IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1077∼1086
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1077-1086

New family of error-correcting codes based on genetic algorithms

El Mehdi Bellfkih[1], Said Nouh[1], Imrane Chemseddine Idrissi[1], Khalid Louartiti[2], Jamal Mouline[1]
[1]Department of Mathematics and Computer Science, Faculty of Science Ben M'sick, University Hassan II, Casablanca, Morocco
[2]Department of Mathematical Sciences and Decision Support, ENSA, Abdelmalek Essaâdi University, Tetouan, Morocco

Article Info

Article history:

Received Nov 28, 2023
Revised Nov 17, 2024
Accepted Nov 24, 2024

Keywords:

Decoding
Design
Error-correcting codes
Generator vector
Genetic algorithm
Minimum distance

ABSTRACT

This paper introduces a novel error-correcting code (ECC) construction and decoding approach utilizing genetic algorithms (GAs). Classical ECCs often struggle with efficiency in correcting multiple errors due to time-consuming matrix-based encoding and decoding processes. Our GA-based method optimizes generator vectors to maximize the minimum distance between codewords, enhancing error correction capabilities. Specifically, we construct a new family of ECCs with code length 31, dimension 12, and minimum distance 7, reducing complexity from $O(kn)$ to $O(k(n-k))$ by encoding message blocks with vectors instead of matrices. In the decoding phase, the GA effectively corrects errors in received codewords. Experimental results show that at a signal-to-noise ratio (SNR) of 7.7 dB, our method achieves a bit error rate (BER) of 10^{-5} after only 9 generations of the GA. These results demonstrate improved error correction and decoding performance compared to traditional methods. This study contributes an innovative approach using GAs for error correction, offering simpler encoding and robust performance in coding schemes.

Corresponding Author:

El Mehdi Bellfkih
Department of Mathematics and Computer Science, Faculty of Science Ben M'sick, University Hassan II
Casablanca, Morocco
Email: elmehdi.bellfkih@gmail.com

1. INTRODUCTION

The transmission and storage of information are susceptible to corruption due to various physical or logical faults, which can result in system-wide failures. To mitigate such risks, robust testing and fault tolerance mechanisms are essential for ensuring secure and stable communication flows. Error-correcting codes (ECCs) play a pivotal role in safeguarding data integrity and reliability by incorporating redundant information into transmitted messages.

The efficacy of ECCs lies in their ability to detect and/or correct errors that may arise during data transmission or storage. This error-correction capability is crucial for maintaining data integrity under adverse conditions. While linear block codes, such as Hamming codes, offer decent error-correction capability, they are inherently limited in their scope. In contrast, nonlinear block codes, exemplified by turbo codes, exhibit superior error-correction capabilities but are accompanied by higher decoding complexities [1], [2]. As shown in the Figure 1, the minimum distance of a code is directly related to its error detection and correction capability. A code with a larger minimum distance can detect and correct more errors compared to a code with a smaller minimum distance.

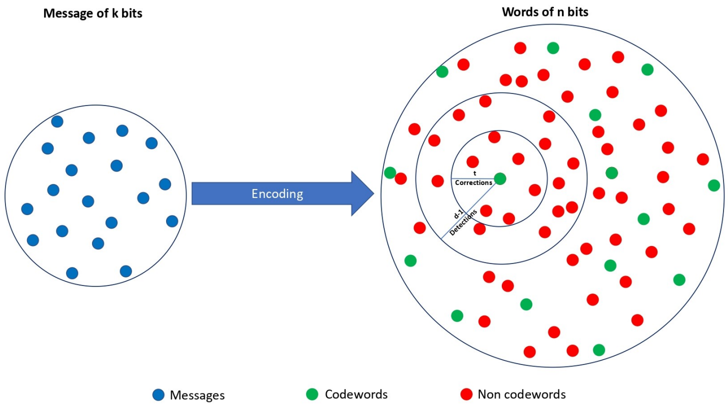

Figure 1. Correlating minimum Hamming distance with error detection and correction capabilities

In the realm of ECC construction, linear block codes, grounded in linear algebra, are renowned for their simplicity of implementation, analysis, and comprehension. They excel at detecting and correcting errors within a confined bit range. Examples include Hamming codes, Reed-Solomon codes, and Bose-Chaudhuri-Hocquenghem (BCH) codes. Conversely, nonlinear block codes present a more intricate landscape, demanding deeper analytical understanding and implementation efforts. Yet, they boast broader error-correcting capabilities, effectively managing errors across a larger bit spectrum. Notable examples encompass Reed-Muller codes, Golay codes, and BCH codes.

