The construction of DNA codes using a computer algebra system

Niema Ali Aboluion

Mathematics and Statistics
Faculty of Advanced Technology
University of Glamorgan
Pontypridd, CF37 1DL, Wales, U.K.

A submission presented in partial
fulfilment of the requirements of the
University of Glamorgan/Prifysgol Morgannwg
for the degree of Doctor of Philosophy

April 19, 2011

Contents

List of Tables

Acknowledgments

I am heartily thankful to my supervisor, Professor Derek Smith, who has supported me throughout my thesis with his patience, kindness and knowledge. Without his support and guidance, my thesis would not have been possible.
I am grateful also to my second supervisor, Dr. Stephanie Perkins for all of her help and encouragement.
Finally, I thank my family, my parents, my brother Jumah and my friends for supporting me throughout my study.

Abstract

Coding theory has several applications in Genetics and Bioengineering. This thesis concentrates on a specific application from Computational Biology. This concerns the construction of new DNA codes which satisfy certain combinatorial constraints, using an alphabet of four symbols. The interest in these codes arises because it is possible to synthesise short single strands of DNA known as oligonucleotides. The codes can be useful in the design of these oligonucleotides. For example, the codes are used in DNA computing, as bar codes in molecular libraries and in microarray technologies.

The computer algebra system Magma, which deals successfully with coding theory computation, is applied initially to the construction of DNA codes satisfying a GC-content constraint and a minimum Hamming distance constraint. The constraints are specified to avoid unwanted hybridizations and to ensure uniform melting temperatures. Additionally, another constraint, known as a reverse-complement constraint, is added to further prevent unwanted hybridizations. This additional constraint is studied using involutions in a permutation group. Codes constructed in this thesis are derived from linear codes over $GF(4)$ and Z_4 and additive codes over $GF(4)$. Previous approaches to the construction of these codes are extended in several ways. Longer codes are constructed, the examination of cyclic and extended cyclic codes is more comprehensive, and cosets of codes are considered. In addition, attention is paid to the mapping from field or ring elements to the DNA nucleotides; different mappings can give different lower bounds. Further improvements have been made after the techniques of shortening and puncturing are applied to the table of best codes, and also by searching for codes in the tables that have an all-ones vector in their dual. The use of a database of best known linear codes is also considered. In many cases codes are obtained which are larger than the best codes currently known. In the case of codes of length greater than twenty, linear DNA codes have not been constructed previously and so all codes obtained are the best known results. Generator polynomials are given for the codes constructed. Coset leaders are also given in cases where cosets of linear codes are used. Thus it is possible for the reader to construct the codes without repeating the work presented in the thesis. Additionally, files of codewords are available online when the codes constructed are the best known codes and have fewer than 50000 codewords.

Chapter 1

Introduction

1.1 Introduction to DNA codes

There has been growing interest recently in the applications of coding theory in genetics and bioengineering, in particular to the construction of synthetic deoxyribonucleic acid (DNA) strands. The design of codes that satisfy combinatorial constraints has been studied for many years, motivated by the problem of sending information reliably over a noisy channel. The aim of this thesis is to design DNA strand sets that are suitable for DNA computing and for other applications involving synthetic DNA. The combinatorial constraints that must be satisfied are different here. Every DNA molecule consists of two complementary strands which are sequences of four different nucleotide bases. These are called adenine (A), cytosine (C), guanine (G) and thymine (T). As a result, each strand can be regarded as a word constructed from the alphabet {A,C,G,T}.

Direct applications of techniques of coding theory arise in the construction of these synthetic DNA strands (*oligonucleotides*). The quality of a set of DNA strands depends on the possibilities for *hybridization*, the process of joining two single strands of DNA to form a double stranded molecule. The strands can be used as probes in DNA microarray technologies. The highly predictable hybridization chemistry of DNA can also be exploited in the use of oligonucleotides as tags or bar codes in chemical libraries [6]. These tagged libraries can be used for drug screening purposes. A further application is in DNA computing, where a critical step is to construct an appropriate encoding of the problem in DNA oligonucleotide sequences in such a way that hybridization finds the desired solution. Unwanted cross-hybridizations can introduce errors and reduce efficiency. The library of words must be large enough to represent the necessary information.

Most attempts at word design for DNA computation have used combinatorial methods; see for example [30]. Several papers have proposed different techniques to construct a set of DNA codewords that are unlikely to form undesirable bonds with each other by hybridization. In [16] four different constraints on a DNA code $\mathcal{C}$ are considered. These are the Hamming distance constraint, the reverse constraint, the reverse-complement constraint and the fixed GC-content constraint. References to earlier papers considering these constraints are also given in [16]. The purpose of the first and third constraints is to make

non-desirable hybridization unlikely to occur. The fixed GC-content constraint is used to ensure that similar melting temperatures are obtained, where DNA melting is the process by which double-stranded DNA unwinds and separates into single strands through the breaking of hydrogen bonding between the bases. This allows hybridization of multiple words to take place simultaneously [35]. Similar melting temperatures can be approximately achieved by ensuring that each word contains the same number of positions which are either G or C, referred to as constant GC-content [15]. This is because GC base-pairing is generally stronger than AT base-pairing, which is due to one extra hydrogen bond in the GC pair. The four constraints are defined as follows:

1. Let $H(x, y)$ denote the Hamming distance between two words (i.e. the number of positions in which they differ). The Hamming distance constraint is that $H(x, y) \geq d$ for all $x, y \in \mathcal{C}$ with $x \neq y$, for some prescribed minimum distance d.

2. The reverse constraint is that $H(x^R, y) \geq d$ for all $x, y \in \mathcal{C}$, where x^R is the reverse of a codeword x. Note that $x = y$ is included.

3. The reverse-complement constraint is that $H(x^{RC}, y) \geq d$ for all $x, y \in \mathcal{C}$, where x^{RC} is the reverse-complement of x obtained by taking x^R and performing the symbol interchanges $A \leftrightarrow T$, $C \leftrightarrow G$ (this is called taking Watson-Crick complements). Again $x = y$ is included.

4. The GC-content constraint is that each codeword $x \in \mathcal{C}$ has the same GC-content. The GC-content of a DNA word is defined to be the number of positions in which the word has coordinate C or G.

1.2 Computer algebra systems

The software packages Maple and Magma have been used in teaching and research in many Universities throughout the world. They can be used for the construction of linear and non-linear codes. It is important to assess the relative merits of the two packages for the construction of codes.

1.2.1 Maple

Maple is an advanced and powerful system for symbolic computation, designed to cover many different application areas. It was first conceived in November 1980 by the symbolic computation group at the University of Drexel and the Swiss Federal Institute of Technology (ETH) in Zurich. The researchers' main purpose was to create a symbolic system accessible to researchers and students, providing good facilities for symbolic computation [29], [42]. The Maple computer algebra package has extensive facilities designed for doing complicated mathematics quickly and precisely in such areas as algebra, calculus, geometry, linear algebra and number theory [24]. Some of the features of Maple are:

1. A *Help* facility is available

2. Maple can draw curves, surfaces and data plots in two and three dimensions.

3. Maple can solve a variety of problem types, such as equations and inequalities, ordinary differential equations, partial differential equations, integer equations and linear systems of equations.

4. Maple can be used to add, subtract, multiply, or divide numbers or algebraic expressions.

5. Maple has a compiler facility which is included in versions of Maple from Maple 11, This is able to compile numerical Maple procedures to native code in order to accelerate computations.

For more details refer to [20] and [21].

1.2.2 Magma

Magma is a computer algebra system designed to deal with a wide variety of problems in algebra, number theory, geometry and combinatorics. It is produced and distributed by the Computational Algebra Group within the School of Mathematics and Statistics of the University of Sydney. In June 1996 Magma *Version 2.0* was released and after that a new version of Magma has been released approximately once per year. Magma contains many of the most advanced and efficient known algorithms for the areas which it covers, such as, groups, rings, fields, algebras, vector spaces, algebraic geometries, lattices, graphs, combinatorics and codes. It has facilities provided for linear codes over fields $GF(q)$, including codes constructed in terms of generator matrices, parity check matrices and generating polynomials. This facility will be used later to construct linear codes. In addition, a large number of constructions for particular families of codes, (e.g. quadratic residue codes) are available. There are also algorithms for the calculation of the minimum weight and weight enumerator, including the MacWilliams transform. For more details refer to [41].

