mathematics

MDPI

Article

DNA Code Design Based on the Cosets of Codes Over $\mathbb{Z}_4$

Adel N. Alahmadi [1,*], Fatimah Anas Melibari [1] and Manish K. Gupta [2,*]

[1] Research Group of Algebraic Structures and Applications, Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia; fmohammedmelebari@stu.kau.edu.sa
[2] Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar 382007, India
* Correspondence: analahmadi@kau.edu.sa (A.N.A.); mankg@guptalab.org (M.K.G.)

Abstract: DNA code design is a challenging problem, and it has received great attention in the literature due to its applications in DNA data storage, DNA origami, and DNA computing. The primary focus of this paper is in constructing new DNA codes using the cosets of linear codes over the ring $\mathbb{Z}_4$. The Hamming distance constraint, GC-content constraint, and homopolymers constraint are all considered. In this study, we consider the cosets of Simplex alpha code, Kerdock code, Preparata code, and Hadamard code. New DNA codes of lengths four, eight, sixteen, and thirty-two are constructed using a combination of an algebraic coding approach and a variable neighborhood search approach. In addition, good lower bounds for DNA codes that satisfy important constraints have been successfully established using Magma software V2.24-4 and Python 3.10 programming in our comprehensive methodology.

Keywords: DNA codes; DNA word design; cosets of codes; GC-content constraint; homopolymers constraint

MSC: 81P45; 05C50; 11E16

Citation: Alahmadi, A.N.; Melibari, F.A.; Gupta, M.K. DNA Code Design Based on the Cosets of Codes Over $\mathbb{Z}_4$. *Mathematics* **2023**, *11*, 4732. https://doi.org/10.3390/math11234732

Academic Editor: Jiyou Li

Received: 7 September 2023
Revised: 21 October 2023
Accepted: 23 October 2023
Published: 22 November 2023

1. Introduction

In 1994, in a seminal work, Adleman [1] solved an instance of the Hamilton path problem using a bunch of DNA strands, giving birth to a new branch of DNA computing. This further resulted in many other new branches, such as DNA-based data storage, DNA origami, and chemical computing. The backbone of DNA computing is DNA hybridization, which is also the source of errors. To mitigate the errors in DNA computing, often, we require sufficiently dissimilar DNA strands (called DNA codes) that satisfy certain constraints, such as the Hamming distance constraint, the GC-content constraint, and the homopolymers constraint. To avoid undesirable hybridization, the Hamming distance constraint is applied to measure the difference between two DNA strings [2,3]. When we store data in DNA, different types of errors occur. The DNA data storage is a two-step process, viz., DNA synthesis (writing) and DNA sequencing (reading). The most frequent errors that occur during these processes of DNA storage are deletion (certain symbols are deleted), insertion (certain symbols are inserted), and substitution (certain symbols are flipped between A, C, G, and T). The deletion errors occur more often if the encoded DNA strand has a repetition of symbols (such as if A is repeated in $CGAAAAAATCG$), called homopolymers. If the Hamming distance between the DNA stands is high, it forces fewer substitution errors. If the encoded DNA strands have a higher GC-content, that will affect sequencing errors. The ideal range for the GC-content is $\frac{n}{2}$. Similarly, the presence of homopolymers increases the error rate. Thus, it is very important to construct a large set of DNA codes that satisfy these constraints. Many bounds on the DNA codes have been studied with respect to some of these constraints [4]. For more details about these bounds and for a 16-year overview of DNA coding theory, the reader is referred to [4].

Recently, conflict-free DNA codes have been constructed in [5]. This was further extended by S. T. Dougherty and his team in [6], where they provided better results using MAGMA and MATLAB. In this work, we are focusing on a special case of conflict-free DNA codes, that is, a homopolymer-free construction of degree one, using our new algorithm. Note that 1-conflict-free DNA codes are known as DNA codes free from homopolymers. The homopolymers constraint is the most essential constraint for DNA codes since it prevents errors from occurring in the DNA synthesis. Several attempts have been made to improve new lower bounds using various DNA code construction methodologies, such as lexicographic approaches and stochastic local search [7]. Additionally, the variable neighborhood search approach), the simulated annealing approach, and the evolutionary algorithm approach are three metaheuristics used in [8].

Niema Aboluion used cosets of a linear code to improve the lower bounds on DNA codes in her Ph.D. thesis [9] because it was discovered that cosets of a linear code sometimes create more code words with constant GC-content than the linear code itself. In this paper, we propose a new method for constructing the DNA codes. The current work focuses on the coset formations of linear codes over $\mathbb{Z}_4$ using several DNA code construction methodologies. We have used Magma software V2.24-4 for generating the code words over $\mathbb{Z}_4$, and then Python programming is used to apply the constraints to generate the final results.