However, despite the advancements in ECC design, the process of decoding remains a challenging task. Traditional decoding methods often encounter computational bottlenecks, particularly when dealing with complex codes. Herein lies the potential for employing metaheuristic approaches to decode ECCs efficiently. Metaheuristic algorithms, renowned for their adaptability and problem-solving prowess, offer a promising avenue for tackling the intricacies of ECC decoding. By leveraging metaheuristic techniques, such as genetic algorithms (GAs), simulated annealing, or particle swarm optimization, researchers can explore novel decoding strategies capable of surmounting the complexities associated with ECCs.

There is various classes of codes in coding theory, and various method to construct them aiming to achieve the best results e.g., the reliable communication, better complexity, easy construction of code. Let $\mathbb{F}_2$ be a field of order 2 and $\mathbb{F}_2^k$ be a vector space of length k. Here we present our new k-dimensional binary linear code $\mathcal{C}$ over $\mathbb{F}_2^n$ whose G is its generator matrix, or $g(x)$ is its polynomial generator (the rows of G form a basis for $\mathcal{C}$). $[n, k, d]_2$ denotes a 2-ary linear code with length n, dimenion k and minimum distance d. An element of $\mathcal{C}$ is called a codeword, its weight is the number of nonzero coordinate. The minimum distance of $\mathcal{C}$ is the smallest Hamming distance between distinct codewords (is also the smallest weight in case of binary linear codes) denoted by d($\mathcal{C}$). The Singleton bound as in (1) states that a (n, k, d)-code or $[n, k, d]$-code satisfy.

$$d(\mathcal{C}) \leq n - k + 1 \qquad (1)$$

A code with linearity condition and achieves the equality in the Singleton bound is called maximum distance separable (MDS) code. To achieve the goal of finding high-performing ECC, our approach takes advantage of the optimization nature of the problem. By formulating the problem as an optimization problem, we can leverage the power of optimization algorithms, such as GAs [3], [4], to search for the best possible solutions. The GA framework allows us to efficiently explore the vast solution space and find good ECC with high minimum distances, making it an ideal approach for this type of problem. GAs are a type of optimization algorithm that is inspired by the process of natural selection and evolution [5]. GAs are used to solve complex problems by simulating the process of evolution, where a population of potential solutions evolves over time towards an optimal solution [6]–[9]. They works by representing a problem as a set of candidate solutions, also known as a population. Each candidate solution is encoded as a string of parameters, called a chromosome. The chromosomes in the population are then evaluated using a fitness function that assigns a numerical score to each chromosome based on how well it solves the problem. The chromosomes with the highest fitness scores are selected as parents and used to produce offspring, which are new candidate solutions, through a process called crossover. In this process, the genetic information from the parent chromosomes is combined to produce a new chromosome (child). This process is repeated over several generations, leading to the evolution of the population towards better solutions. GAs also include a mechanism for introducing random variations into the population, called mutation. This allows the algorithm to explore new regions of the solution space and helps to prevent getting stuck in local optima. They are well-suited for problems that have multiple solutions or where the solution space is complex and difficult to explore using traditional optimization methods. In the next section, we will consider related research findings to contextualize and augment our study's conclusions.

2. RELATED WORKS

The field of ECCs plays a crucial role in ensuring data integrity and reliability in various applications. Despite the widespread use of existing ECCs, there are challenges that hinder their efficiency, particularly in terms of the time-consuming encoding and decoding processes. To address these limitations, researchers have turned to innovative approaches such as GA for code design and decoding. GA offer a promising avenue for generating codes with high minimum distances, thus enhancing error detection and correction capabilities. A range of studies have explored the design and decoding of ECC [10]–[12]. Natarajan *et al.* [13] developed algebraic ECC for informed receivers, while Elkelesh *et al.* [11] proposed a GA-based low density parity check (LDPC) code design scheme. Das and Touba [14] introduced a new class of single burst ECC with parallel decoding, and Zhang *et al.* [15] presented a decoding algorithm for five-error-correcting binary quadratic residue codes. These studies collectively contribute to the advancement of ECC, with a focus on informed receivers, LDPC codes, burst error correction, and decoding algorithms.

