metric scenarios. The proposed method complements established theory while opening new avenues for hybrid metaheuristic–algebraic and machine learning–aided MRD code construction.

Keywords: maximum rank-distance codes; particle swarm optimization; rank metrics; error-correcting codes; combinatorial optimization

1. Introduction

Maximum Rank-Distance (MRD) codes are optimal error-correcting codes for the rank metric, achieving the Singleton-like bound and offering maximum error resilience in rank-metric channels. Since the foundational works of Delsarte in 1978 [1] and Gabidulin in 1985 [2], they have been central to applications in network coding, cryptography, and distributed storage [3].

Despite their theoretical appeal, MRD code construction remains challenging. Classical families, such as Gabidulin and their generalisations [4], cover many, but not all, parameter regimes. Recent advances, including twisted Gabidulin [5], generalised twisted Gabidulin [6], scattered-subspace constructions [7], and Maximum Flag-Rank Distance codes [8], have revealed inequivalent MRD codes with distinct structural and weight properties. Yet, a comprehensive existence theory for arbitrary field sizes and dimensions is still lacking.

In parallel, computational approaches have emerged as a complementary route to algebraic methods. Classification efforts for small parameters [9] and heuristic searches, most notably genetic

algorithms for Hamming-metric codes [10,11], have shown that metaheuristics can effectively navigate discrete code spaces. Deep-learning-based code design [12,13] and learning-driven inverse problem solving [14] further illustrate the potential of data-driven exploration when explicit constructions are elusive. However, to our knowledge, Particle Swarm Optimization (PSO) [15,16] has not been applied to MRD code construction.

PSO is a population-based metaheuristic known for efficiently exploring high-dimensional, non-linear, and multimodal search spaces. By encoding generator matrices directly over finite fields and embedding rank-metric constraints into the objective, PSO can adaptively balance exploration (discovering new code structures) and exploitation (refining promising candidates). This makes it especially attractive for MRD regimes where algebraic constructions are unknown, convergence behaviour must be analysed, and global optima are difficult to guarantee.

In this work, we:

1. **Formulate** MRD code construction as a constrained combinatorial optimisation problem over $\mathbb{F}_q^{k \times n}$.
2. **Design** a finite-field PSO variant with rank-aware velocity clamping, adaptive penalties, and structured seeding from Gabidulin codes.
3. **Provide** a theoretical analysis of the PSO-guided search process in the MRD code context, including its adaptability to high-dimensional and nonlinear search spaces.
4. **Release** an open-source Python implementation to support reproducibility and to facilitate further exploration of MRD code existence in uncharted parameter regimes.

This contribution bridges a gap between algebraic theory and heuristic search, positioning PSO as a flexible, extensible framework for MRD code discovery and potentially inspiring analogous applications in machine learning-aided code design.

2. Related Work

2.1. Evolution and Recent Advances in Rank Metric and MRD Code Constructions

The theory of rank-metric codes began with the foundational work of Delsarte [1], who introduced the rank metric and exhibited MRD codes via bilinear forms, in parallel with the role that MDS codes play in the Hamming metric [17]. Gabidulin then provided a general polynomial-evaluation construction [2], later extended in scope and applications [3,4].

Beyond these classical families, inequivalent MRD codes arise from twisted and generalized twisted constructions [5,6], as well as skew polynomial frameworks [18]. Geometric methods based on scattered linear sets yield additional MRD families [7], and there are explicit constructions outside the $n \leq m$ regime such as the $n = \frac{3m}{2}$ case for even m [19,20]. Recent work on maximum flag-rank distance (MFRD) codes provides new linear spaces with larger generalized rank weights and new structural invariants [8]. In a related direction, MRD convolutional codes have been investigated as a streaming or time-varying analogue of block MRD codes; the recent work by Napp *et al.* [21] constructs novel classes of such convolutional codes for a broad set of parameters, extending MRD theory into the convolutional domain.

Concurrently, several construction paradigms continue to expand the frontier. A 2024 switching framework builds MRD codes via puncturing and product operations, enabling transitions among inequivalent families [22]. In the sum-rank setting, which unifies Hamming and rank metrics, there are fresh explicit MSRD constructions and decoding-friendly variants [23–25]. Very recent preprints report additional MRD and sum-rank constructions that leverage subspace designs and orthogonal space methods [26].

The cumulative effect of these developments is a significant expansion of MRD code theory, from bilinear and polynomial constructions to geometric, algebraic-combinatorial, and now convolutional frameworks, with ongoing discoveries continuing to reshape the boundaries of known existence regions.

2.2. Heuristic, Learning-Based, and Automated Searches for MRD Codes

Computational search complements algebraic theory where existence questions remain open. Classification for small parameters combines structure with exhaustive or guided enumeration [9]. In the broader coding landscape, metaheuristics such as genetic algorithms have been effective for discovering or approximating good codes in Hamming settings [10,11], suggesting portability to rank-metric search.