2. Problem Description

The problem considered in this paper is finding the largest set of DNA codes (a DNA code $\ell_{DNA}(n, M, d) \subset \sum_{DNA}^{n} = \{A, T, C, G\}^n$, with each DNA code word of length n, size M, and minimum Hamming distance d) that satisfies the following combinatorial constraints:

- **Hamming Distance Constraint (HD):** For some Hamming distance d, $d_H(x_{DNA}, y_{DNA}) \geq d \quad \forall x_{DNA}, y_{DNA} \in \ell_{DNA}$, d_H denotes the Hamming distance between any two code words.
- **GC-Content Constraint:** Each code word $x_{DNA} \in \ell_{DNA}$ has the same GC content. GC-content is denoted by w, and it is the number of positions in which the word has coordinate C or G. Generally, $w = [\frac{n}{2}]$.
- **Homopolymers Constraint:** For a DNA code $\ell_{DNA}(n, M, d)$, the homopolymers constraint means that no two consecutive elements in a DNA code word are identical. For example, $ACCT$ is not considered in the set of DNA code words since it has a repeated C.

Following [10], $A_4^{GC}(n, d, w)$ denotes the maximum size of a DNA code of length n with constant GC-content $w = [\frac{n}{2}]$ that satisfies the HD constraint for a given d and the homopolymers constraint. The purpose of this paper is to improve the lower bounds for $A_4^{GC}(n, d, w)$.

2.1. Codes over $\mathbb{Z}_4$

Definition 1. *Following [11], a* ***code over*** $\mathbb{Z}_4$ *is defined to be any non-empty subset $\mathcal{C}$ of $\mathbb{Z}_4^n$, where $\mathbb{Z}_4^n$ is the set of n-tuples over the ring of integers modulo four. The length of $\mathcal{C}$ is the positive integer n, and code words of $\mathcal{C}$ are formed as n-tuples over $\mathbb{Z}_4$. Note that if $\mathcal{C}$ is an additive subgroup of $\mathbb{Z}_4^n$, then $\mathcal{C}$ is called a $\mathbb{Z}_4$-**linear code** of length n. The generator matrix of $\mathcal{C}$ mentioned in [12] has the following form:*

$$G = \begin{bmatrix} I_{k_0} & A & B_1 + 2B_2 \\ 0 & 2I_{k_1} & 2C \end{bmatrix},$$

where A, B_1, B_2, and C have entries 0 and 1, and I_k is the identity matrix of order k.

As in [11], a code over $\mathbb{Z}_4$ is called a quaternary code, and a $\mathbb{Z}_4$-linear code is called a quaternary linear code. Some linear codes over $\mathbb{Z}_4$ that have been used in this paper are described below. The first three codes have been taken from [12].

2.1.1. Simplex Alpha Code

Let G_k be a $k \times 2^{2k}$ matrix over $\mathbb{Z}_4$ consisting of distinct columns. Inductively, G_k can be written as:

$$G_k = \left[\begin{array}{c|c|c|c} 00\ldots0 & 11\ldots1 & 22\ldots2 & 33\ldots3 \\ \hline G_{k-1} & G_{k-1} & G_{k-1} & G_{k-1} \end{array}\right],$$

where $G_1 = [0123]$. The code generated from this generator matrix G_k is called a **Simplex alpha code**. It has a length 2^{2k} and 2-dimension $2k$. Note that a code $\mathcal{C}$ is said to be a Simplex alpha code if $d_H = [\frac{d_L}{2}]$, where d_H is the minimum Hamming distance of $\mathcal{C}$ and d_L is the minimum Lee distance. For more information, see [12], which gives some fundamental features of this type of code over $\mathbb{Z}_4$.

Many non-linear binary codes are represented [13] as linear codes over $\mathbb{Z}_4$ under the Gray map φ, defined as follows: $\varphi(0) = 00$, $\varphi(1) = 01$, $\varphi(2) = 11$, $\varphi(3) = 10$. The binary code $C = \varphi(\mathcal{C})$ is called a $\mathbb{Z}_4$-linear code if $\mathcal{C}$ is a linear $\mathbb{Z}_4$ code; C need not be linear. Note that the Gray map is an isometry that transforms Lee distances defined in the quaternary codes to Hamming distances defined in the binary codes. Using the Gray map, the following codes are represented, and they are all linear codes over $\mathbb{Z}_4$.