McGuire and Sabin [16] have employed GA to search for linear binary codes with optimal minimum distance. In another paper, Maini *et al.* [17] developed suboptimal soft decision decoders for linear block codes. GA have also been utilized to tackle the problem of finding ECCs that correct a maximum number of errors [18]. These studies highlight the effectiveness of GA in addressing various aspects of error correction code design and decoding, and recognized as one of the most powerful optimization methods due to its versatility and ease of implementation across various problem domains. One of the key strengths of GA lies in their diverse set of operators and options, which allow for flexible exploration and exploitation of the search space [19]–[21]. These operators, including selection, crossover, and mutation, provide a rich toolbox that can be tailored to specific optimization problems. Moreover, the GA's inherent parallelism and population-based nature enable it to effectively handle complex and multimodal optimization landscapes. In fact, it can be viewed as a variant of the minimum distance problem, which is known to be NP-hard. The minimum distance of a code plays a crucial role in its error detection and correction capabilities. However, determining the exact minimum distance of a code is computationally complex and requires exhaustive search over all codewords. This computational hardness motivates the exploration of heuristic approaches, such as GA, to efficiently search for codes with large minimum distances. The design of ECCs has traditionally relied on coding-theoretic principles, aiming to optimize code properties such as minimum Hamming distance and decoding threshold. However, recent advancements have explored the application of artificial intelligence (AI) techniques, particularly GA, for ECC design. Huang *et al.* [10] investigate an AI-driven approach using GA to design optimal codes within specific families, showcasing comparable performance to existing codes and even superior performance in certain cases. Amirzadeh *et al.* [22] focus on joint GA and linear programming optimization for LDPC codes, striving for low complexity, high coding threshold, and decoding stability. Mahran [23] explores the optimization of turbo product codes (TPC) parameters using GA, finding a balance between error performance and code complexity. Joundan *et al.* [24] present a GA approach for designing linear codes with large minimum weight and small dual minimum distance, demonstrating effective error correction performance. These studies collectively highlight the potential of GA in ECC design, offering opportunities for improving code performance, complexity, and error correction capabilities in various communication systems.

GAs have emerged as a powerful tool for ECCs decoding. Chaibi *et al.* [25] present a GA-based decoder for LDPC codes, demonstrating its superior performance compared to the sum-product decoder. Azouaoui *et al.* [26] propose hard-decision and soft-decision decoding techniques based on GAs for general ECC, showcasing their effectiveness over various transmission channels. Broulím *et al.* [27] explore the application of GA optimization algorithms to design parity-check matrices for LDPC codes, enabling the correction of burst errors. Nouh *et al.* [28] focus on decoding block codes using GAs and permutations set, showing comparable error correcting performances to existing methods. Elkelesh *et al.* [11] present a decoder-tailored polar code design using GAs, achieving the same error-rate performance as existing decoding algorithms while reducing the decoding complexity. Berkani *et al.* [29] propose compact GAs with larger tournament size for improved decoding of linear block codes, demonstrating the effectiveness of larger tournament sizes in soft decision decoding. These studies collectively highlight the potential of GAs in ECC decoding and code design, offering enhanced error correction performance, reduced complexity, and improved decoding capabilities in various communication.

3. PROPOSED METHODS

In this section, our GA-based methods are proposed using the principal factors (fitness function, crossover, and mutation factors). we will delve into the application of GAs based methods in the encoding and decoding phases of ECCs. Specifically, we will explore how GAs can be utilized to optimize these crucial stages of the coding process. For the encoding phase, we will discuss the use of GAs based methods to determine optimal generator vectors, considering factors such as code properties and encoding complexity. In the decoding phase, we will examine how GA based method can aid in finding the corrected corrupted received words, focusing on factors such as decoding performance, and error correction capability. Through a detailed analysis, we aim to shed light on the main factors and considerations when employing GA for efficient encoding and decoding of ECCs.

3.1. Construction phase

Our primary objective is to identify a generator vector that maximizes the distance between encoded messages. By employing GA in the encoding phase, we aim to find the most suitable generator vector that enhances error correction capabilities. However, we will rely on encoding through multiplying by generator vector and conversion based on binary and decimal.

The Figure 2 showcases the sequential steps involved in encoding a message using a generator vector. The process begins with the division of the message into blocks, represented in decimal form. These blocks are then converted into their corresponding binary forms, ensuring that the message is represented using binary digits. The next stage focuses on the encoding process itself. The binary message, consisting of k bits, undergoes multiplication with a generator vector of n-k bits. This multiplication results in a binary message of length n bits, which represents the encoded message with added redundancy for error correction or detection. Finally, the encoded message is converted back to its original decimal form. This figure provides a clear visualization of the encoding process, emphasizing the transformation from decimal to binary representation, the application of the generator vector for encoding, and the subsequent conversion back to decimal form.