1.2.3 Maple, Magma and binary codes

The two software packages Maple and Magma have been applied to construct several classes of $[n, k, d]$ binary linear codes such as Hamming codes, BCH codes, Reed Muller codes and cyclic codes. For more details about the constructions of these codes refer to Appendix I. In fact Magma has many built-in facilities which aid the construction of linear codes. It is shown in Appendix I that Magma software is very much more convenient for the construction of linear codes, and makes the construction of $[n, k, d]$ binary linear codes much easier than Maple, saving both time and effort in developing the software. In terms of computation time, Magma appears generally to be faster than Maple. On this basis Magma should be applied to construct codes that are suitable for DNA computing and other DNA applications in preference to Maple.

1.3 Aims and achievement of the work

In this thesis the problem of designing DNA codes is considered. This problem is motivated by the task of reliably storing and retrieving information in synthetic

DNA strands, for use in DNA computing or as molecular bar codes in chemical libraries. Good strand design is important in order to minimize errors due to non-specific hybridization (where not all pairs consisting of a base of a DNA strand and a corresponding base of a reverse strand are complementary bases), to achieve a higher information density, and to obtain large sets of strands for large scale applications. Techniques from coding theory have been used to provide constructions and bounds on sets of code strands satisfying the four constraints specified in Section 1.1. Appendix 2 present a collection of tables listing the best known bounds for code sets over alphabets of four symbols. Following [16], the maximum number of codewords of length n, minimum Hamming distance d and GC-content w is denoted $A_4^{GC}(n, d, w)$. Also, $A_4^{GC,RC}(n, d, w)$ denotes the maximum size of a code satisfying a reverse-complement constraint of length n, minimum Hamming distance d with GC-content w. As the actual value of w is unimportant, the aim of the thesis is to improve lower bounds for $\max_w(A_4^{GC}(n, d, w))$ and $\max_w(A_4^{GC,RC}(n, d, w))$. The maximum often occurs for $w = \lfloor n/2 \rfloor$, but there are exceptions [16].

Gaborit and King [16] proposed new constructions for DNA codes satisfying the required constraints, derived from additive and linear codes over $GF(4)$ and from linear codes over Z_4 . The emphasis here will be on these linear and additive code constructions, rather than on algorithmic methods, such as those recently presented in [31]. In comparison with the work done by Gaborit and King [16], constructions will be extended in several directions, in order to obtain improved lower bounds for $\max_w(A_4^{GC}(n, d, w))$ and $\max_w(A_4^{GC,RC}(n, d, w))$. In [16] specific families of linear and additive codes were considered. This work will concentrate on cyclic and extended cyclic codes and codes that can be derived from them. This will allow a much more comprehensive approach to be used. In detail, the extensions are:

1. In [16, 31] codes are constructed with $n \leq 20$; here this is extended to $n \leq 30$. Longer codes are certainly important in applications; for example, the DNA microarray technology *Affymetrix* use $n = 25$ base pairs.

2. In the case of constructions of DNA codes with constant GC-content and DNA codes with constant GC-content with the addition of a reverse-complement constraint, a comprehensive search of all possible cyclic or extended cyclic codes is carried out. This is done by considering all possible generator polynomials.

3. In [16] a fixed mapping from the field or ring to $\{A, C, G, T\}$ is used. Here it is shown that the mapping chosen is unimportant for linear codes over GF(4). However, for both additive codes and codes over $\mathbf{Z}_4$ the 24 possible mappings reduce to two essentially different cases, which can give different lower bounds.

4. When $n \leq 20$, cosets of the cyclic and extended cyclic codes are considered in addition to the codes themselves.

5. As well as the standard puncturing and shortening operations, a nonlinear shortening operation which sometimes gives more codewords is applied.

6. Some of the codes need only be described by their generator polynomials. Codes obtained from cosets also need the coset leader for their construc-

tion. A notation is introduced that allows all shortening and puncturing operations to be replicated from the information given in the table of results. Thus all codes given here should be easily reproducible without the need to repeat the search that led to their selection.

The general approach used to find the cyclic codes is to compute all possible generator polynomials. The cyclic codes found are also used to find codes derived from them by extending, puncturing, shortening or considering cosets. For each code found, its complete weight enumerator and minimum Hamming distance is computed. This allows the GC-weight enumerator to be determined. From the GC-weight enumerator the maximum number of codewords of constant GC-weight is determined. The code is tested against the best current code of that length and minimum distance. In the case of codes satisfying a reverse-complement constraint a method using involutions of the permutation group is also applied, as proposed in [16].

1.4 Structure of the Thesis

The remainder of the thesis is organized as follows: Chapter 2 reviews the literature on DNA codes, coding theory in Magma and the computer algebra system Magma.

Chapter 3 gives an introduction to the construction of linear codes.

Chapter 4 gives detailed descriptions of the construction of DNA codes satisfying a Hamming distance constraint, with constant GC-content.

Chapter 5 gives detailed descriptions of the construction of DNA codes satisfying a Hamming distance constraint, with constant GC-content and with the addition of a reverse-complement constraint.

Chapter 6 aims to improve previous results by searching for codes that have an all-ones vector in their dual.

Chapter 7 aims to improve previous results by using a database of the best linear codes known.

Chapter 8 summarises the results of this work and concludes the thesis.

Appendix I assesses the relative merits of the software packages Maple and Magma for the construction of linear codes.

Appendix II includes all tables of codes constructed.

Chapter 2

Literature review

2.1 DNA codes

DNA molecules are now used for purposes that go far beyond their function in nature, and can be used as another way of coding information by dealing with DNA as a sequence in a test tube rather than a cell. Biomolecular computing is a new domain of study that is concerned with the use of biological molecules as fundamental components of computing devices. This requires considerable knowledge in a variety of different fields such as chemistry, physics and molecular biology, in order to know about the physical and chemical properties of DNA and the laboratory techniques for handling DNA. Also it relies on expertise in computer science and mathematics. The first successful DNA computation using strands of DNA was in 1994, when Adleman [2] demonstrated in a wet-lab the solution of an example of a known problem in combinatorics using standard tools of molecular biology and this first suggested the use of random sets of DNA strands for code design. He implemented an algorithm to solve a seven-point Hamiltonian Path Problem (HPP) by using DNA, molecular chemistry and the storage capacity of DNA. Although Adleman opened the door for more work on DNA computing, his approach was specific to solving the HPP. Lipton [26] realised this and proposed extending Adleman's algorithm in a way that allows DNA computers to solve other problems, not just the Hamiltonian Path Problem. In 1995 Lipton described a methodology for solving the satisfiability problem (SAT) using DNA. The underlying principle was the same as Adleman's approach by generating all possible solutions to the problem, then gradually filtering out strands until any remaining strands must be a solution to the problem. Lipton aimed to show how any difficult problem may be solved using this approach. Since these initial experiments, interest in DNA computing has increased dramatically, and it is now a well-established area of research. There are now many papers describing significant developments in molecular computing. Shapiro and his team [3] demonstrated a simple molecular scale autonomous programmable computer composed of enzymes and synthetic strands of DNA. This allowed both input and output information to be in molecular form. Such a computer device is able to detect signs of cancer, and release an anti-cancer drug to treat the disease. Since then, other researchers have demonstrated new computational systems that make use of enzymes that naturally

occur in living cells.