Learning-based design has emerged for linear codes, with neural constructions and optimization-oriented frameworks that learn generator structure [12,13]. Although most such results target Hamming or Euclidean metrics, they indicate a pathway for data-driven construction in rank-metric problems. Complementary perspectives from inverse-problem machine learning also demonstrate how learned models can replace explicit derivations when closed-form structures are unavailable [14]. Beyond pure code design, AI-driven rank-metric evaluation has also been explored in other domains; for example, Chen *et al.* [27] implemented an AI-based MRD evaluation and prediction model (albeit for clinical data), illustrating the broader applicability of machine-learning MRD frameworks across fields.

Within this landscape, Particle Swarm Optimization [15,16] remains underexplored for MRD construction. Its balance of exploration and exploitation in high-dimensional, nonlinear spaces motivates the PSO-guided framework we develop in this work. Table 1 includes a comprehensive comparison.

Table 1. Comparison of MRD code construction approaches.

Approach Type	Key Examples	Strengths	Limitations	Applicability to Unexplored Regimes
Algebraic constructions	Gabidulin [2]; Twisted Gabidulin [5]; Generalized Twisted Gabidulin [6]; Geometric/scattered subspace [7,19,20]	Exact MRD guarantee when applicable; solid theory	Limited to specific (m, n, k, q) ; beyond known families can be intractable	Low: restricted by proven existence results
Other metaheuristics	Genetic algorithms [10,11]	Flexible search in discrete spaces; can add constraints	May need heavy tuning; convergence speed varies	Medium: adaptable yet can struggle in high dimensional rank-metric spaces
Learning-based / AI methods	Neural code design [12]; inverse problem ML [14]	Learns patterns from data; can generalize	Needs training data; no exact MRD guarantees in general	Medium: depends on data quality and diversity
Proposed PSO-guided approach	This work	No need for closed-form structure; rank-aware constraints; adaptive exploration and exploitation; seeding from known MRD codes	No formal global optimality proof; fitness evaluation cost	High: can target regimes without known algebraic constructions

3. Algebraic Background

This section reviews the concepts and notation that underlie maximum-rank-distance (MRD) codes and their construction. We also formalise the optimisation target that will be addressed by the PSO framework in later sections.

3.1. Rank Metric

Let $\mathbb{F}_q$ denote the finite field with q elements, where q is a prime power. The *rank metric* measures the distance between two matrices, or equivalently between vectors on an extension field, by the rank of their difference:

$$d(A, B) = \text{rank}_{\mathbb{F}_q}(A - B), \quad A, B \in \mathbb{F}_q^{m \times n}.$$

(1)

A rank-metric code $\mathcal{C}$ is a subset of $\mathbb{F}_q^{m \times n}$ used for error detection and correction when this metric is applied. Its minimum rank distance is

$$d_r = \min_{\substack{A, B \in \mathcal{C} \\ A \neq B}} \text{rank}_{\mathbb{F}_q}(A - B). \quad (2)$$

The code $\mathcal{C}$ is called $\mathbb{F}_q$ -linear if it is a k -dimensional subspace of $\mathbb{F}_q^{m \times n}$ with $k \in \{1, \dots, mn\}$. If $\mathcal{C}$ is $\mathbb{F}_q$ -linear it admits the generator-matrix form

$$\mathcal{C} = \{G\mathbf{u} : \mathbf{u} \in \mathbb{F}_q^{k \times 1}\}, \quad G \in \mathbb{F}_q^{k \times n}. \quad (3)$$

Each $\mathbf{c} \in \mathcal{C}$ is a codeword, viewed either as a vector over $\mathbb{F}_q$ or, via a fixed $\mathbb{F}_q$ -linear isomorphism $\Phi : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^{m \times n}$, as an $m \times n$ matrix over the base field.

3.2. Singleton Bound and Singleton-like Bound

Classical Hamming case. For a Hamming-metric block code of length n , dimension k , and minimum distance d , the classical Singleton bound is [17]:

$$d \leq n - k + 1. \quad (4)$$

Rank-metric analogue. For $\mathcal{C} \subseteq \mathbb{F}_q^{m \times n}$ with minimum rank distance d and $\mathbb{F}_q$ -dimension k , Delsarte and Gabidulin [1–3] proved the Singleton-type bounds:

$$|\mathcal{C}| \leq q^{m(n-d+1)}, \quad (5)$$

$$k \leq m(n-d+1), \quad (6)$$

$$d \leq \min\{m, n\} - k + 1. \quad (7)$$

3.3. Maximum Rank-Distance (MRD) Codes

A code $\mathcal{C} \subseteq \mathbb{F}_q^{m \times n}$ is an MRD code when it achieves equality in (7):

$$|\mathcal{C}| = q^{m(n-d+1)} \iff k = m(n-d+1) \iff d = \min\{m, n\} - k + 1. \quad (8)$$

Lemma 1. Let $G \in \mathbb{F}_q^{k \times n}$ generate a $\mathbb{F}_q$ -linear code $\mathcal{C}$. Then $\mathcal{C}$ is MRD if and only if

$$\text{rank}_{\mathbb{F}_q}(YG) = k \quad \text{for every } Y \in \mathbb{F}_q^{k \times n} \text{ with } \text{rank}_{\mathbb{F}_q}(Y) = k. \quad (9)$$

Hence an MRD code can be identified by the full-rank preservation of its generator matrix under premultiplication by any full-rank $k \times n$ matrix over the base field [3].