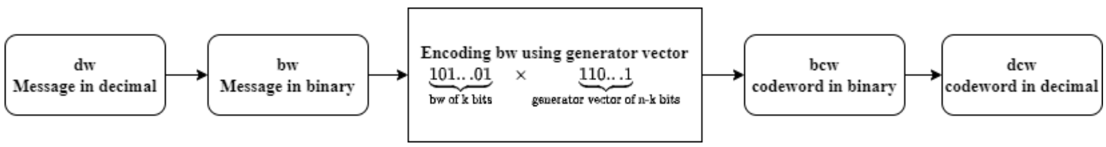

Figure 2. Encoding process using generator vector

The diagram in Figure 3 illustrates a GA-based method for finding an optimal generator vector. The GA operates on a population of candidate generator vectors, with the number of generations and initial population size specified as input parameters. Elitism is employed as the selection strategy, ensuring that the fittest individuals are preserved in each generation. The fitness function, defined as the minimum distance achieved by a generator vector, guides the evaluation and selection process. Crossover and mutation operators are applied to introduce diversity and explore new solutions within the population. The initial population consists of generator vectors with $n - k$ bits, where n is the total number of code-word bits and k is the number of message bits. The GA iteratively evolves the population to converge towards an optimal generator vector that maximizes the minimum distance.

$$f_{gen} = \min\{d(\mathcal{C}) : \mathcal{C} = \{bw_i \times gen, \forall i < 2^k\}\} \tag{2}$$

Where bw_i are messages of k bits and gen is a generator vector of n-k bits.

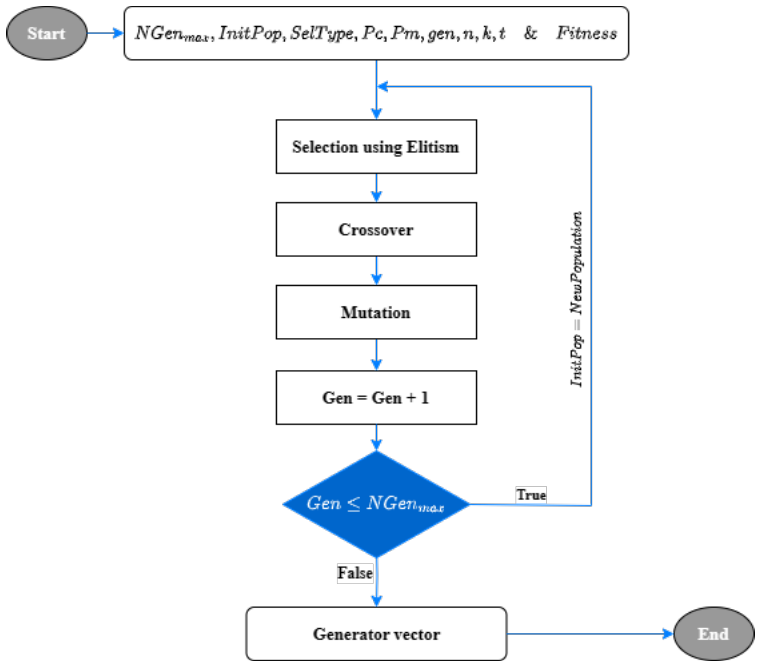

Figure 3. Diagram of the method based on the GA to find the generator vector for an ECC

The provided Figure 4 demonstrates the crossover operation in our GA-based method. If a randomly generated probability is less than or equal to a predefined value ($p_m = 0.97$), the crossover is applied. Two parent individuals, each represented by a binary sequence of n bits, are selected based on fitness function value as mentioned in (2). A random position, denoted as p, is chosen within the length of the sequence. The first child is created by combining the section from the first parent starting from position 0 up to position p, with the section from the second parent starting from position p up to position n. Similarly, the second child is formed by combining the section from the second parent from position 0 to p, and from the first parent from position p to n. Additionally, the figure indicates that the mutation operation follows a similar principle. If a randomly generated number between 0 and 1 is less than or equal to a predefined value ($p_c = 0.02$), the mutation occurs. It involves flipping the value at a specific position in the child's binary sequence.

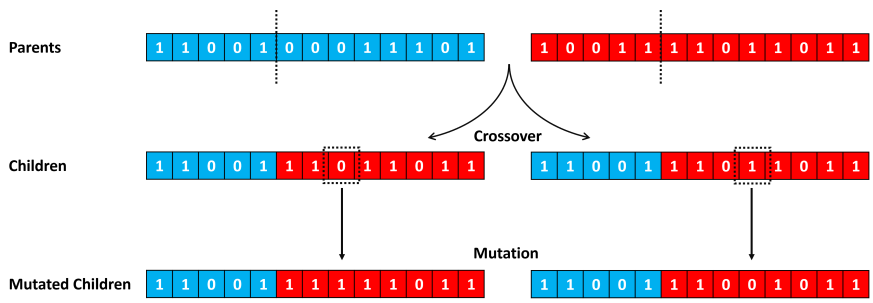

Figure 4. Crossover and mutation factors

3.2. Decoding phase

We present a GA-based method for correcting corrupted received code-words in ECCs. Our objective is to accurately recover the original information from the received word, even in the presence of errors. The proposed method utilizes GA to iteratively search for the optimal solution that converge to the correct code-word.