DNA codes are also used as molecular bar codes, or tags, in chemical libraries to improve the drug discovery process. In 1992 Brenner and Learner [6], [5] exploited the power of genetic systems by extending the range of analysis to chemicals that are not part of biological systems. He encoded each molecule of a chemically synthesized entity to a particular oligonucleotide sequence by using two parallel combinatorial syntheses which allow the genetic tag to be linked to the synthesized chemical structure. In 1996 Shoemaker *et al.* used a highly parallel molecular bar coding strategy to analysis yeast deletion mutants by labelling each deletion strain with a unique 20-base tag sequence. This helped in determining the biological function of thousands of newly opened reading frames in Saccharomyces Cerevisiae [35].

DNA applications use DNA microarrays. These are hard flat surfaces made of plastic, glass, fused silica or gold, with maximally large sets of DNA oligonucleotides called probes fixed on the surface. They have been used in studying the expression of large numbers of genes in a single experiment [36]. Also, they help in the investigation of many biological problems in different areas include gene discovery, disease diagnosis and species identification. The implementation of the idea of attaching multiple DNA sequences on a plate was achieved by Fodor et al. [14], and improved by Affymetrix, a California-based biotechnology company. Accuracy in the results of microarray applications require good design strategies for probes, with good quality sets of unique probes that should not be able to hybridize well with incorrect targets in the solution. Many techniques are used to design DNA strands for microrrays, some use combinatorial criteria and some use thermodynamic criteria for more accurate results.

Good applications require high quality sets of strands to achieve a higher information density. In these applications the important operations performed are specific hybridizations between the oligonucleotides and their Watson-Crick complements, forming the double helix. On this basis, avoiding undesirable hybridizations by reducing the probability of hybridization error for sets of DNA strands, is one of the main concerns in DNA bioengineering. Both combinatorial and thermodynamic properties can be considered. This leads us to DNA word design problems where the main concern is to identify maximal sets of codewords satisfying certain constraints. Combinatorial constraints include uniform GC-content across strands so that desired hybridization between strands and their complements have similar melting temperatures. They also include specifying a large number of mismatches between two strands.

Many approaches have been considered for DNA word design. From the perspective of combinatorial codes, several papers proposed different algorithms to design DNA strands in order to limit nonspecific hybridization in specific applications. Deaton *et al.* [12], [13] proposed the use of Hamming distance for DNA word design by describing genetic algorithms for finding DNA codes that satisfy Hamming and reverse complement constraints. Marathe *et al* [30] proposed dynamic programing algorithms based on Hamming distance and free energy and give upper and lower bounds for the dimension of DNA codes. Brenner [5] introduced algorithms for the generation of DNA codes satisfying a Hamming distance constraint. Frutos *et al.* [15] proposed a template-map strategy used for storing and manipulating information in DNA molecules attached to a surface. Li *et al.* [27] demonstrated a template map strategy for designing sets of non interacting DNA strands for applications in DNA and provided

lower bounds on $A_4^{GC}(n,d,w)$ and $A_4^{GC,RC}(n,d,w)$ for limited ranges of length and dimension. King [23] derived theoretical upper and lower bounds on the maximum size of DNA codes for all parameters n, d, w using a lexicographic construction to find explicit codes that improve on many of the lower bounds in [27],[39], [40]. Tulpan *et al* [39] used stochastic search algorithms to design DNA strands. Montemanni and Smith [32], [31] developed and combined four new local search algorithms into a variable neighbourhood search framework which gives improved codes in many cases. In term of an algebraic approach, Rykov *et al* [34] used sets of reverse complement cyclic codes to generate DNA codes. However, Rykov's results were limited to reversible cyclic codes and asymptotic lengths. Gaborit and King [16] used techniques from coding theory to construct codes that satisfy certain combinatorial constraints. They provided a survey of the best lower bounds for the size of oligonucleotide libraries. On the other hand, thermodynamic criteria offer more accurate measures of hybridization. Many papers have used thermodynamic constraints to design high quality sets of DNA oligonucleotides in order to maximize desired hybridizations between strands and their complements [38].

2.2 Magma

The computer algebra system Magma has already been mentioned in Section 1.2.2. It will now be described in more detail, together with the associated literature. Magma is designed to solve problems in various Mathematical areas. It enables users to rapidly formulate and perform calculations in the more abstract parts of mathematics such as algebra, number theory, geometry and combinatorics. It runs on Unix-like and Linux based operating systems, as well as Windows. From 1982 to 1993 the predecessor of the Magma system was called Cayley. Magma was officially released in August 1993 as *Version 1.0*. *Version 2.0* of Magma was released in June 1996 and subsequent versions of the form 2.X have been released approximately once per year.

In late 2006, the book Discovering Mathematics with Magma [4], was published. The book introduces the reader to the role Magma plays in advanced mathematical research through 14 case studies. Simultaneously it introduces a new programming language in the context of contemporary research problems, and gives an exposition of the types of problem that can be investigated using computational algebra. The Computational Algebra Group maintains a list of publications which cite Magma [41], and as of 2010 there are about 2600 citations, mostly in pure mathematics, but also including papers from areas as diverse as economics and geophysics. The Handbook of Magma Functions [10], constitutes the main reference work on Magma. It aims to provide a comprehensive description of the Magma language and the mathematical facilities of the system. In particular, it documents every function and operator available to the user. It was based on a similar Handbook written for Cayley in 1990. Up until 1997 the Handbook was mainly written by Wieb Bosma, John Cannon and Allan Steel but in more recent times, as Magma expanded into new areas of mathematics, additional people became involved.

Magma provides good performance both in terms of the algorithms used and their implementation. The facilities provided in Magma include algorithms for factorizing polynomials. In fact, the Magma command Factorization(f) is

used to generate all generator polynomials for the codes constructed in this work. Given a univariate polynomial f over the ring R, this command returns the factorization of f as a factorization sequence Q, that is, a sequence of pairs, each consisting of an irreducible factor q_i and a positive integer k_i (its multiplicity). The coefficient ring R must be one of the following: a finite field F_q, the ring of integers Z, the field of rationals Q, an algebraic number field $Q(x)$, or a polynomial ring, function field (rational or algebraic) or finite-dimensional affine algebra (which is a field) over any of the above. For factorization over very small finite fields, the Berlekamp algorithm is used, which depends on fast linear algebra [25], [18]. For certain codes in Chapter 5, especially those with a large permutation group, it is easy to use Magma to construct the conjugacy classes of the permutation group in order to check whether or not an involution exists. The algorithm used for computing the conjugacy classes of a permutation group is Derek Holt's algorithm [22]. This is a significantly improved algorithm for computing the conjugacy classes of a finite permutation group obtained by generalizing the Cannon/Souvignier method [11] to all classes of the group, avoiding the random part of the Cannon/Souvignier method.

2.2.1 Coding theory in Magma

Magma has extensive tools for computing in coding theory. In addition, Magma has access to a large mathematical database containing information that may be used in searches for interesting examples or which form an integral part of certain algorithms. Examples of current databases include *best known linear codes*, a database that gives the user access to every best known linear code over GF(2) of length up to 256, and best known linear codes over GF(4) in all but 40 of 5150 cases of length up to 100. Many of the codes constructed in this database are vast improvements on the previously known bounds for best codes over GF(4). The Magma database of best linear codes, is based on results in a web site previously maintained by A.E. Brouwer [7] and now superseded by a web site maintained by Grassl [19]. Many entries in the Magma best known linear codes database provide better codes than the corresponding ones listed originally by Brouwer. Most of the machinery Magma provides for studying linear codes over finite fields is based on the theory described by MacWilliams and Sloane [28]. For example, algorithms used for the calculation of the minimum weight and various forms of weight enumerator, together with constructions of new codes from existing codes such as construction by shortening and puncturing, can be found in [28].

Chapter 3

Introduction to the construction of linear codes

Linear codes are an important class of codes used in error correction and detection schemes. In this section basic definitions and facts on the construction of linear codes are given.

3.1 Linear codes

The work in this section deals with linear codes over the finite field $GF(q)$.