2.1.2. Preparata Code and Kerdock Code

In 1972, Kerdock gave a construction of a $(2^m, 2^{2m}, 2^{m-1} - 2^{(m-2)/2})$ binary nonlinear code called **Kerdock code**, denoted by $K(m)$, where m is even and larger than 4.

Preparata code is a binary nonlinear code that was given by Preparata in 1968. It is denoted by $P(m)$ and has the parameters $(2^m, 2^{(2^m-2m)}, 6)$, with m being even and $m \geq 4$.

Note that, under the Gray map of linear codes over $\mathbb{Z}_4$, Preparata code and Kerdock code can be formed as binary images. Hammons et al. have shown that both Kerdock and Preparata codes are linear codes over $\mathbb{Z}_4$ in [13]. Our study considers these two codes because they have more code words than any other known linear code with the same minimal distance.

2.1.3. Hadamard Code [14]

Under the Gray map, linear codes over $\mathbb{Z}_4$ that give a binary Hadamard code are called quaternary linear Hadamard codes, and we call the corresponding $\mathbb{Z}_4$-linear codes $\mathbb{Z}_4$-linear Hadamard codes. A Hadamard code over $\mathbb{Z}_4$ that has the same parameters as the binary Hadamard code of length 2^m is one that has a length of 2^{m-1} after the Gray map. Given an integer $m \geq 3$ and an integer δ such that $\delta \in \{1, ..., [\frac{m+1}{2}]\}$, we return a Hadamard code over $\mathbb{Z}_4$ of length 2^{m-1} and type $2^\gamma 4^\delta$, where the value of m is given by the formula $m = \gamma + 2\delta - 1$. For $n = 32$ with $m = 6$, there are three possible generator matrices for the Hadamard code according to the value of δ ($\delta = 1, 2, 3$), and we chose $\delta = 3$ to form the cosets in the work of length 32 explained later.

Simplex alpha code, Preparata code, Kerdock code, and Hadamard code are the four types of codes over $\mathbb{Z}_4$ from which the cosets were derived in this paper. The next section has more details about these cosets over $\mathbb{Z}_4$.

2.2. Cosets

Niema Aboluion [9] has considered cosets of codes in synthesizing DNA codes. She began working on the cosets of linear codes in order to obtain DNA codes that satisfy HD, GC-content, and RC constraints. The following is the definition of a coset of a code over $\mathbb{Z}_4$.

Let $\mathcal{C} \subseteq \mathbb{Z}_4^n$ be a linear code of length n, and let y be any code word of length n; the coset of $\mathcal{C}$ determined by y is denoted by $\mathcal{C}(y)$ and defined as:

$$\mathcal{C}(y) = y + \mathcal{C} = \{y + c \mid c \in \mathcal{C}, y \in \mathbb{Z}_4^n\}. \quad (1)$$

Following [9], for a given ring R and positive integer n, $[n, n, 1]$, the universe code contains all possible code words over the ring R. We have computed the cosets of linear

codes over $\mathbb{Z}_4$ with $n = 4, 8, 16$, and 32. More information on code cosets can be found in [9]. Further, Aboluion's previous usage of cosets is expanded further by us. The code $\mathcal{C}$ in (1) represents either Simplex alpha code, Preparata code, or Kerdock code in this paper. The code word y has been taken from the universe code over $\mathbb{Z}_4$.

There are 40 different cosets for each value of n. Twenty of them are generated from the first code over $\mathbb{Z}_4$ that involved in the union, and the remaining twenty are derived from the second code included in the union for that particular value of n. For $n = 4$, 40 different cosets have been constructed from Simplex alpha code and Preparata code. For $n = 8$, Kerdock and Preparata codes have been used to create the cosets in the union. Simplex alpha code and Kerdock code have been involved at the union of length 16. At $n = 32$, Hadamard code and Kerdock code have been used in the union of cosets. Coset formation was instrumental in building new DNA codes with combinatorial constraints, since it gave more code words than the other methods.