$$f_{codeword} = d(receivedword, codeword) \quad (3)$$

The diagram in Figure 5 illustrates a GA-based method for decoding received words in ECCs. The algorithm takes several inputs, including the length n and dimension k of the ECC, the number of corrections allowed t, the number of generations for the algorithm to iterate, and the initial population consisting of code-words generated using the available generator vectors. Elitism is employed as the selection strategy, and the fitness function is defined as the minimum distance between the received word and the code-words as in (3) in the population. The crossover and mutation operations are applied with specific rates and with the same strategy as shown in the Figure 4, aiming to explore and exploit the solution space. The initial population is initialized with generator vectors of size n-k bits. Through the iterations of the GA, the method aims to decode the received word and recover the original information accurately.

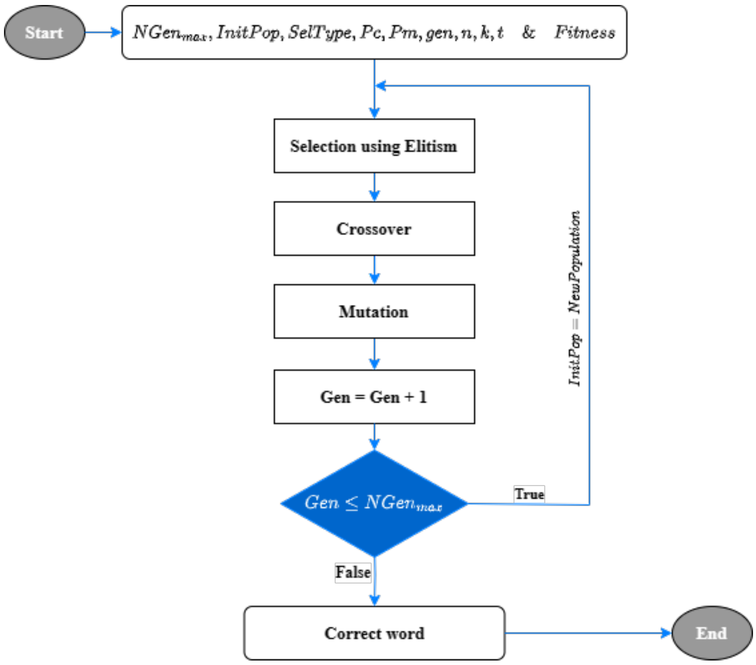

Figure 5. Diagram of the method based on the GA for decoding new ECCs

4. RESULTS AND DISCUSSION

In this section, we present the results obtained from our study on the construction and decoding of ECCs. The subsections below detail the outcomes of our investigations into both the construction and decoding phases, highlighting the performance and efficacy of our proposed methodologies. Through rigorous experimentation and analysis, we assess the effectiveness of our approach in achieving robust error correction capabilities and efficient decoding processes.

4.1. Construction of error-correcting codes

The provided Table 1 outlines the default parameters used in running the GA-based method for finding the generator vector of the ECC with a length of 31 and a dimension of 12. These parameters, which include settings such as population size, crossover rate, and mutation probability, serve as the initial configurations for the GA, providing a starting point for the optimization process. By carefully selecting these default parameters, the algorithm efficiently navigates the search space to identify generator vectors that maximize the minimum distance between codewords, thereby enhancing the ECC's error-correcting capabilities.

After running the GA-based method with the default parameters mentioned in Table 1, we obtained a set of generator vectors for an ECC of length 31 and dimension 26. The Table 2 results include the minimum distance achieved by these generator vectors, which is equal to the known lower bound. This suggests that the GA effectively identified generator vectors that offer optimal error correction capabilities for the given ECC dimensions.

Table 1. The default parameters for GA based method for codes of moderate lengths

Parameter	value
Initial population size	20000
Selection	elitism
Crossover rate	0.93
Mutation rate	0.02
Number of generations	50

Table 2. Set of ECCs of parameters (31,12)

n	k	d	Generator
31	12	7	1110001001011000001
31	12	7	1110000111101100101
31	12	7	1101001101011000101
31	12	7	1010000110111010111
31	12	7	1110010110100100101
31	12	7	1011100111111000111
31	12	7	1111011100111010011
31	12	7	1111010010101000111
31	12	7	1111100110001111011
31	12	7	1111111010010011001
31	12	7	1111001011000011001
31	12	7	1110101001111100111
31	12	7	1111110111011010011
31	12	7	1110100010101001111

The application of the GA-based approach resulted in the discovery of ECCs with dimension 12 and length 31, showcasing minimum distances that equal to the known lower bound. This significant achievement holds promising implications for error detection and correction in practical scenarios. These codes exhibit an exceptional capability to detect and correct errors, surpassing the performance of previously known codes. The listed codes in Table 2 exemplify superior error-correcting properties, indicating their potential for enhancing data integrity and ensuring reliable information transmission and storage. Also, Our GA based method has successfully identified optimal generator vectors, enabling a more efficient encoding process. Instead of multiplying message blocks of length k by a matrix of dimension (k,n), we now multiply them by a vector of length n-k. This reduction in dimensionality results in significant complexity gains, leading to improved efficiency in the encoding process. The results are summarized in the Table 3.