Definition 1 *Let v be a word of length n. The* weight *of v is the number of non-zero coordinates in v.*

Definition 2 *Let C be a code and let $x, y \in C$. The* Hamming distance $d(x, y)$ *is the number of positions in which x and y differ.*

Definition 3 *The minimum distance of the code C is $d(C) = \min\{d(x, y) \mid x, y \in C, x \neq y\}$.*

Definition 4 *A linear code C of length n and dimension k is a linear subspace of a vector space F_q^n of n-vectors over the field $F = GF(q)$.*

An equivalent formulation of a linear code is that C is linear if $a_1c_1 + a_2c_2 \in C$ for all $c_1, c_2 \in C$ and $a_1, a_2 \in F$.

Each vector in the code is called a codeword and the size of C, $\mid C \mid$ is q^k, where k is called the dimension of C.

A code with size M and minimum distance d is referred to as an (n, M, d) code.

Definition 5 *A $k \times n$ matrix G in which the rows are a basis of C, is called a* generator matrix *of C.*

Example 1 *If the matrix*

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

is a generator matrix of a code over $GF(2)$, then the codewords are:

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{array}$$

Definition 6 *The* dual code $C^{\perp}$ *is the set of all vectors orthogonal to all codewords of the code C.*

An equivalent formulation of the *dual code* is that it consist of the set of vectors y such that $x \cdot y = 0$ for all x in C.

The generator matrix of the dual code is called the *parity check matrix* H. It is an $(n-k) \times n$ matrix.

Proposition 1 *Let G be a generator matrix for a code C and let H be a parity check matrix. Then $GH^T = 0$.*

For the construction of binary codes using the software packages Maple and Magma refer to Appendix I.

3.2 Hamming codes

Hamming codes are a simple class of linear codes which are easy to encode and decode, and allow the correction of all single bit errors. They were invented by R. W. Hamming in 1950.

Definition 7 *A binary linear code of length $n = 2^m - 1, m \geq 2$, with parity check matrix H whose columns consist of all nonzero vectors of length m is called* a Hamming code*, and is characterized by the parameters $[n, k] = [2^m - 1, 2^m - 1 - m]$.*

Example 2 *The $[7, 4]$ Hamming code has a parity check matrix.*

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

A generator matrix for $[7, 4]$ Hamming code will be,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Thus, the code has $2^{2^m - 1 - m} = 2^4 = 16$ codewords.

For the construction of Hamming codes in Maple and Magma refer to Appendix I.

3.3 Reed-Muller codes

This section is devoted to the study of the construction of another class of codes known as *Reed-Muller-codes*. The codes are described in terms of Boolean functions.

Definition 8 *Let $V = (v_1, v_2, \ldots, v_m)$ range over the set of all binary m-tuples. A function $f(V) = f(v_1, v_2, \ldots, v_m)$ which also takes one of the values* 0 *and* 1 *is called a* Boolean function.

Definition 9 *The r^{th} order binary* Reed-Muller-code *$R(r, m)$ with parameters $[2^m, 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}, 2^{m-r}]$, for $0 \leq r \leq m$, is the set of all vectors corresponding to the Boolean functions $f(V)$ of degree at most r evaluated at each m-tuple of V in turn.*

In fact, the Reed-Muller-code $R(r, m)$ consists of all linear combinations of the vectors$(1, v_1, v_2, \ldots, v_m)$ corresponding to products:

$$1, v_1, \ldots, v_m, v_1v_2, v_1v_3, \ldots, v_{m-1}v_m, \ldots, v_{m-r+1} \ldots v_{m-1}v_m.$$

These products form a basis for the code of dimension $k = 1 + \binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{r}$.

Example 3 *The basis vectors of the Reed-Muller-code $R(1, 4)$ are:*

$1:$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$v_1:$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$v_2:$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$v_3:$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$v_4:$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

The construction of Reed-Muller codes using Maple and Magma is illustrated in Appendix I.

3.4 Cyclic codes

Another class of linear codes are *cyclic codes*. These codes have found important applications in error detection and correction. The class of cyclic codes contains many other codes, such as BCH codes and certain Hamming codes. This section begins by defining cyclic codes, before discussing their algebraic structure and some properties.

Definition 10 *A linear code C of length n over a field $GF(q)$ is called a cyclic code, if for every codeword $v = (v_1, v_2, \ldots, v_n) \in C$, the word $(v_n, v_1, \ldots, v_{n-1})$ which is obtained by cyclic shift of components, is also a codeword in C.*

A cyclic code can be generated by a codeword v and cyclic shifts of its components. In fact, it is easy to describe cyclic codes in term of polynomials, where the codeword $(v_0, \ldots, v_{n-1})$ of a cyclic code C is represented by the polynomial $v_0 + xv_1 + \ldots + v_{n-1}x^{n-1}$. In this case the elements of a cyclic codeword are associated with the coefficients of the polynomial.

On the other hand, the cyclic code can be defined to be an ideal in the algebra of polynomials modulo $x^n - 1$.

Example 4 *The cyclic code* $C = \{000, 110, 101, 011\}$ *can be represented in terms of polynomials of the form* $C(x) = \{0, 1+x, 1+x^2, x+x^2\}$. *Since* C *is a linear code, it is closed under addition, and for every* $v(x) \in C(x)$, $xv(x)$ *mod* (x^3-1) *is also in* $C(x)$. *Thus* C *is an ideal.*

In addition, the code C can be generated by taking the unique element $g(x)$ in $C(x)$ of minimum degree and computing $x^i g(x)$ mod $1+x^n$ for $i = 0, 1, \ldots, n-1$. This can be written as $C(x) =< g(x) >$. $g(x)$ is called the *generator polynomial*.

Theorem 1 *[28] Let* C *be a cyclic code of length* n. *There is a unique monic polynomial* $g(x)$ *of minimal degree in* $C(x)$ *such that* $C(x) =< g(x) >$ *and* $g(x)$ *is a factor of* $x^n - 1$.

The generator matrix G of a cyclic code C can be obtained easily from the generator polynomial $g(x)$.

Theorem 2 *[28] If* $g(x) = g_0 + g_1 x + \ldots + g_m x^m$ *is the generator polynomial of a cyclic code* C, *then* C *has a basis* $B = [g(x), xg(x), \ldots, x^{n-m-1}g(x)]$ *and a generator matrix*

$$G = \begin{bmatrix} g_0 & g_1 & ... & g_m & 0 & 0 & 0 & ... & 0 \\ 0 & g_0 & g_1 & ... & g_m & 0 & 0 & ... & 0 \\ 0 & 0 & g_0 & g_1 & ... & g_m & 0 & ... & 0 \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & ... & g_m \end{bmatrix}$$

It can be seen that in G every row consists of a right cyclic shift of the row above it.

To see how Maple and Magma can be used to construct cyclic codes refer to Appendix I.

3.5 Finite fields

Finite fields form an essential part of the study of error-correcting codes. The purpose of this section is to describe the construction of a finite field.

Definition 11 *A* finite field *is a field with a finite number of elements.*

The order of a finite field is the number of elements in the field. It is always of the form p^m, where p is a prime number called the *characteristic* of the field and the arithmetic in a finite field is performed modulo p.

Theorem 3 *[28] For every prime number* p *and integer* $m \geq 1$, *there exist a finite field with* p^m *elements.*

A finite field of order p^m is usually called a *Galois field* and is denoted by $GF(p^m)$.

A finite field $GF(p^m)$ where $m > 1$ can be represented as a set of polynomials over $GF(p)$; the field will contain p^m distinct polynomials $r(x)$ and the arithmetic is performed modulo $\pi(x)$, where $\pi(x)$ is an irreducible polynomial of degree m over $GF(p)$.

Example 5 *Consider the field $GF(2^3)$ of order 8 and let $f(\alpha) = 1 + \alpha + \alpha^3$ (an irreducible polynomial).*

Then $\alpha^3 \equiv \alpha + 1$ and the field elements that correspond to subsequent power of α can be constructed as follows.