3.4. Gabidulin Codes

Let q be a prime power and $m, n, k \in \mathbb{N}$ with $k \leq n \leq m$ (since for Gabidulin codes the length n must not exceed the extension degree m , and the dimension k must not exceed the length) [3]. Choose

$$\alpha = (\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_n) \in \mathbb{F}_q^{1 \times n}, \quad \text{with entries } \mathbb{F}_q\text{-linearly independent.} \quad (10)$$

For a q -linearised polynomial

$$P(x) = \sum_{i=0}^{k-1} a_i x^{[i]}, \quad x^{[i]} := x^{q^i}, \quad a_i \in \mathbb{F}_q, \quad (11)$$

define the evaluation map:

$$\text{Ev}_\alpha(P) = \begin{pmatrix} P(\alpha_1) & P(\alpha_2) & \cdots & P(\alpha_n) \end{pmatrix} \in \mathbb{F}_q^{1 \times n}. \quad (12)$$

The *Gabidulin code* is then:

$$\text{Gab}_q(n, k, m) = \{ \text{Ev}_\alpha(P) : \deg_q(P) < k \}, \quad (13)$$

forming an $\mathbb{F}_q$ -linear $[n, k, d]_{q^m}$ code with

$$d = n - k + 1, \quad |\text{Gab}_q(n, k, m)| = q^{mk}. \quad (14)$$

Its generator matrix is the Moore (or σ -Vandermonde) matrix:

$$G_{\text{Gab}} = \begin{pmatrix} \alpha_1^{[0]} & \alpha_2^{[0]} & \cdots & \alpha_n^{[0]} \\ \alpha_1^{[1]} & \alpha_2^{[1]} & \cdots & \alpha_n^{[1]} \\ \vdots & \vdots & & \vdots \\ \alpha_1^{[k-1]} & \alpha_2^{[k-1]} & \cdots & \alpha_n^{[k-1]} \end{pmatrix} \in \mathbb{F}_q^{k \times n}. \quad (15)$$

Since G_{Gab} satisfies Lemma 1, $\text{Gab}_q(n, k, m)$ is MRD.

3.5. Optimisation Viewpoint for MRD Construction

In the context of this work, the construction problem can be expressed as:

$$\max_{G \in \mathbb{F}_q^{k \times n}} \min_{\mathbf{u} \in \mathbb{F}_q^k \setminus \{0\}} \text{rank}_{\mathbb{F}_q}(\Phi(G\mathbf{u})) \quad \text{s.t.} \quad \text{rank}_{\mathbb{F}_q}(G) = k. \quad (16)$$

The MRD condition is achieved when the inner minimum equals the bound in (7). This formalisation directly motivates the PSO-based search strategy described in the theoretical framework.

3.6. Remark on computational complexity

Evaluating (16) exactly requires computing $\binom{q^k-1}{1}$ rank operations on $m \times n$ matrices per candidate G , each costing $\mathcal{O}(\min(m, n)^3)$ with Gaussian elimination. This high cost underlines the need for efficient search heuristics.

4. Theoretical Framework

This section lays out the foundations for using PSO to construct MRD codes. We first summarise PSO, then formulate code construction as a constrained combinatorial optimisation problem, and finally discuss its theoretical suitability for this task.

4.1. Particle Swarm Optimisation

Particle swarm optimisation (PSO) is a population-based metaheuristic inspired by the collective motion of biological swarms [15,16]. A swarm of S particles explores the search space; particle i keeps track of its position $\mathbf{x}_i$, velocity $\mathbf{v}_i$, personal best $\mathbf{p}_i$, and the global best $\mathbf{g}$ found so far.

The standard updates are

$$\mathbf{v}_i(t+1) = w \mathbf{v}_i(t) + c_1 r_{1,i}(t) (\mathbf{p}_i - \mathbf{x}_i(t)) + c_2 r_{2,i}(t) (\mathbf{g} - \mathbf{x}_i(t)), \quad (17)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1), \quad (18)$$

where $r_{1,i}(t), r_{2,i}(t) \stackrel{\text{i.i.d.}}{\sim} \mathcal{U}(0, 1)$ are independent scalars (or component-wise vectors) drawn anew each iteration, w is the inertia weight, and c_1, c_2 are the cognitive and social acceleration coefficients.

4.2. Problem Formulation

Let $[n, k, d]_{q^m}$ be the target parameters of an MRD vector code. We seek a generator matrix $G \in \mathbb{F}_q^{k \times n}$ such that the $\mathbb{F}_q$ -linear code $\mathcal{C}$ given by (3) attains minimum rank distance

$$d_{\min}(\mathcal{C}) = \min_{\mathbf{u} \in \mathbb{F}_q^k \setminus \{0\}} \text{rank}_{\mathbb{F}_q}(\Phi(G\mathbf{u})) \geq d, \quad (19)$$

where $\Phi : \mathbb{F}_{q^m}^n \rightarrow \mathbb{F}_q^{m \times n}$ is the fixed $\mathbb{F}_q$ -linear isomorphism that converts each length- n vector over the extension field into its $m \times n$ matrix representation [3]. The rank in (19) is taken over $\mathbb{F}_q$ on this unfolded matrix.

The construction task can thus be expressed as the optimisation problem (16) in Section 3, with the additional MRD constraint from (7).