Table 3. Encoding complexity

Encoding process	Complexity
Encoding via generator matrix	$\mathcal{O}(kn)$
Encoding via generator vector	$\mathcal{O}(k(n-k))$

4.2. Decoding

After successfully finding a set of generator vectors that maximize the error-correcting capabilities of our ECCs, we proceed to the decoding phase, where we introduce a GA-based method for decoding these new codes. This method leverages GA to efficiently correct errors in the received codewords by exploring possible solutions and selecting the most optimal one based on a fitness function. The focus of this section is on evaluating the bit error rate (BER) performance of the decoding process, demonstrating how effectively our GA-based decoder restores the original messages under various levels of noise.

The Table 4, presents the chosen default parameters for the GA-based decoding method include a relatively small population size and a limited number of generations. This decision was made to ensure a manageable computational complexity during the decoding process. Our algorithm is designed to create a population of candidate words derived from a received word. Specifically, the algorithm generates a set of $InitPop$ words closely related to the input received word. Additionally, we implement an adjustment by increasing the minimum allowable distance between generated words. These two strategic steps collectively serve to reduce algorithmic complexity and enhance computational efficiency in terms of speed. Furthermore, in instances where correction of the received word is not feasible due to an error count surpassing the predefined threshold value (t), the algorithm provides a set of proximate words. This information proves valuable in scenarios where understanding the proximity of the received data is of significance.

Table 4. The default parameters for GA based method for decoding ECCs

Parameter	Value
Initial population size	500
Selection	elitism
Crossover rate	0.93
Mutation rate	0.07
Number of generations	1000

The Figure 6 illustrates the exceptional decoding performances of our method for our found code with a length of 31, dimension 12, and a minimum distance of 7. Notably, at an signal-to-noise ratio (SNR) of 7.7 dB, the BER stands at 10^{-5}, highlighting the decoder's initial performance. As the SNR increases to 8.5 dB, the BER decreases to 10^{-6}, underscoring the decoder's enhanced error-correcting capabilities with improved SNR. This progression signifies the decoder's effectiveness in achieving higher levels of data accuracy under varying signal conditions.

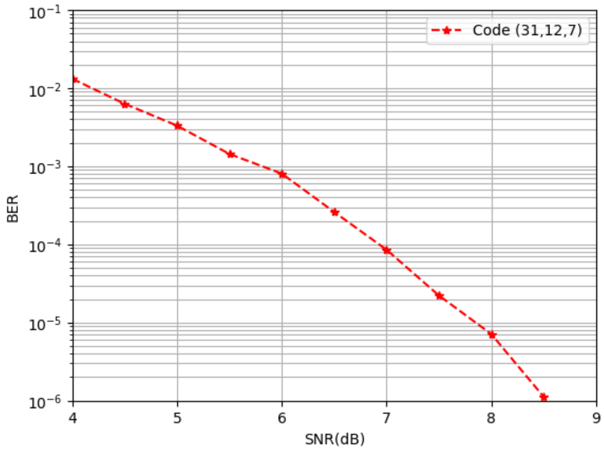

Figure 6. BER performance of GA-based decoder

In spite of the substantial increase in the number of generations as indicated in Table 4, intended to ensure the successful decoding of received words irrespective of the number of errors, the achieved outcomes remain below the values specified in Table 4. This observation is substantiated by the statistical summary presented in Table 5, which provides insights into the average and standard deviation. Notably, the low values of both parameters in Table 5 signify the commendable efficiency and effectiveness of our algorithm in the decoding process across varying SNRs.

Table 5. Statistical summary of algorithm performance

Number of words	Max number of generations	Avg number of generations	Std number of generations
100000	1000	≈ 8.7	≈ 8.2

5. CONCLUSION

This research article has demonstrated the effectiveness of utilizing GA-based methods for both the construction and decoding of ECCs. By employing these methods, we have successfully identified generator vectors with high minimum Hamming distances, thereby streamlining the encoding process and enhancing the BER performance of the codes. However, we acknowledge the limitation of achieving relatively low rates. Moving forward, our future objectives entail addressing this limitation by optimizing the generation of generator vectors for specified parameters of length n and dimension k, as well as refining our decoder to further improve the BER performances of the ECCs. Through these endeavors, we aim to bolster the efficiency and efficacy of ECCs in real-world communication and storage systems.