Power	*Field elements*
0	$-$
α^0	1
α^1	α
α^2	α^2
α^3	$\alpha + 1$
α^4	$\alpha(\alpha^3) \equiv \alpha(\alpha + 1) \equiv \alpha^2 + \alpha$
α^5	$\alpha(\alpha^2 + \alpha) \equiv \alpha^2 + \alpha + 1$
α^6	$\alpha(\alpha^2 + \alpha + 1) \equiv \alpha^2 + 1$

The set of polynomials in the second column are closed under addition and multiplication modulo $\alpha^3 + \alpha + 1$. It can easily be shown that $\alpha^7 \equiv \alpha(\alpha^2 + 1) \equiv \alpha^3 + \alpha \equiv 1$ modulo $\alpha^3 + \alpha + 1$.

On the other hand, the field elements can be represented as m-tuples of polynomial coefficients.

In the above example the elements can be described as 3-tuples as follows.

$3 - tuple$	*Field elements*
000	$-$
100	1
010	α
001	α^2
110	$\alpha + 1$
011	$\alpha^2 + \alpha$
111	$\alpha^2 + \alpha + 1$
101	$\alpha^2 + 1$

Now, it is easy to compute the sums of elements using the 3-tuples and the products of elements using the powers of α. For example, the product of the elements $\alpha^2 + 1$ and $\alpha^2 + \alpha$ in $GF(2^3)$ is

$$(\alpha^2 + 1)(\alpha^2 + \alpha) = (\alpha^6)(\alpha^4) = \alpha^{10} = \alpha^{10 mod 7} = \alpha^3 = \alpha + 1$$

Maple and Magma have a package *GF* for constructing finite (*Galois*) fields. To see how this package is used to construct finite fields, refer to Appendix I.

3.6 BCH codes

In Coding Theory the *Bose-Chaudhuri-Hocquengham codes*, or *BCH* codes, form an important class of multiple-error-correcting codes. This section discusses the construction of 2-error-correcting BCH codes as generalizations of Hamming codes.

By adding m more rows to the parity check matrix H of a Hamming code of length $n = 2^m - 1$, the parity check matrix $H^{'}$ of the $[2^m - 1, 2^m - 2m - 1]$ BCH code is obtained.

Example 6 *Consider the parity check matrix H of the* $[15, 11]$ *Hamming code.*

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

As has been seen before, the columns in the matrix H (4-tuples) can be represented as polynomials of the form α^{i-1}. Thus, the matrix H can be written in the form:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \end{bmatrix}$$

Now by adding 4 more rows to H, where each column in these rows is a 4-tuple, the i^{th} column of a new matrix $H^{'}$ can be created of the form

$$H^{'} = \begin{pmatrix} \alpha^{(i-1)} \\ \alpha^{3(i-1)} \end{pmatrix}$$

The matrix $H^{'}$ is:

$$H^{'} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{bmatrix}$$

By expanding $H^{'}$, the parity check matrix of a $[15, 7]$ BCH *is obtained.*

$$H^{'} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The construction of BCH codes in Maple and Magma is described in more detail in Appendix I. It should be noted that both *BCH* and *Hamming* codes are always equivalent to cyclic codes, and so can be obtained by a cyclic construction.

3.7 The weight enumerators of linear codes

This section is devoted to the study of Hamming weight enumerators of linear codes over $GF(q)$.

Definition 12 *Let C be an $[n, k, d]$ linear code over $GF(q)$ and A_i be the number of codewords of weight i in C. The polynomial $\sum_{i=0}^{n} A_i z^i$ is called the* weight enumerator *of C and denoted by $W_C(z)$.*

The weight enumerator of its dual can be defined in the same way as follows: $W_{C^\perp}(z) = \sum_{i=0}^{n} A_i^{'} z^i$, where $A_i^{'}$ denotes the number of codewords of weight i in $C^\perp$.

Example 7 *Consider the* $[7, 4, 3]$ *Hamming code*

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

which has the dual code

$$C^{\perp} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The weight enumerator of the Hamming code is:

$$W_C(z) = 1 + 7z^3 + 7z^4 + z^7$$

and the weight enumerator of its dual is:

$$W_{C^{\perp}}(z) = 1 + 7z^4$$

The following famous theorem of MacWilliams relates the weight enumerator of a code to the weight enumerator of the dual code.

Theorem 4 *[43]* (MacWilliams theorem for linear codes) Let C be a linear code of dimension k over $GF(q)$ with Hamming weight enumerator $W_C(z)$ and let $C^{\perp}$ be its dual code with weight enumerator $W_{C^{\perp}}(z)$, then

$$W_C(z) = q^k \left(q^{-1} + \frac{(q-1)}{q} z\right)^n W_{C^{\perp}}\left(\frac{1-z}{1+(q-1)z}\right). \tag{3.1}$$

Substituting $z = \frac{(1-y)}{1+(q-1)y}$ gives:

$$W_{C^{\perp}}(y) = q^{n-k} \left(q^{-1} + \frac{(q-1)}{q} y\right)^n W_C\left(\frac{1-y}{1+(q-1)y}\right) \tag{3.2}$$

which relates the weight enumerator of the dual code to the weight enumerator of the code in the same way. Equations (3.1) and (3.2) are known as the *MacWilliams Identities*. More details can be found in [28].

Example 8 *Consider the codes of Example 7. The dual Hamming code has weight enumerator* $W_{C^\perp}(z) = 1 + 7z^4$*, and by using* MacWilliams Theorem

$$\begin{aligned} W_C(y) &= 2^4 \left(\frac{1+y}{2}\right)^7 \left(1 + 7\left(\frac{1-y}{1+y}\right)^4\right) \\ &= \frac{1}{2^3}\left[(1+y)^7 + 7(1+y)^3(1-y)^4\right] \\ &= \frac{1}{8}\left[(1 + 7y + 21y^2 + 35y^3 + 35y^4 + 21y^5 + 7y^6 + y^7) + \right. \\ &\quad \left. 7(1 - y - 3y^2 + 3y^3 + 3y^4 - 3y^5 - y^6 + y^7)\right] \\ &= 1 + 7y^3 + 7y^4 + y^7 \end{aligned}$$

which is the weight enumerator of the Hamming code. Similarly, using the weight enumerator of the Hamming code and the MacWilliams identity, it can be checked that the weight enumerator of the dual Hamming code is $W_{C^\perp}(z) = 1 + 7z^4$.

The MacWilliams identity can be obtained easily using Maple. Appendix I illustrates how this can be done.

3.8 Additive codes over $GF(4)$

In addition to linear codes, additive codes over $GF(4)$ are considered in this thesis. These additive codes are additive subspaces of $(GF(4))^n$. They have a generator matrix with rows consisting of vectors over $GF(4)$, but only binary linear combination are considered (i.e. the scalars are from $GF(2)$).

Example 9 *Consider the generator matrix obtained from two linearly independent vectors in* $(GF(4))^3$

$$G = \begin{bmatrix} 1 & 0 & w^2 \\ 0 & w & 0 \end{bmatrix}$$

The size of linear code over $GF(4)$ *generated by* G *is* $4^2 = 16$

$$C_1 = \begin{bmatrix} 1 & 1 & \omega^2 \\ \omega & 1 & 1 \\ \omega^2 & 1 & \omega \\ 0 & 1 & 0 \\ 1 & \omega & \omega^2 \\ \omega & \omega & 1 \\ \omega^2 & \omega & \omega \\ 0 & \omega & 0 \\ 1 & \omega^2 & \omega^2 \\ \omega & \omega^2 & 1 \\ \omega^2 & \omega^2 & \omega \\ 0 & \omega^2 & 0 \\ 1 & 0 & \omega^2 \\ \omega & 0 & 1 \\ \omega^2 & 0 & \omega \\ 0 & 0 & 0 \end{bmatrix}$$

The corresponding GF(2)-additive code will contain $2^2 = 4$ *codewords*

$$C_2 = \begin{bmatrix} 1 & \omega & \omega^2 \\ 0 & 0 & 0 \\ 1 & 0 & w^2 \\ 0 & w & 0 \end{bmatrix}$$