3. Approaches

Since the number of cosets might often exceed a million code words, the software is required to construct them from the four codes over $\mathbb{Z}_4$. Magma [15] is a computer algebra system that can be used to compute coding theory problems. It was employed to calculate the vectors of the universe code discussed in the preceding section and to derive the code words of codes over $\mathbb{Z}_4$. Additionally, it was used to determine the maximum number of cosets of the four codes by the Magma command L : = CosetLeaders (C). In this study, we applied two methods of constructing DNA codes:

- **Algebraic Coding Approach [4]:** Fields and rings are used to create DNA codes by mapping the elements of the field and rings to DNA nucleotides.
  - **Codes over Rings:** Over the ring of integers $\mathbb{Z}_4$, DNA sequences have been created. The mapping is from $\{0, 1, 2, 3\}$ to $\{A, C, G, T\}$ with respect to the codes over $\mathbb{Z}_4$. There are two possible mappings from $\mathbb{Z}_4$ to DNA nucleotides, as shown in [9]. The first is with two non-invertible elements, combining 0 and 2 to form G, C. As a result, 0123 corresponds to $GACT$. The second mapping has one invertible element and one non-invertible element, i.e., G, C is obtained by pairing 0 with 1. That is, 0123 corresponds to $GCAT$.
- **Variable Neighborhood Search Approach:** This method uses a variety of local search methods to search the DNA codes [16,17]; one of these methods is:
  - **Seed Building (SB)**: In this method, we determined an initial set of code words with certain constraints called seed code words, and then we examined all the possible code words randomly with respect to seed code words [8].

Note that this work requires many computations with a large number of cosets and code words. Python [18] programs have been created to perform the calculations we need in this paper.

3.1 Our Method

In this section, we demonstrate our new technique for improving the lower bounds of the DNA codes.

Coset Formation

To construct the cosets of the four codes over $\mathbb{Z}_4$, we follow the next steps:

- Step 1. First, a large number of code words from the universe code with $n = 4$, $n = 8$, $n = 16$, and $n = 32$ have been taken using Magma by the following Magma command: V: = UniverseCode(Z4,n). We also obtain the code words of the four codes over $\mathbb{Z}_4$ using Magma. Now, we use Python to complete the rest of the steps.
- Step 2. Employ the coset formation technique outlined in the preceding section to create 40 cosets from the codes over $\mathbb{Z}_4$ we indicated earlier. To construct the next coset, we should make sure that our next choice of y does not exist in all the previous

cosets. Note that, as Aboluion used 40 cosets in her thesis, we also considered that number of cosets here.

- Step 3. To obtain better bounds, the union of the cosets has been applied. The following describes how the union of cosets has been constructed with the different values of n that we considered in this paper.

  1. $n = 4$
     For this length, a union of 20 cosets each from Simplex alpha code and Preparata code has been considered. Note that, although that Preparata code and Kerdock code are defined as $m = 4$, here we are using Magma output for $m = 2$. The total number of cosets in this union is 40. Each coset has four code words, since four is the size of the Simplex alpha code and Preparata code of length four. The code words of these two codes are derived by the Magma commands:
     $C := SimplexAlphaCodeZ4(k)$ and
     $P := PreparataCode(m)$.
  2. $n = 8$
     The size of the Kerdock code and Preparata code of length 8 is 256, so each coset of these two codes has 256 code words. At length 8, a union of 20 cosets each from the Kerdock and Preparata codes has been used. Therefore, at the union of this length, the Python program will check 40 cosets that contain 10,240 DNA code words.
     $K := KerdockCode(k)$ is the Magma command to get the Kerdock code over $\mathbb{Z}_4$.
  3. $n = 16$
     The union of 20 cosets each from the Simplex alpha code and the Kerdock code have been involved in the calculations. Simplex alpha code with length 16 has size 16, and each coset of this code has 16 code words. The Kerdock code of length 16 has a size of 1024, so each coset here has 1024 code words. A Python program was run here over 40 cosets with 20,800 DNA code words.
  4. $n = 32$
     Hadamard code was used with the Kerdock code at the union of this length. There are 20 cosets of size 128 from the Hadamard code of length 32 given by this command:
     $H := HadamardCodeZ4(3, 6)$ and 20 cosets of size 4096 from the Kerdock code. Therefore, the union of these codes over $\mathbb{Z}_4$ has 40 cosets needed to work on them, with 84,480 DNA code words of length 32.

After we obtain the union of the cosets from the two codes over $\mathbb{Z}_4$ at each value of n, we have the three remaining steps to complete our work on the cosets.

- Step 1. After removing the duplicated elements in the union, we map $\{0, 1, 2, 3\}$ to A, C, G, T considering the two previous mappings. Then we apply the no-repetition constraint on the code words.
- Step 2. Apply the SB approach to the union by first choosing two random seed code words that satisfy the given three constraints. These seed code words satisfy HD and GC-content constraints for a given d with $w = [\frac{n}{2}]$ and the no-repetition constraint to find lower bounds for $A_4^{GC}(n, d, w)$. Note that lower bounds depend on the choice of seed code words, and we chose seed code words with better lower bounds.
- Step 3. The program then examines all the DNA code words that exist in the set of the union and keeps those code words that satisfy the given constraints with respect to the seed code words. The program then collects these code words in a new set and gives its size to obtain lower bounds for DNA codes. Note that, for the lower bounds of $A_4^{GC}(n, d, w)$ that we obtained, we should make sure they do not exceed the upper bound of $A_4^{GC}(n, d, w)$. The upper bound has been taken from Theorem 5 in reference [19].