REFERENCES

[1] A. Said, "Introduction to arithmetic coding – theory and practice," *arXiv-Computer Science*, 2023.

[2] J. H. V. Lint, "Introduction to coding theory," *Discrete Applied Mathematics*, Berlin, New York: Springer, vol. 6, no. 1, 1983, doi: 10.1016/0166-218X(83)90114-2.

[3] S. Benghazouani, S. Nouh, and A. Zakrani, "Enhancing breast cancer diagnosis: a comparative analysis of feature selection techniques," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 4, pp. 4312–4322, 2024, doi: 10.11591/ijai.v13.i4.pp4312-4322.

[4] K. Kangra and J. Singh, "A genetic algorithm-based feature selection approach for diabetes prediction," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 2, pp. 1489–1498, 2024, doi: 10.11591/ijai.v13.i2.pp1489-1498.

[5] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, Massachusetts: The MIT Press, 1992, doi: 10.7551/mitpress/1090.001.0001.

[6] M. A. Albadr, S. Tiun, M. Ayob, and F. AL-Dhief, "Genetic algorithm based on natural selection theory for optimization problems," *Symmetry*, vol. 12, no. 11, 2020, doi: 10.3390/sym12111758.

[7] T. Harada and E. Alba, "Parallel genetic algorithms: A useful survey," *ACM Computing Surveys*, vol. 53, no. 4, 2020, doi: 10.1145/3400031.

[8] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-a literature review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 380–384, doi: 10.1109/COMITCon.2019.8862255.

[9] O. Castillo and L. T. Aguilar, "Genetic algorithms," in *Type-2 Fuzzy Logic in Control of Nonsmooth Systems*, 2019, pp. 23–39, doi: 10.1007/978-3-030-03134-3_2.

[10] L. Huang, H. Zhang, R. Li, Y. Ge, and J. Wang, "AI coding: learning to construct error correction codes," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 26–39, 2020, doi: 10.1109/TCOMM.2019.2951403.

[11] A. Elkelesh, M. Ebada, S. Cammerer, and S. T. Brink, "Decoder-tailored polar code design using the genetic algorithm," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4521–4534, 2019, doi: 10.1109/TCOMM.2019.2908870.

[12] A. Elkelesh, M. Ebada, S. Cammerer, L. Schmalen, and S. T. Brink, "Decoder-in-the-loop: genetic optimization-based LDPC code design," *IEEE Access*, vol. 7, pp. 141161–141170, 2019, doi: 10.1109/ACCESS.2019.2942999.

[13] L. Natarajan, Y. Hong, and E. Viterbo, "New error correcting codes for informed receivers," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 2839–2843, doi: 10.1109/ISIT.2016.7541817.

[14] A. Das and N. A. Touba, "A new class of single burst error correcting codes with parallel decoding," *IEEE Transactions on Computers*, vol. 69, no. 2, pp. 253–259, 2020, doi: 10.1109/TC.2019.2947425.

[15] Y. Zhang, X. Bao, Z. Yuan, and X. Wu, "Decoding of the five-error-correcting binary quadratic residue codes," *American Journal of Mathematical and Computer Modelling*, vol. 2, no. 1, pp. 6–12, 2017.

[16] K. M. McGuire and R. E. Sabin, "Using a genetic algorithm to find good linear error-correcting codes," in *Proceedings of the 1998 ACM symposium on Applied Computing - SAC '98*, 1998, pp. 332–337, doi: 10.1145/330560.330834.

[17] H. Maini, K. Mehrotra, C. Mohan, and S. Ranka, "Genetic algorithms for soft-decision decoding of linear block codes," *Evolutionary Computation*, vol. 2, no. 2, pp. 145–164, 1994, doi: 10.1162/evco.1994.2.2.145.

[18] M. D. J. Simon, J. A. G. Pulido, M. A. V. Rodriguez, J. M. S. Perez, and J. M. G. Criado, "A genetic algorithm to design error correcting codes," in *2006 IEEE Mediterranean Electrotechnical Conference*, 2006, pp. 807–810, doi: 10.1109/MELCON.2006.1653221.

[19] B. El Mehdi, S. Nouh, I. C. Idrissi, A. Ettaoufik, K. Louartiti, and J. Mouline, "On the computation of the automorphisms group of low density parity check codes using genetic algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 1059–1066, 2022, doi: 10.11591/ijeecs.v25.i2.pp1059-1066.