4.3. Objective Function

By (8), a linear rank-metric code is MRD precisely when its minimum rank distance equals the Singleton-like bound. We therefore define the fitness

$$F(G) = \min_{\mathbf{u} \in \mathbb{F}_{q^m}^k \setminus \{0\}} \text{rank}_{\mathbb{F}_q}(\Phi(G\mathbf{u})) \quad (20)$$

and maximise $F(G)$ subject to $F(G) \geq d$. If multiple candidates achieve the primary target, secondary criteria such as larger generalised rank weights [6] or sparsity of G can be incorporated.

4.4. Search Space and Constraints

The search domain consists of all full-rank $k \times n$ matrices over $\mathbb{F}_q$. Constraint handling is implemented via: **(i)** a penalty term reducing the score when $F(G) < d$, and **(ii)** a repair routine that replaces dependent rows with random full-rank ones if a velocity update introduces rank deficiency. These mechanisms help maintain feasibility while allowing occasional exploration into near-feasible regions.

4.5. Representation and Initialisation

Each PSO particle represents a candidate generator matrix G in (20). Positions are initialised either: **(a)** uniformly at random over $\mathbb{F}_q$, subject to full rank, or **(b)** as low-rank perturbations of Gabidulin matrices [10,11,16] to seed the swarm with known good structures while preserving diversity.

4.6. Theoretical Adaptability of PSO

PSO is well-suited for MRD construction for several reasons:

- **High-dimensional exploration:** Generator matrices over $\mathbb{F}_q$ can be large, making the search space exponential in kn . PSO's population-based search can maintain coverage across many distant regions simultaneously.
- **Nonlinear, discrete landscape:** The rank-distance function is highly non-smooth and discrete. Probabilistic velocity updates allow PSO to traverse such landscapes without relying on gradient information.
- **Convergence control:** By adjusting w , c_1 , c_2 , and velocity clamping, the algorithm can trade exploration for exploitation, mitigating premature convergence to sub-optimal matrices.
- **Global optimality considerations:** While PSO does not guarantee global optima, the combination of structured seeding, adaptive penalties, and rank-based diversity control reduces the risk of stagnation in poor local optima.

4.7. Parameter Roles and Selection

The main PSO parameters are:

- S : swarm size, typically scaled with problem dimension kn .
- $T_{\max}$: maximum iterations, influencing total search effort.

- w : inertia weight, controlling momentum. Larger w promotes exploration, while smaller w promotes convergence.
- c_1, c_2 : cognitive and social coefficients, balancing attraction to personal vs global bests.
- $r_{\max}$: velocity-rank clamp, limiting disruptive changes to generator matrices.

Stable parameter regions from PSO theory [28–30] can be adapted to the finite-field setting by treating velocity updates as probabilistic component changes rather than real-valued vectors.

4.8. PSO Algorithm for MRD Code Construction

At every iteration the swarm updates velocities and positions via (17)–(18). After each move, the fitness (20) is evaluated; personal and global bests are updated accordingly, steering the particles toward matrices satisfying the MRD property.

This completes the theoretical basis. Section 5 translates these ideas into a concrete PSO-guided construction algorithm.

5. Proposed PSO-Guided Construction Method

This section details the PSO-guided method for constructing MRD codes, emphasising the finite-field adaptations and constraint-handling strategies that differentiate it from a vanilla PSO. The method directly implements the optimisation model in (16).

5.1. Initialisation

The swarm consists of N particles, each representing a $k \times n$ generator matrix $G^{(j)} \in \mathbb{F}_q^{k \times n}$. Two complementary strategies ensure both diversity and structural quality:

(i) **Purely random sampling.** For approximately half of the particles, entries are drawn independently from $\mathbb{F}_q$ until $\text{rank}_{\mathbb{F}_q}(G) = k$. A practical procedure is:

- Fix a normal basis $\{\beta, \beta^{[q]}, \dots, \beta^{[q^{m-1}]}\}$ of $\mathbb{F}_{q^m}$.
- Fill a $k \times n$ array with uniformly random $\mathbb{F}_q$ elements.
- Map each m -tuple of $\mathbb{F}_q$ -coefficients to a field element $\sum_{i=0}^{m-1} a_i \beta^{[q^i]} \in \mathbb{F}_{q^m}$.
- If full $\mathbb{F}_q$ -rank is not achieved, repeat.

(ii) **Perturbed Gabidulin seeds.** The remaining particles are seeded as

$$G^{(j)} = G_{\text{Gab}} + R^{(j)}, \quad \text{rank}_{\mathbb{F}_q}(R^{(j)}) \leq r_{\max} \ll k, \quad (21)$$

where G_{Gab} is a Gabidulin generator [3] and $R^{(j)} = \mathbf{a}^{(j)} \mathbf{b}^{(j)\top}$ with $\mathbf{a}^{(j)} \in \mathbb{F}_q^{k \times 1}$ and $\mathbf{b}^{(j)} \in \mathbb{F}_q^{n \times 1}$. This preserves known MRD structure while introducing controlled diversity.

Initial velocities $\mathbf{v}^{(j)}(0) \in \mathbb{F}_q^{k \times n}$ are generated independently and uniformly, ensuring no bias toward initial positions.