The codewords of C_2 *arise only through addition of the generators; scalar multiplication is not permitted.*

Chapter 4

DNA codes satisfying a Hamming distance constraint and with constant GC-content

The Magma system provides facilities which support the construction of linear codes over $GF(q)$, additive codes over $GF(4)$ and linear codes over Z_4. For more details about linear codes refer to Chapter 3. In addition, highly optimized algorithms for the calculation of the minimum weight and the complete weight enumerator are available. Full details of the facilities used can be found in [10]. In this chapter these facilities are applied to construct codes satisfying a Hamming distance constraint and with constant GC-content. Following [16], codes can be considered which are sets of words of fixed GC-content w in (i) linear codes over the field GF(4), (ii) linear codes over the ring $\mathbf{Z}_4$ and (iii) additive codes, which are additive subspaces of $(\mathrm{GF}(4))^n$ (refer to 3.8). The aim is to obtain codes which have more codewords than the best previously known codes as presented at [17].
The *complete weight enumerator* of a code C over an alphabet of four symbols is a four variable polynomial

$$cw(x, y, z, w) = \sum_{u \in C} x^{s_1(u)} y^{s_2(u)} z^{s_3(u)} w^{s_4(u)}$$

where $s_i(u)$ is the number of components of the codeword u equal to the ith symbol of the alphabet.

The *GC weight enumerator* of a code C of length n over an alphabet $\{A, C, G, T\}$ is a two variable polynomial

$$GC(X, Y) = \sum_{u \in C} X^{n - s_{G,C}(u)} Y^{s_{G,C}(u)}$$

where $s_{G,C}(u)$ is the number of components of the codeword u equal to G or C. For a fixed value of $s_{G,C}(u)$ the coefficient of $X^{n - s_{G,C}(u)} Y^{s_{G,C}(u)}$ gives the number of codewords of GC-content $s_{G,C}(u)$. These codewords can be

selected to give a subcode of constant GC-content. The approach presented here of computing the *GC weight enumerator* will be used later with codes which also satisfy the reverse-complement constraint. Many of the best codes obtained are cyclic, extended cyclic, or can be obtained from such codes from (repeated) shortening and puncturing. Here constructions of such codes will be extended in several directions, in order to obtain improved lower bounds for $\max_w(A_4^{GC}(n,d,w))$

4.1 Constructions of cyclic and extended cyclic codes

In this section we use the theory of cyclic codes to construct codes satisfying the constraints specified. In particular, both linear and additive cyclic and extended cyclic codes over $GF_(4)$ and linear cyclic and extended cyclic codes over Z_4 are constructed.

4.1.1 Cyclic codes construction

Cyclic codes have been defined in Definition 10.

Definition 13 *A DNA code of length n is a set of codewords* $\{x \mid x = (x_1, x_2, \ldots, x_n)\}$ *with* $x_i \in \{A, C, G, T\}$ *(representing the four nucleotides in DNA).*

A finite field $GF(4)$ can be represented as a set of polynomials over $GF(2)$; the field will contain 4 distinct polynomials $r(x)$ and the arithmetic is performed modulo $p(x)$, where $p(x)$ is an irreducible polynomial of degree 2 over $GF(2)$.

Example 10 *Let* $f(\alpha) = 1 + \alpha + \alpha^2$ *(an irreducible polynomial).*

Then $\alpha^2 \equiv \alpha + 1$ *and the field elements that correspond to subsequent powers of* α *can be constructed as follows.*

Power	*Field elements*
0	$-$
α^0	1
α^1	α
α^2	$\alpha + 1$

The set of polynomials in the second column are closed under addition modulo 2 *and multiplication modulo* $\alpha^2 + \alpha + 1$.

Fuller details of the construction of a finite field are given in Chapter 3.

Following [16], The field elements $0, 1, \omega, \omega^2 = 1 + \omega$ are mapped to the alphabet $\{A, C, G, T\}$ in some order.

Definition 14 *An additive code C over* $GF(4)$ *of length n is an additive subgroup of* $(GF(4))^n$.

As C is a free $GF(2) - module$, it contains 2^k, codewords for some $0 \leq k \leq 2n$. We call C an $(n, 2^k)$ code. It has a basis, as a $GF(2) - module$, consisting of k basis vectors. A generator matrix of C will be a $k \times n$ matrix with entries in GF(4) whose rows are a basis of C.

Definition 15 *A linear code C of length n over Z_4 is defined to be an additive submodule of the Z_4 module Z_4^n.*

The elements $0, 1, 2, 3 \in Z_4$ are identified with the nucleotides A, C, G, T in some order. An equivalent formulation of cyclic codes over Z_4 is that Z_4 is cyclic if and only if it is an ideal in the ring $R = Z_4[x]/(x^n - 1)$.

The following three theorems describe the construction of cyclic codes in the three cases considered:

Theorem 5 *[28] Let $\mathcal{C}$ be an $[n, k]$ linear cyclic code of length n over GF(4). Then $\mathcal{C} = \langle g(x) \rangle$ where $g(x)$ is a monic polynomial of degree $n - k$ in GF(4)$[x]$ that divides $x^n - 1$.*

Theorem 6 *[8] Let $\mathcal{C}$ be an $(n, 2^k)$ additive cyclic code of length n over $GF(4)$ (with elements $0, 1, \omega, \omega^2$). Then $\mathcal{C} = \langle \omega p(x) + q(x), r(x) \rangle$ where $p(x)$, $r(x)$ are binary polynomials that divide $(x^n - 1)$ mod 2, $r(x)$ divides $q(x)(x^n - 1)/p(x)$ mod 2, and $k = 2n - deg\ p - deg\ r$.*

Note that, if $\langle \omega p(x) + q(x), r(x) \rangle$ and $\langle \omega p'(x) + q'(x), r'(x) \rangle$ are two representations of an additive cyclic code then $p'(x) = p(x)$, $r'(x) = r(x)$ and $q'(x) \equiv q(x)$ mod $r(x)$.

The structure of ideals in R is more complicated than the structure of ideals in $GF(4)[x]$. In the cyclic codes over fields the cyclic codes are principal ideals generated by a divisor of $x^n - 1$. For rings several divisors of $x^n - 1$ may be necessary. In the case where n is odd, $(x^n - 1)$ factors uniquely over Z_4. However for even n, the factorization of $x^n - 1$ over Z_4 is not unique. For example, $x^4 - 1 = (x - 1)(x + 1)(x^2 + 1) = (x - 1)(x - 1)(x^2 + 2x - 1) = (x + 1)(x + 1)(x^2 + 2x - 1)$ over Z_4.

The structure of cyclic codes over Z_{p^m} has been studied by Calderbank and Sloane [9], who found the simplest form of the set of ideals in R.

Theorem 7 *[9] If $q = p^a$, $1 \leq a < \infty$, any ideal in R has the form*

$$< f_0(x), pf_1(x), \ldots, p^{a-1} f_{a-1}(x) >$$

where the $f_i(x)$ are divisors of $x^n - 1$ satisfying

$$f_{a-1}(x) \mid f_{a-2}(x) \mid \ldots \mid f_0(x)$$

In the case $R = Z_4$, every cyclic code is of the form $< f(x), 2g(x) >$, where $g(x) \mid f(x) \mid x^n - 1$ in $Z_4[x]$, and $f(x)$, $g(x)$ are monic polynomials.

Two approaches proved feasible for the generation of polynomials. Either $x^n - 1$ was factorized using Magma and all combinations of factors were considered. Alternatively, all polynomials $P(x)$ in GF(4)$[x]$, GF(2)$[x]$ or $\mathbf{Z}_4[x]$ of degree up to $\lfloor (n+1)/2 \rfloor$ were taken, (together with the polynomial $(x^n - 1)/P(x)$), but the polynomial was only used in the appropriate construction if it divided $x^n - 1$. To construct all cyclic codes in the three cases requires generation of:

- a single polynomial in the case of linear cyclic codes over GF(4). Both methods above can be used, but only the first approach is described in detail here. All cyclic codes over GF(4) are constructed by first factorizing the polynomial $f(x) = x^n - 1 \in GF(4)[x]$ using the Magma command $Factorization(x^n - 1)$. The Magma command *CyclicCode(n, g)* is used to construct the cyclic code C over GF(4) of length n corresponding to each generator polynomial $g(x)$ (where $g(x)$ is a product of factors). Repeated factors are dealt with as described for binary linear cyclic codes in Appendix I. Linear cyclic codes over GF(4) were computed for $4 \leq n \leq 30$.