[20] J. Alander, "Population size, building blocks, fitness landscape and genetic algorithm search efficiency in combinatorial optimization," in *Practical Handbook of Genetic Algorithms*, CRC Press, 1998, pp. 459–486, doi: 10.1201/9781420050080.ch13.

[21] M. Tabassum, "A genetic algorithm analysis towards optimization solutions," *International Journal of Digital Information and Wireless Communications*, vol. 4, no. 1, pp. 124–142, 2014, doi: 10.17781/p001091.

[22] A. Amirzadeh, M. H. Taieb, and J.-Y. Chouinard, "On the design of good LDPC codes with joint genetic algorithm and linear programming optimization," in *2017 15th Canadian Workshop on Information Theory (CWIT)*, 2017, pp. 1–5, doi: 10.1109/CWIT.2017.7994822.

[23] A. Mahran, "Optimizing the parameters of turbo product codes using genetic algorithms," in *2017 IEEE Aerospace Conference*, 2017, pp. 1–7, doi: 10.1109/AERO.2017.7943565.

[24] I. A. Joundan, S. Nouh, and A. Namir, "Design of good linear codes for a decoder based on majority voting procedure," in *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, 2016, pp. 1–6, doi: 10.1109/ACOSIS.2016.7843918.

[25] H. Chaibi, A. Berkani, and M. Ahmad, "Syndrome weight decision based genetic algorithm decoder for LDPC codes," *International Journal of Computer Applications*, vol. 127, no. 6, pp. 38–43, 2015, doi: 10.5120/ijca2015906403.

[26] A. Azouaoui, I. Chana, and M. Belkasmi, "Efficient information set decoding based on genetic algorithms," *International Journal of Communications, Network and System Sciences*, vol. 5, no. 7, pp. 423–429, 2012, doi: 10.4236/ijcns.2012.57052.

[27] J. Broulím, A. Ayriyan, and H. Grigorian, "Genetic optimization of LDPC codes to improve the correction of burst errors," *EPJ Web of Conferences*, vol. 226, 2020, doi: 10.1051/epjconf/202022602006.

[28] S. Nouh, I. Chana, and M. Belkasmi, "Decoding of block codes by using genetic algorithms and permutations set," *International Journal of Communication Networks and Information Security*, vol. 5, no. 3, pp. 201–209, 2013, doi: 10.17762/ijcnis.v5i3.428.

[29] A. Berkani, A. Azouaoui, M. Belkasmi, and B. Aylaj, "Improved decoding of linear block codes using compact genetic algorithms with larger tournament size," *International Journal of Computer Science Issues*, vol. 14, no. 1, pp. 15–24, 2017, doi: 10.20943/01201701.1524.

BIOGRAPHIES OF AUTHORS

El Mehdi Bellfkih holds a Ph.D. in applied mathematics from Hassan II University, specializing in coding theory, error-correcting codes, artificial intelligence, and machine learning. His research explores innovative solutions in these fields to address complex computational problems. He can be contacted at email: elmehdi.bellfkih@gmail.com.

Said Nouh holds a Ph.D. in computer sciences at National School of Computer Science and Systems Analysis (ENSIAS), Rabat, Morocco in 2014. He is currently professor (higher degree research (HDR)) at Faculty of sciences Ben M'Sick, Hassan II University, Casablanca, Morocco. His current research interests artificial intelligence, machine learning, deep learning, telecommunications, information, and coding theory. He can be contacted at email: said.nouh@univh2m.ma.

Imrane Chemseddine Idrissi is a Ph.D. in computer science at Faculty of Sciences Ben M'Sik (FSBM), Hassan II University, Casablanca, Morocco. He received a master's thesis in data science and big data at ENSIAS Mohammed V university in 2019. His current research interests include networks and systems, telecommunications, information, coding theory, machine learning, and deep learning. He can be contacted at email: imrane.chemseddine-etu@etu.univh2c.ma or imran.chems@gmail.com.

Khalid Louartiti originally hailing from Taounate, Morocco, he earned his Ph.D. from Sidi Mohamed Ben Abdellah University in Fes, Morocco. Presently serving as a Professor at the National School of Applied Sciences (ENSA) in Tetouan, Morocco. His research focuses on graph theory, modules, ideals, commutative algebra, and amalgamated algebra. He can be contacted at email: lokha2000@hotmail.com.

Jamal Mouline originally from Ouazzane, Morocco, he earned his Ph.D. from Provence University in France. Presently, he holds the position of a Professor in the Department of Mathematics and Informatics at Hassan II University in Morocco. His research focuses on fixed point theory and combinatorial theory. He can be contacted at email: mouline61@gmail.com.