5.2. Fitness Evaluation

Each particle's primary fitness is $F(G)$ from (20). Feasible particles satisfy $F(G) \geq d$; ties are resolved lexicographically:

- maximise $F(G)$,
- then maximise the second-smallest rank (proxy for the second generalised rank weight),
- then minimise the Hamming weight of G to favour sparsity.

This composite score is applied consistently to personal- and global-best updates.

5.3. Velocity and Position Update

Equations (17)–(18) govern updates, adapted for $\mathbb{F}_q$:

- Addition is over $\mathbb{F}_q$, component-wise.
- Random scalars $r_{1,i}, r_{2,i}$ define per-entry probabilities of applying cognitive/social terms.

- Velocity rank is clamped to $\text{rank}_{\mathbb{F}_q}(\mathbf{v}_i) \leq r_{\max}$ to avoid destructive changes.

5.4. Constraint Handling

Two mechanisms ensure feasibility:

- **Penalty:** subtract $\lambda[d - F(G)]$ when $F(G) < d$, with λ increasing periodically to discourage persistent infeasibility.
- **Repair:** if $\text{rank}_{\mathbb{F}_q}(G) < k$, iteratively replace dependent rows with random full-rank rows.

5.5. Termination Criteria

Stop if:

- $t = T_{\max}$,
- a particle attains $F(G) = d$ and meets all secondary criteria,
- no global-best improvement for τ iterations, $\tau \approx 0.1T_{\max}$.

5.6. Algorithm Summary

Algorithm 1 Finite-Field PSO for MRD Code Construction

- 1: Initialise N particles: half random, half perturbed Gabidulin seeds.
 - 2: Assign random finite-field velocities.
 - 3: **for** $t = 1$ to $T_{\max}$ **do**
 - 4: **for** each particle j **do**
 - 5: Evaluate $F(G^{(j)})$; apply tie-breakers if $F(G^{(j)}) \geq d$.
 - 6: Update personal best p_j if improved.
 - 7: **end for**
 - 8: Identify global best g .
 - 9: **for** each particle j **do**
 - 10: Update $\mathbf{v}^{(j)}$ via (17) (finite-field form).
 - 11: Rank-clamp $\mathbf{v}^{(j)}$ if needed.
 - 12: Update $G^{(j)} \leftarrow G^{(j)} + \mathbf{v}^{(j)}$ over $\mathbb{F}_q$.
 - 13: Apply repair if $\text{rank}(G^{(j)}) < k$.
 - 14: **end for**
 - 15: **if** termination criterion met **then break**.
 - 16: **end if**
 - 17: **end for**
 - 18: **return** best-found G^* .
-

Step Interaction Recap. The algorithm begins by generating a diverse swarm of candidate generator matrices, with half sampled randomly under full-rank constraints and the remainder obtained by low-rank perturbations of known Gabidulin codes. This initialisation ensures a balance between structural guidance and exploratory diversity. Each particle is assigned a random finite-field velocity, initiating movement through the search space. At each iteration, velocities are updated using finite-field PSO rules, where inertia, cognitive, and social terms control the trade-off between exploration and exploitation. Velocity-rank clamping limits disruptive changes, while penalties dynamically reduce the fitness of infeasible solutions to steer the swarm toward MRD-compliant regions. After each position update, repair routines restore full rank when necessary, ensuring particles remain viable candidates. This interplay between guided initialisation, controlled velocity updates, adaptive penalties, and corrective repair maintains both feasibility and diversity, enabling the swarm to converge toward high-quality MRD generator matrices. The process terminates when a stopping condition is met, such as reaching the maximum iterations, satisfying all MRD criteria, or stalling in improvement, and the best-found generator matrix is returned as the output.

5.7. Implementation and Reproducibility

A complete Python implementation, mirroring Algorithm 1, is provided at https://github.com/behnamde/pso_mrd [31], using the `galois` library [32] for $\mathbb{F}_q$ arithmetic. Key features include:

- Exact reproduction of finite-field PSO updates, rank clamping, and probabilistic velocity application.
- Configurable seeding, penalty scheduling, and repair.
- Parameter exposure $(N, T_{\max}, w, c_1, c_2, r_{\max}, \lambda, \tau)$ for systematic study.
- Deterministic mode via random seed control for reproducibility.

This codebase enables researchers to replicate the results, experiment with parameter regimes, and extend the framework to related rank- or sum-rank-metric problems.

5.8. Computational Complexity

We summarise the cost of one PSO iteration over N particles for the proposed finite-field scheme. Throughout, let

$$m, n, k \in \mathbb{N}, \quad q \text{ a prime power}, \quad d \text{ the target rank distance.}$$

Denote by $C_{\text{mul}}^{(q^m)}$ the unit cost of a multiplication in $\mathbb{F}_{q^m}$ and by $C_{\text{rank}}(m, n)$ the cost of computing the rank of an $m \times n$ matrix over $\mathbb{F}_q$ using Gaussian elimination:

$$C_{\text{rank}}(m, n) = \mathcal{O}(\min\{m, n\} m n) = \begin{cases} \mathcal{O}(m n^2), & n \leq m, \\ \mathcal{O}(n m^2), & m < n. \end{cases} \quad (22)$$

Per-particle fitness evaluation.