The following is an example of our method with $n = 4$.

Example 1. *Letting $n = 4$, $d = 4$, and $w = 2$, Table 1 shows the set of code words that represent the vector y from the universe code over $\mathbb{Z}_4$ with $n = 4$. Let C_1 be the Simplex alpha code over $\mathbb{Z}_4$ such that $C_1 = [0000, 0123, 0202, 0321]$, and let C_2 be the Preparata code over $\mathbb{Z}_4$ such that $C_2 = [0000, 1111, 2222, 3333]$. A pictorial flow is given towards the end of the paper.*

Table 1. The set of code words from the universe code over $\mathbb{Z}_4$ with $n = 4$.

0000	1000	2000	3000	0100	1100
2100	3100	0200	1200	2200	3200
0300	1300	2300	3300	0010	1010
2010	3010	0110	1110	2110	3110
0210	1210	2210	3210	0310	1310
2310	3310	0020	1020	2020	3020
0120	1120	2120	3120		

Now, construct 20 cosets of the Simplex alpha code using the formula in (1) to obtain the following cosets:

$$
\begin{aligned}
C_1(0000) &= [0000, 0123, 0202, 0321] \\
C_1(1000) &= [1000, 1123, 1202, 1321] \\
C_1(2000) &= [2000, 2123, 2202, 2321] \\
C_1(3000) &= [3000, 3123, 3202, 3321] \\
C_1(0100) &= [0100, 0223, 0302, 0021] \\
C_1(1100) &= [1100, 1223, 1302, 1021] \\
C_1(2100) &= [2100, 2223, 2302, 2021] \\
C_1(3100) &= [3100, 3223, 3302, 3021] \\
C_1(0200) &= [0200, 0323, 0002, 0121] \\
C_1(1200) &= [1200, 1323, 1002, 1121] \\
C_1(2200) &= [2200, 2323, 2002, 2121] \\
C_1(3200) &= [3200, 3323, 3002, 3121] \\
C_1(0300) &= [0300, 0023, 0102, 0221] \\
C_1(1300) &= [1300, 1023, 1102, 1221] \\
C_1(2300) &= [2300, 2023, 2102, 2221] \\
C_1(3300) &= [3300, 3023, 3102, 3221] \\
C_1(0010) &= [0010, 0133, 0212, 0331] \\
C_1(1010) &= [1010, 1133, 1212, 1331] \\
C_1(2010) &= [2010, 2133, 2212, 2331] \\
C_1(3010) &= [3010, 3133, 3212, 3331]
\end{aligned}
$$

Also construct 20 cosets of the Preparata code using the formula in (1) to obtain the following cosets:

$$
\begin{aligned}
C_2(0110) &= [0110, 1221, 2332, 3003] \\
C_2(1110) &= [1110, 2221, 3332, 0003] \\
C_2(2110) &= [2110, 3221, 0332, 1003] \\
C_2(3110) &= [3110, 0221, 1332, 2003] \\
C_2(0210) &= [0210, 1321, 2032, 3103] \\
C_2(1210) &= [1210, 2321, 3032, 0103] \\
C_2(2210) &= [2210, 3321, 0032, 1103] \\
C_2(3210) &= [3210, 0321, 1032, 2103] \\
C_2(0310) &= [0310, 1021, 2132, 3203] \\
C_2(1310) &= [1310, 2021, 3132, 0203] \\
C_2(2310) &= [2310, 3021, 0132, 1203] \\
C_2(3310) &= [3310, 0021, 1132, 2203] \\
C_2(0020) &= [0020, 1131, 2202, 3313] \\
C_2(1020) &= [1020, 2131, 3202, 0313]
\end{aligned}
$$

$$C_2(2020) = [2020, 3131, 0202, 1313]$$
$$C_2(3020) = [3020, 0131, 1202, 2313]$$
$$C_2(0120) = [0120, 1231, 2302, 3013]$$
$$C_2(1120) = [1120, 2231, 3302, 0013]$$
$$C_2(2120) = [2120, 3231, 0302, 1013]$$
$$C_2(3120) = [3120, 0231, 1302, 2013]$$

Now, create the union of these two cosets and remove the duplicated elements. Then, apply the second mapping of the elements 0123 to $GCAT$. After that, apply the homopolymers constraint to obtain the code words listed in Table 2.