- three binary polynomials (with the third satisfying a divisibility condition) in the case of additive codes over GF(4). In this case the second approach is used. After creating all the possible choices for the binary polynomials $p(x)$, $q(x)$ and $r(x)$, a single representative for each equivalence class mod $r(x)$ is produced by choosing the remainder when $q(x)$ is divided by $r(x)$. This is achieved by ensuring that the degree of the $q(x)$ generated is less than the degree of the polynomial $r(x)$ generated. Next, the command $IsDivisibleBy$ is used to test whether or not $r(x)$ divides $q(x)(x^n - 1)/p(x)$ *mod* 2. If it does, then the Magma command $AdditiveCyclicCode(n, g, r)$ is used to construct the cyclic additive code C of length n corresponding to the two generator polynomials $g(x) \in GF(4)[x]$ and $r(x) \in GF(2)[x]$ where $g(x) = \omega p(x) + q(x)$. Additive cyclic codes were computed for $4 \leq n \leq 20$.

- two polynomials of $Z_4[x]$ with the second satisfying a divisibility condition in the case of linear cyclic codes over $\mathbf{Z}_4$. Since the factorizations of $x^n - 1$ are no longer unique, the Magma command $Factorization(x^n - 1)$ would fail to factorize $x^n - 1$ in the case of n even. However, $IsDivisibleBy(f, g)$ can be used to test all possible factors of $x^n - 1$ to see if they divide exactly, and the quotient is given if they do. If the two polynomials $f(x)$ and $g(x)$ both divide $x^n - 1$ and $g(x) \mid f(x)$, the code will be $< f(x), 2g(x) >$. Two codes are generated, $< f(x) >$, $< 2g(x) >$ using the Magma commands $C1 := CyclicCode(n, f)$ and $C2 := CyclicCode(n, 2 * g)$. The vector space sum $C := C1 + C2$ is then taken because Magma does not contain a two polynomial command. All cyclic codes over $\mathbf{Z}_4$ were computed for $4 \leq n \leq 15$.

In general, the use of a single polynomial $f(x)$ usually gave the best results, so the restricted set of all cyclic codes of the form $\langle f(x) \rangle$ was computed for $16 \leq n \leq 23$. A GC-weight enumerator was calculated in each case (see the beginning of this chapter and [16]) and the code with the largest number of codewords became a candidate for realising a lower bound for $\max_w(A_4^{GC}(n, d, w))$. The algorithms used to provide these facilities are outlined in the Magma documentation.

4.1.2 Extended cyclic codes construction

Extended cyclic codes are obtained by adding an extra position and an overall parity check [28]. This can be implemented using the Magma command $C := ExtendCode(C0, 1)$ before computing a complete weight enumerator, GC-weight enumerator and minimum Hamming distance. The aim is again to find

codes with constant GC-content which have more codewords than the previously known codes. The code is then tested against the best current code of that minimum distance. A similar computation was carried out for all linear extended cyclic codes with $4 \leq n \leq 30$ and additive extended cyclic codes for $4 \leq n \leq 20$. All extended cyclic codes over $\mathbf{Z}_4$ were computed for $4 \leq n \leq 16$ and all extended codes from cyclic codes of the form $\langle f \rangle$ were computed for $17 \leq n \leq 24$.

4.2 Mappings from field or ring elements to $\{A, C, T, G\}$

Here there are three cases to consider.

4.2.1 The mapping of the elements $\{0, \omega, \omega^2, 1\}$ to $\{A, C, T, G\}$ in the case of linear codes over GF(4)

There are 24 possible mappings from $\{0, \omega, \omega^2, 1\}$ to $\{A, C, T, G\}$. Before presenting results for cyclic codes over GF(4), it will be shown that for linear codes over GF(4) the actual mapping chosen is unimportant. If a cyclic code over GF(4) exists providing a lower bound for $A_4^{GC}(n, d, w)$ for a particular value of w, then a code providing a lower bound for $A_4^{GC}(n, d, n-w)$ can be obtained by the pair of transpositions $A \leftrightarrow G$, $T \leftrightarrow C$. As the aim is to find the maximum number of codewords of constant GC-weight irrespective of the actual GC-weight, it is unimportant whether the element 0 maps to a letter of $\{A, T\}$ or of $\{C, G\}$. Similarly, the transpositions $A \leftrightarrow T$, $G \leftrightarrow C$ do not affect the constant GC-weight property. Therefore it is unimportant whether the zero element maps to the letter A,C,T or G. Suppose without loss of generality that the zero element maps to G and consider the three different mappings $1 \to C$, $\omega \to C$, $\omega^2 \to C$. These can be obtained from $1 \to C$ by multiplying the codewords of the cyclic code by 1, ω and ω^2 respectively. Such multiplications give automorphisms of the cyclic code, and so the same lower bound is obtained.

4.2.2 The mapping of the elements $\{0, \omega, \omega^2, 1\}$ to $\{A, C, T, G\}$ in the case of additive codes over GF(4)

In additive codes over $GF(4)$ either ω or ω^2 can be chosen to multiply the first generator polynomial, and an isomorphic code is obtained, i.e.:

$$\langle \omega p(x) + q(x), r(x) \rangle \cong \langle \omega^2 p(x) + q(x), r(x) \rangle$$

This follows from the existence of the Frobenius automorphism $f(x) = x^2$ that maps $0 \to 0$, $1 \to 1$, $\omega \to \omega^2$, $\omega^2 \to \omega$, which means that if we interchange ω and ω^2, the same field will be obtained. On this basis it can be seen that there are two cases which should be considered in computing additive codes over $GF(4)$. Pairing 0 with ω or 0 with ω^2 for G,C gives the same GC-weight enumerator. However, pairing 0 with 1 for G,C can give different GC-weight enumerators to the previous two cases. Thus two distinct mappings need to be considered for additive codes over $GF(4)$.

4.2.3 The mapping of the elements $\{0, 1, 2, 3\}$ to $\{A, C, T, G\}$ in the case of linear codes over Z_4

In $\mathbf{Z}_4$, pairing 0 with 2 to give G,C can give different GC-weight enumerators to the other two cases (pairing 0 with 3 or 0 with 1). This occurs because 0 and 2 have no inverse. The cases of pairing 0 with 1 and 0 with 3 are the same. One can be obtained from the other by multiplying all codewords by -1. Thus two distinct mappings need to be considered.

4.3 Cosets

Definition 16 *If C is a linear code of length n, and if u is any word of length n, the coset of C determined by u is the set of all words of the form $u+v$, $v \in C$.*

It was observed that cosets of linear codes sometimes give more codewords of constant GC-content than the linear codes themselves. The approach used here is to compute all possible generator polynomials and generate all coset leaders, or as many as possible. In fact, the Magma command,

```
L:=CosetLeaders(C)
```

can be used to return a set of coset leaders (a word of minimum weight in any particular coset) of the cyclic code C, and works well if the number of coset leaders L is up to 4^{10} =1048576. However, It does not work in the case that the number of coset leaders is at least $4^{11} = 4194304$. This is because calculating so many coset leaders requires too much memory. On this basis such cases were omitted and a test on the number of coset leaders was avoided by choosing a selection of 40 coset leaders randomly. For each coset found, its complete weight enumerator and minimum Hamming distance is computed. This allows the GC-weight enumerator to be determined. From the GC-weight enumerator the maximum number of codewords of constant GC-weight is determined and the best candidate code found is recorded.