For a candidate $G \in \mathbb{F}_q^{k \times n}$, the fitness $F(G)$ in (20) requires:

- Forming $G\mathbf{u}$ in $\mathbb{F}_{q^m}^n$: $\mathcal{O}(kn C_{\text{mul}}^{(q^m)})$.
- Unfolding via Φ : $\mathcal{O}(mn)$ in $\mathbb{F}_q$.
- Rank computation over $\mathbb{F}_q$: $C_{\text{rank}}(m, n)$ from (22).

If evaluated exactly over all nonzero $\mathbf{u}$, the number of candidates is $Q := q^{mk} - 1$ when $\mathbf{u}$ is taken over $\mathbb{F}_{q^m}$. Hence the worst-case exact per-particle fitness cost is:

$$\mathcal{O}\left(Q \cdot (kn C_{\text{mul}}^{(q^m)} + mn + C_{\text{rank}}(m, n))\right). \quad (23)$$

In practice, a fixed probe budget $T_u \ll Q$ with early stopping is used, giving:

$$\mathcal{O}\left(T_u \cdot (kn C_{\text{mul}}^{(q^m)} + mn + C_{\text{rank}}(m, n))\right). \quad (24)$$

Per-particle PSO update.

Velocity and position updates over $\mathbb{F}_q$ are linear in the number of entries:

$$\mathcal{O}(kn). \quad (25)$$

Rank-clamping of $\mathbf{v}$ costs $\mathcal{O}(\min\{k, n\} k n)$ unless low-rank factored forms are used, in which case:

$$\mathcal{O}(r_{\max}(k + n)), \quad (26)$$

with $r_{\max}$ the velocity-rank bound.

Repair and feasibility checks.

Checking $\text{rank}(G)$ matches (22); replacing dependent rows costs the same order.

Total per-iteration cost.

For dense velocities and probe budget T_u , the dominant cost is fitness:

$$\begin{aligned} \text{Per-iteration: } & \mathcal{O}\left(N T_u \cdot (kn C_{\text{mul}}^{(q^m)} + mn + C_{\text{rank}}(m, n))\right. \\ & \left. + N \cdot \min\{k, n\} kn\right), \\ \text{Total runtime over } T_{\text{max}}: & \mathcal{O}\left(T_{\text{max}} N T_u \cdot (kn C_{\text{mul}}^{(q^m)} + mn + C_{\text{rank}}(m, n))\right. \\ & \left. + T_{\text{max}} N \cdot \min\{k, n\} kn\right). \end{aligned} \quad (27)$$

Memory complexity.

Positions and velocities in dense form require:

$$\mathcal{O}(Nkn) \quad \text{field elements}, \quad (28)$$

while low-rank velocity storage with bound r_{max} uses:

$$\mathcal{O}(N r_{\text{max}}(k + n)). \quad (29)$$

Remarks.

- Exact fitness (23) is infeasible for large k ; the practical form (24) dominates runtime.
- For moderate m, n , $C_{\text{rank}}(m, n)$ is the main bottleneck.
- Low-rank velocity factorisation (26) can yield substantial speed-ups in large (k, n) regimes.

6. Discussion

The proposed PSO-guided construction method introduces a metaheuristic framework for identifying maximum-rank-distance (MRD) codes by leveraging the global search capabilities of particle swarm optimisation (PSO). Unlike purely algebraic methods, such as Gabidulin's construction [2,3], which rely on explicit closed-form derivations, PSO offers a flexible, data-driven approach that remains applicable when no known theoretical construction exists or when the target parameters fall outside established families.

In addition to the qualitative advantages discussed throughout this work, Table 1 in Section 2 provides a structured comparison between our PSO-guided framework, classical algebraic constructions, alternative metaheuristics, and recent AI-based approaches. This comparison underscores PSO's ability to operate in parameter regimes where algebraic existence results are unknown, while also maintaining flexibility to incorporate structural seeding and advanced constraint-handling. In contrast to algebraic techniques bound by proven construction theorems and learning-based frameworks dependent on extensive training data, our PSO formulation combines rank-metric awareness with adaptive exploration and exploitation, enabling efficient search over vast high-dimensional discrete spaces.

6.1. Advantages of the PSO Approach

- **Generality:** Applicable to a wide range of (m, n, k, q) parameters, including those beyond classical Gabidulin families [5,6,8] and more recent convolutional MRD frameworks [21].
- **Scalability:** Handles larger problem sizes by tuning swarm size N , inertia weight w , and evaluation frequency. Complexity analysis in Section 5.8 shows that low-rank velocity factorisation can substantially mitigate the $\mathcal{O}(mn^2)$ rank-computation bottleneck.
- **Adaptability:** Can seamlessly incorporate domain knowledge, such as Gabidulin-based seeding, without constraining the search to a narrow solution family.

- **Exploratory Capability:** Maintains population diversity, balancing the discovery of new code structures with the refinement of promising candidates in the inherently non-convex MRD search landscape.