Table 2. The set of DNA code words with homopolymers constraint for $n = 4$.

$GCAT$	$GAGA$	$GTAC$	$CAGA$	$CTAC$
$ACAT$	$ATAC$	$TCAT$	$TAGA$	$GTGA$
$CTGA$	$CGAC$	$ATGA$	$AGAC$	$TGAC$
$GTAT$	$GCAC$	$CTAT$	$ATAT$	$ACAC$
$TCAC$	$GCGA$	$CGAT$	$AGAT$	$ACGA$
$TGAT$	$TCGA$	$GACA$	$CGCG$	$CACA$
$AGCG$	$TGCG$	$TACA$	$GACG$	$CTAC$
$AGTA$	$TCGT$	$CACG$	$ATAC$	$TGTA$
$GCGT$	$TACG$	$GTAC$	$CGTA$	$ACGT$
$GTCG$	$CGAC$	$ACTA$	$TAGT$	$CTCG$
$AGAC$	$TCTA$	$GAGT$	$ATCG$	$TGAC$
$GCTA$	$CAGT$	$CGAG$	$ACTC$	$TAGA$
$GTCT$	$AGAG$	$TCTC$	$GAGA$	$CTCT$
$TGAG$	$GCTC$	$CAGA$	$ATCT$	$GCAG$
$CATC$	$ATGA$	$TGCT$	$ACAG$	$TATC$
$GTGA$	$CGCT$	$TCAG$	$GATC$	$CTGA$
$AGCT$				

Next, apply the SB approach to the DNA code words in Table 2 using these two seed-building code words: $CATC$, $TCGA$ with $d = 4$ and $w = 2$. Choose only DNA code words with $w = 2$ to compare with $CATC$ and $TCGA$. For example, take $ATCG$: $d_H(CATC, ATCG) = 4$ and $d_H(TCGA, ATCG) = 4$. After checking all the code words in the table, we see that $ATCG, GTCT, AGAG$, and $AGCT$ are all have a distance of 4 from the SB code words, and their $w = 2$. Therefore, the final set of DNA code words that satisfies the homopolymers, HD, and GC-content constraints is: $[ATCG, GTCT, AGAG, AGCT]$. Therefore, $A_4^{GC}(4, 4, 2) = 4$. The approach is summarized in Figure 1.

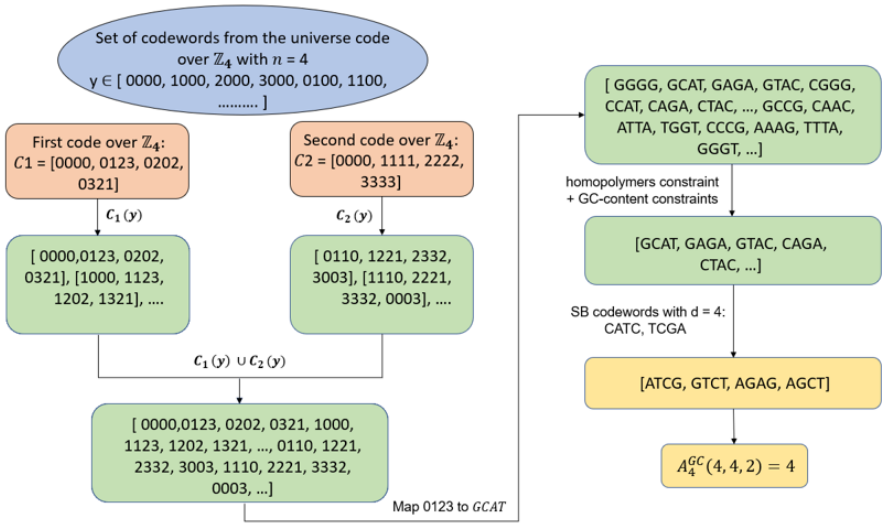

Figure 1. Our approach applied on the example $n = 4$, $d = 4$, $w = 2$.

4. Results

Finally, the program gives the lower bounds for DNA codes after completing all steps described in the previous section. Table 3 shows the improved lower bounds that were obtained by our method. Note that all values are computed by map2, since map1 does not give any good results.

Table 3. Solutions obtained by our approach for different size instances.