6.2. Limitations and Challenges

- **Combinatorial Explosion:** The search space scales exponentially with kn , making convergence more challenging as problem size increases.
- **Fitness Evaluation Cost:** Computing $F(G)$ in (20) for all non-zero $\mathbf{u}$ is expensive, with complexity analysis in 5.8 confirming that rank evaluation dominates runtime for most parameter sets.
- **No Global-Optimality Guarantee:** As with other metaheuristics, PSO yields high-quality solutions but cannot guarantee global optimality; algebraic verification remains essential.
- **Initialisation Sensitivity:** Poor diversity in the initial swarm can lead to premature convergence, underscoring the value of structured seeding.

6.3. Future Work

- **Hybrid Methods:** Combine PSO with algebraic insights or complementary metaheuristics (e.g., genetic algorithms [10,11]) to increase robustness.
- **Enhanced Objectives:** Extend $F(G)$ in (20) to capture generalised rank weights [6], decoding complexity, or convolutional rank constraints [21].
- **Parallelisation:** Use GPU and distributed architectures to accelerate velocity updates and rank evaluations, as suggested by the complexity model in 5.8.
- **Applied Evaluation:** Deploy discovered codes in network coding, distributed storage, or AI-driven MRD evaluation frameworks [27] to assess performance under realistic channel models.
- **Quantum-Inspired Optimisation:** Investigate PSO variants leveraging quantum principles for potential acceleration in discrete combinatorial search spaces.

7. Conclusion

We have formulated MRD-code construction as a constrained combinatorial optimisation problem over $\mathbb{F}_q^{k \times n}$ and proposed a PSO-based algorithm tailored to the finite-field rank-metric setting. The framework incorporates rank-aware velocity clamping, adaptive penalties, and structured seeding from Gabidulin codes, allowing it to search efficiently in high-dimensional, non-linear, discrete spaces.

The method offers a general, extensible approach that can explore MRD parameter regimes not covered by existing algebraic constructions, including convolutional and emerging AI-driven contexts. While its adaptability and exploratory capability make it a promising addition to the MRD-construction toolbox, the approach still faces scalability challenges due to the combinatorial size of the search space and the high cost of exact fitness evaluation. The complexity analysis in 5.8 quantifies these bottlenecks and highlights potential optimisations, such as low-rank velocity updates and parallel evaluation.

Future research will focus on integrating deeper algebraic structure into the search, refining constraint-handling mechanisms, and exploring hybrid frameworks that combine PSO with other metaheuristics or machine-learning models. Such developments have the potential to further bridge the gap between theoretical code design and computational discovery, expanding the range of MRD codes available for applications in communication, cryptography, and storage systems.

Acknowledgments: The authors would like to thank Professor Dr. João Pedro Mendonça Assunção Silva from the Department of Engineering at the University of Minho for his valuable supervision, insightful guidance, and continuous support throughout this research.

References

1. Delsarte, P. Bilinear forms over a finite field, with applications to coding theory. *Journal of Combinatorial Theory, Series A* **1978**, 25, 226–241. [https://doi.org/10.1016/0097-3165\(78\)90015-8](https://doi.org/10.1016/0097-3165(78)90015-8).
2. Gabidulin, E.M. Theory of codes with maximum rank distance. *Problems of Information Transmission* **1985**, 21, 1–12.

3. Gabidulin, E.M. *Rank Codes*; TUM.University Press: Technical University of Munich, Arcisstrasse 21, 80333 Munich, 2021. Translated by Vladimir Sidorenko. Layout design, typesetting, and cover design by Vladimir Sidorenko and Caroline Ennemoser., <https://doi.org/10.14459/2021md1601193>.
4. Kshevetskiy, A.; Gabidulin, E. The new construction of rank codes. In Proceedings of the Proceedings. International Symposium on Information Theory, 2005. ISIT 2005., 2005, pp. 2105–2108. <https://doi.org/10.1109/ISIT.2005.1523717>.
5. Sheekey, J. A new family of linear maximum rank distance codes. *Advances in Mathematics of Communications* **2016**, *10*, 475–488. <https://doi.org/10.3934/amc.2016019>.
6. Lunardon, G.; Trombetti, R.; Zhou, Y. Generalized twisted Gabidulin codes. *Journal of Combinatorial Theory, Series A* **2018**, *159*, 79–106. <https://doi.org/10.1016/j.jcta.2018.05.004>.
7. Csajbók, B.; Marino, G.; Polverino, O.; Zullo, F. Maximum scattered linear sets and MRD-codes. *Journal of Algebraic Combinatorics* **2017**, *46*, 193–213. <https://doi.org/10.1007/s10801-017-0762-6>.
8. Alfarano, G.N.; Neri, A.; Zullo, F. Maximum flag-rank distance codes. *Journal of Combinatorial Theory, Series A* **2024**, *207*, 105908. <https://doi.org/10.1016/j.jcta.2024.105908>.
9. Csajbók, B.; Marino, G.; Polverino, O.; Zhou, Y. MRD codes with maximum idealizers. *Discrete Mathematics* **2020**, *343*, 111985. <https://doi.org/10.1016/j.disc.2020.111985>.
10. Cuéllar, M.; Gómez-Torrecillas, J.; Lobillo, F.; Navarro, G. Genetic algorithms with permutation-based representation for computing the distance of linear codes. *Swarm and Evolutionary Computation* **2021**, *60*, 100797. <https://doi.org/10.1016/j.swevo.2020.100797>.
11. Azouaoui, A.; Belkasm, M. Efficient Dual Domain Decoding of Linear Block Codes Using Genetic Algorithms. *Journal of Electrical and Computer Engineering* **2012**. <https://doi.org/10.1155/2012/503834>.
12. Choukroun, Y.; Wolf, L. Learning linear block error correction codes. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning, JMLR.org, 2024, ICML'24. <https://dl.acm.org/doi/10.5555/3692070.3692422>.
13. Huang, L.; Zhang, H.; Li, R.; Ge, Y.; Wang, J. AI Coding: Learning to Construct Error Correction Codes. *IEEE Transactions on Communications* **2020**, *68*, 26–39. <https://doi.org/10.1109/TCOMM.2019.2951403>.
14. Gao, Y.; Liu, H.; Wang, X.; Zhang, K. On an artificial neural network for inverse scattering problems. *Journal of Computational Physics* **2022**, *448*, 110771. <https://doi.org/10.1016/j.jcp.2021.110771>.
15. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the Proceedings of ICNN'95-International Conference on Neural Networks. IEEE, 1995, Vol. 4, pp. 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>.
16. van Zyl, J.P.; Engelbrecht, A.P. Set-Based Particle Swarm Optimisation: A Review. *Mathematics* **2023**, *11*, 2980. <https://doi.org/10.3390/math11132980>.
17. Hamming, R.W. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal* **1950**, *29*, 147–160. <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>.
18. Sheekey, J. 13. MRD codes: constructions and connections. In *Combinatorics and Finite Fields*; Schmidt, K.U.; Winterhof, A., Eds.; De Gruyter: Berlin, Boston, 2019; pp. 255–286. <https://doi.org/10.1515/9783110642094-013>.
19. Bartoli, D.; Giulietti, M.; Marino, G.; Polverino, O. Maximum Scattered Linear Sets and Complete Caps in Galois Spaces. *Combinatorica* **2018**, *38*, 255–278. <https://doi.org/10.1007/s00493-016-3531-6>.
20. Bartoli, D.; Csajbók, B.; Montanucci, M. On a conjecture about maximum scattered subspaces of $\mathbb{F}_{q^6} \times \mathbb{F}_{q^6}$. *Linear Algebra and its Applications* **2021**, *631*, 111–135. <https://doi.org/10.1016/j.laa.2021.08.023>.
21. Napp, D.; Pinto, R.; Santana, F.; Vela, C. On the construction of MRD convolutional codes. *Linear and Multilinear Algebra* **2024**, *72*, 2653–2673. <https://doi.org/10.1080/03081087.2023.2300673>.
22. Shi, M.; Krotov, D.S.; Özbudak, F. Constructing MRD codes by switching. *Journal of Combinatorial Designs* **2024**, *32*, 219–237. <https://doi.org/10.1002/jcd.21931>.
23. Chen, H. New Explicit Good Linear Sum-Rank-Metric Codes. *IEEE Transactions on Information Theory* **2023**, *69*, 6303–6313. <https://doi.org/10.1109/TIT.2023.3292162>.
24. Martínez-Peñas, U. New constructions of MSRD codes. *Computational and Applied Mathematics* **2024**, *43*, 195. <https://doi.org/10.1007/s40314-024-02915-z>.
25. Borello, M.; Zullo, F. Geometric dual and sum-rank minimal codes. *Journal of Combinatorial Designs* **2024**, *32*, 238–273. <https://doi.org/10.1002/jcd.21934>.
26. Liu, X.; Zhang, J.; Wang, G. Constructions and List Decoding of Sum-Rank Metric Codes Based on Orthogonal Spaces over Finite Fields. arXiv preprint arXiv:2507.16377, 2025. 27 pages, 1 table. <https://arxiv.org/abs/2507.16377>.

27. Chen, J.; Xiong, J.; Wang, Y.; Xin, Q.; Zhou, H. Implementation of an AI-based MRD Evaluation and Prediction Model for Multiple Myeloma. *Frontiers in Computing and Intelligent Systems* **2024**, *6*, 127–131. <https://doi.org/10.54097/zJ4MnbWW>.
28. Clerc, M.; Kennedy, J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* **2002**, *6*, 58–73. <https://doi.org/10.1109/4235.985692>.
29. Trelea, I.C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters* **2003**, *85*, 317–325. [https://doi.org/10.1016/S0020-0190\(02\)00447-7](https://doi.org/10.1016/S0020-0190(02)00447-7).
30. Shi, Y.; Eberhart, R.C. Parameter selection in particle swarm optimization. In Proceedings of the Proceedings of the 7th International Conference on Evolutionary Programming VII. Springer, 1998, EP'98, pp. 591–600. <https://doi.org/10.1007/BFb0040810>.
31. Dehghani, B. *PSO-Guided Construction of MRD Codes in Rank Metric*, 2025. GitHub repository containing implementation and documentation. Available at https://github.com/behnamde/pso_mrd.
32. Hostetter, M. *Galois: A performant NumPy extension for Galois fields*, 2020. GitHub repository containing implementation and documentation. Available at <https://github.com/mhostetter/galois>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.