(n, d)	**Lower Bound with Homopolymers Constraint**
$(4, 4)$	4
$(16, 13)$	12
$(16, 14)$	3
$(32, 27)$	8

We obtained better results than existing bounds based on our comprehensive approach to the $n = 4$ instance for $d = 4, n = 16$ for $d = 13$, and $n = 32$ instance for $d = 27$. In the first two cases, we obtain the same result as in the previous bound, and in the last case with $n = 32$, we are getting very close to the upper bound. Therefore, our method is very good, since it gives us either the previous known bound or close to the upper bound. Additionally, length $n = 32$ with $d = 27$ has not previously been dealt with by the SB approach in the literature. All the calculations are given on this web site, accessed on 22 October 2023 https://dsr-asa.kau.edu.sa/Pages-DNA-code-design-based-on-the-cosets-of-codes.aspx.

For the DNA word design problem $(n = 4; d = 4), (n = 16; d = 13), (n = 16; d = 14)$, and $(n = 32; d = 27)$ we identified a solution of size $4, 12, 3$, and 8, respectively. The solution is given in Tables 4–7 for these values of n with the homopolymers constraint.

Table 4. Solution of 4 words for the $n = 4$, $d = 4$, $w = 2$ instance with the homopolymers constraint; SB code words are: $CATC$ and $TCGA$.

$ATCG$	$GTCT$
$AGAG$	$AGCT$

Table 5. Solution of 12 words for the $n = 16$, $d = 13$, $w = 8$ instance with the homopolymers constraint; SB code words are: $GTCATCTCAGAGCTCT$ and $GCATGTCGATGCAGAT$.

$CTCAGAGAGAGAGAGA$
$AGCAGAGAGAGAGAGA$
$TGCAGAGAGAGAGAGA$
$CAGTATGCGCATATGC$
$TCGACAGCTACGTCTA$
$CGCGAGTGACTATGTA$
$AGTCACTAGCATGAGC$
$ATAGCGACTGCATCTG$
$ACGACAGCTACGTCTA$
$TGTCACTAGCATGAGC$
$TAGCACTAGCATGAGC$
$TATGCGACTGCATCTG$

Table 6. Solution of 3 words for the $n = 16$, $d = 14$, $w = 8$ instance with the homopolymers constraint; SB code words are: $GCATGTCGATGCAGAT$ and $GTCATGAGTACGTAGC$.

$AGTAGCTCGCTAGTGT$
$TGTAGCTCGCTAGTGT$
$CATAGCTCGCTAGTGT$

Table 7. Solution of 8 words for the $n = 32$, $d = 27$, $w = 16$ instance with the homopolymers constraint; SB code words are: $TAGCAGCTCACACATGTGCACATGTGCACATG$ and $TGCATGCGTCACAAGTGTCTGTACACGCATAC$.

$ACGTGATCTGTGTGCACATGTGCACATGTGCA$
$TCGTGATCTGTGTGCACATGTGCACATGTGCA$
$CACTGATCTGTGTGCACATGTGCACATGTGCA$
$GACTGATCTGTGTGCACATGTGCACATGTGCA$
$CTCTGATCTGTGTGCACATGTGCACATGTGCA$
$GTCTGATCTGTGTGCACATGTGCACATGTGCA$
$AGCTGATCTGTGTGCACATGTGCACATGTGCA$
$TGCTGATCTGTGTGCACATGTGCACATGTGCA$

5. Conclusions

This paper aimed to improve the lower bounds for DNA codes that satisfy significant constraints, constructed from the cosets of linear codes over $\mathbb{Z}_4$. By adding the homopolymers constraint, these vectors are more suitable for DNA synthesis applications since they satisfy the homopolymers constraint; this kind of constraint is not well-known in the literature. Therefore, our approach is a comprehensive approach. We obtained four better lower bounds, and the approach is new.

Magma software V2.24-4 and Python programming were used for the implementation, and we succeeded in building new Python programs that calculate bounds for $n = 4, 8, 16$, and 32 with various values of d. The comprehensive approach presented here found four better lower bounds. These lower bound improvements shown in Table 3 satisfy the HD constraint, the GC-content constraint, and the homopolymers constraint and are excellent results that demonstrate the effectiveness of our technique.

In this paper, we focused on $n = 4$, $n = 8$, $n = 16$, and $n = 32$ because of the types of codes over $\mathbb{Z}_4$ that we considered in the union. For $n = 4$, indeed, there are only 4 cosets of Kerdock code at length 4. For a future target, we can use a different combination of codes at the union of the cosets; we may combine Simplex alpha code with Preparata code as well as Kerdock code with Preparata code at length 16 to obtain new DNA codes. In this paper, we are focusing on degree one for the homopolymers constraint. In the next paper, we may construct a higher degree of conflict-free DNA codes. We hope our technique can be further applied to other values of n and also to other codes over finite rings such as $\mathbb{Z}_{p^s}$.

Author Contributions: Conceptualization, F.A.M., M.K.G. and A.N.A.; methodology, F.A.M.; software, F.A.M.; validation and formal analysis, F.A.M., A.N.A. and M.K.G.; investigation, F.A.M.; resources, F.A.M. and A.N.A.; writing—original draft preparation, F.A.M.; writing—review and editing, F.A.M., A.N.A. and M.K.G.; visualization, F.A.M.; supervision, M.K.G. and A.N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Saudi Arabia under grant no. (KEP-MSc-65-130-42). The authors, therefore, acknowledge with thanks DSR technical and financial support.

Data Availability Statement: The computational data generated are available at https://dsr-asa.kau.edu.sa/Pages-DNA-code-design-based-on-the-cosets-of-codes.aspx, accessed on 22 October 2023.

Acknowledgments: The authors gratefully acknowledge that this project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Saudi Arabia under grant no. (KEP-MSc-65-130-42). The authors, therefore, acknowledge with thanks DSR technical and financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science* **1994**, *266*, 1021–1024. [CrossRef] [PubMed]
2. Smith, D.H.; Aboluion, N.; Montemanni, R.; Perkins, S. Linear and nonlinear constructions of DNA codes with Hamming distance d and constant GC-content. *Discret. Math.* **2011**, *311*, 1207–1219. [CrossRef]
3. Brenner, S.; Lerner, R.A. Encoded combinatorial chemistry. *Proc. Natl. Acad. Sci. USA* **1992**, *89*, 5381–5383. [CrossRef] [PubMed]
4. Limbachiya, D.; Rao, B.; Gupta, M.K. The art of DNA strings: Sixteen years of DNA coding theory. *arXiv* **2016**, arXiv:1607.00266.
5. Benerjee, K.G.; Deb, S.; Gupta, M.K. On conflict free DNA codes. *Cryptogr. Commun.* **2021**, *13*, 143–171. [CrossRef]
6. Dougherty, S.T.; Korban, A.; Sahinkaya, S.; Ustun, D. Construction of DNA codes from composite matrices and a bio-inspired optimization algorithm. *IEEE Trans. Inf. Theory* **2022**, *69*, 1588–1603. [CrossRef]
7. Gaborit, P.; King, O.D. Linear constructions for DNA codes. *Theor. Comput. Sci.* **2005**, *334*, 99–113. [CrossRef]
8. Montemanni, R.; Smith, D.H.; Koul, N. Three metaheuristics for the construction of constant GC-content DNA codes. *Lect. Notes Manag. Sci.* **2014**, *6*, 167–175.
9. Aboluion, N.A. *The Construction of DNA Codes Using a Computer Algebra System*; University of South Wales: Newport, UK, 2011.
10. Aboluion, N.; Smith, D.H.; Perkins, S. Linear and nonlinear constructions of DNA codes with Hamming distance d, constant GC-content and a reverse-complement constraint. *Discret. Math.* **2012**, *312*, 1062–1075. [CrossRef]
11. Wan, Z.H. *Quaternary Codes*; World Scientific: Singapore, 1997; Volume 8.
12. Gupta, M.K. *On Some Linear Codes over* $\mathbb{Z}_{2^s}$; Indian Institute of Technology: Kanpur, India, 1999.
13. Hammons, A.R.; Kumar, P.V.; Calderbank, A.R.; Sloane, N.J.A.; Solé, P. The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes. *IEEE Trans. Inf. Theory* **1994**, *40*, 2. [CrossRef]
14. Barrolleta, R.D. Partial Permutation Decoding for $\mathbb{Z}_4$-Linear Hadamard and Kerdock Codes. Ph.D. Thesis, Universitat Autònoma de Barcelona, Barcelona, Spain, 2016.
15. Cannon, J.; Bosma, W.; Fieker, C.; Steel, A. *Handbook of Magma Functions*; The University of Sydney: Sydney, Australia, 2006.
16. Kawashimo, S.; Ono, H.; Sadakane, K.; Yamashita, M. Dynamic neighborhood searches for thermodynamically designing DNA sequence. In *International Workshop on DNA-Based Computers*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 130–139.
17. Hansen, P.; Mladenović, N.; Perez, J.A.M. Variable neighborhood search: Methods and applications. *Ann. Oper. Res.* **2010**, *175*, 367–407. [CrossRef]
18. Beazley, D.M. *Python Essential Reference*; Sams Publishing: Carmel, IN, USA, 2006.
19. King, O.D. Bounds for DNA codes with constant GC-content. *arXiv* **2003**, arXiv:0306197.