The command $CosetLeaders(C)$ can not be used in cyclic codes over Z_4 in order to produce the set of coset leaders of C, therefore, random codewords taken from the set of all possible codewords of C produced using the command $UniverseCode(R, n)$, are used to compute the coset of cyclic codes over Z_4. In addition, the command $CompleteWeightEnumerator(C, u)$ which is normally used to compute the complete weight enumerator for the coset $C + u$, is not available for cyclic codes over Z_4 and although the command does exist in the additive code package in Magma, it does not seem to work correctly. Instead the complete weight-enumerator for the coset $C + u$, is computed directly for cyclic codes over Z_4 and additive codes over GF(4). Cosets were considered for linear codes over GF(4) and over $\mathbf{Z}_4$ with $4 \leq n \leq 20$, for additive cyclic codes with $4 \leq n \leq 15$, and for additive extended cyclic codes with $4 \leq n \leq 16$. The reason why the computation was not continued for $n \geq 21$ is that the calculation of codes obtained from cosets takes a long time to compute. For example, three weeks in total were spent constructing codes for $n \leq 20$.

4.4 Shortening and puncturing

The idea of shortening and puncturing is to modify a good code in order to find from it a shorter code that is also good. The techniques can be applied to a linear code more than once. Here the definition of shortening and puncturing as given in [28], [16] is used.

- *Shortening:* is to construct a new code from a linear code C by selecting only those codewords of C having a zero as their i^{th} component and deleting the i^{th} component from these codewords. Thus, the resulting code will have length $n-1$, but normally fewer codewords. Here the Magma command $ShortenCode(C, i)$ is used to shortening the code C at the position i. The minimum distance of a shortened code is at least as great as that of the original code.

- *Puncturing:* is to construct a new code C' by deleting the i^{th} coordinate from each code word of the code C. Thus, the resulting code will have length $n-1$, and normally has the same number of codewords. Here the Magma command $PunctureCode(C, i)$ is used to puncture the code C at the position i. Puncturing will normally decrease the distance between codewords by 1.

In fact, Tables 10.25 and 10.26, which give best lower bounds for $A_4^{GC}(n, d, w)$, are improved after shortening and puncturing. Starting from the bottom row of the table, each new row is computed from the row beneath. When a new best code is constructed this replaces the existing best code before moving to the row above.

A GC-weight enumerator can be computed and a code of constant GC-weight can then be selected. However, a nonlinear shortening operation sometimes gives more codewords. Given a code $\mathcal{C}$ of constant GC-content over $\{A, C, G, T\}$, compute the frequency of each letter A, C, G, T in each column i of the matrix of codewords. Choose the letter and column of the most frequent occurrence and select all codewords with the chosen letter in the chosen column. Delete this component from all selected codewords and a (normally nonlinear) code of constant GC-content is obtained. As the operation in nonlinear it is only feasible for shorter codes. Normally shortening gives an unchanged Hamming distance, and puncturing reduces the minimum distance by 1. Sometimes, however, the minimum distance is greater than is anticipated. Thus it is necessary to assess all possible shortenings and puncturing of a given code in the table to find the best codes with length reduced by 1.

4.5 Results

Tables 10.1 to 10.24 give the best lower bounds for $A_4^{GC}(n, d, w)$ for cyclic and extended cyclic codes over $GF(4)$ and Z_4, and for their cosets. They also give the best lower bounds for $A_4^{GC}(n, d, w)$ for cyclic additive and extended cyclic additive codes over $GF(4)$ and for their cosets. These codes satisfy the Hamming distance constraint and GC-content constraint. It can be seen that this comprehensive calculation of codes has produced many new bests. In some cases codes are obtained which are larger than the best codes currently known.

As can be seen in Tables 10.1, 10.2, 10.3 and 10.4, the calculation of codes obtained from cyclic and extended cyclic codes over GF(4) has produced six new best cyclic codes and two new best extended cyclic codes with $n \leq 20$. Tables 10.5 and 10.7 give lower bounds for $A_4^{GC}(n, d, w)$ for the best coset of a cyclic code and coset of an extended cyclic code over $GF(4)$. Many cosets are not calculated because there are many cases where there are too many cosets. In order to fill in the missing cases, a selection of coset leaders is chosen randomly. As a result, non-complete results can be inserted in several cases, as shown in Tables 10.6, 10.8. This calculation of codes obtained from cosets of cyclic codes over GF(4) has produced six new best codes and one new best is obtained from cosets of extended cyclic codes over GF(4) with $n \leq 20$. In the Tables 10.9, 10.10, the calculation of codes obtained from cyclic additive codes over GF(4) has produced good results with $n \leq 20$ for both mappings. There are 13 new best codes when pairing 0 with 1 to give the GC-weight and 15 new best codes when pairing 0 with ω or ω^2 to give the GC-weight. In Tables 10.11, 10.12, it can be seen that just one new best code with $n \leq 20$ is obtained in both cases of the codes obtained from extended cyclic additive codes over GF(4). The calculation of codes obtained from cosets of cyclic additive codes over GF(4) produces three new best codes with $n \leq 20$ in the case of pairing 1 with ω (see Table 10.13), and six new best codes with $n \leq 20$ in the case of pairing 0 with 1, (see Table 10.14). The calculation of codes obtained from cosets of cyclic additive codes over GF(4) has just one new best code with $n \leq 20$ in both cases as shown in Tables 10.15, 10.16. It can be seen that the comprehensive calculation of codes obtained from cyclic, extended cyclic, cosets of cyclic and cosets of extended cyclic codes over Z_4 has not produced any new best codes with $n \leq 20$, (see Tables 10.17 to 10.24). However, some codes are better than the corresponding code over $GF(4)$. Tables 10.25 and 10.26 summarise the results for best codes with minimum Hamming distance d and constant GC content for all computations. The results in this table are improved after applying the techniques of shortening and puncturing, with 31 new bests. This is shown in Tables 10.27, 10.28. Cases with $n \geq 20$ have not been computed before and so are all new bests.

The work described in this chapter will appear in Discrete Mathematics [37].

Chapter 5

DNA codes satisfying a Hamming distance constraint, with constant GC-content and satisfying a reverse-complement constraint

This chapter concerns the addition of a further constraint to control the way that the DNA strings might hybridise. This is the reverse complement constraint, which more accurately models the biological requirements. This constraint can be studied using involutions in a permutation group, as has been shown by Gaborit and King [16]. Magma is particularly suitable for studying this interaction of permutation groups and linear codes. The codes found in the previous chapter can be modified to construct codes that also satisfy the reverse-complement constraint. Many improvements to the best known values for $\max_w(A_4^{GC,RC}(n,d,w))$ are found, and tables of the best known codes in this case can be constructed. Note that there were no cases in the previous chapter where a code gave a larger number of codewords using a linear code over the ring $\mathbf{Z}_4$ than could be obtained from a linear code over GF(4) or an additive code. Thus linear codes over $\mathbf{Z}_4$ are not considered further here.

5.1 Involutions

Definition 17 *A permutation group is a finite group G whose elements are permutations of a given set and whose group operation is composition of permutations in G.*

Definition 18 *Two codes over K are permutation-equivalent if one can be obtained from the other by permuting the columns (coordinates).*

Definition 19 *The permutation group of a code of length n is the group of permutations of $\{1, 2, 3, \ldots, n\}$ that, when applied to the columns of the code, maps the code to itself.*

The use of involutions to handle the reverse complement constraint was pioneered by Gaborit and King [16], who stated and proved the following lemma:

Lemma 1 *[16] Let $\mathcal{C}'$ be a code of length n such that:*

- *$n = 2k$ is even and $\mathcal{C}'$ has a fixed-point free involution in its permutation group (i.e. a permutation of the form $(a_1, a_2) \cdots (a_{2k-1}, a_{2k})$ which leaves no column unchanged); or*
- *$n = 2k+1$ is odd and $\mathcal{C}'$ has a one-point fixed involution in its permutation group (i.e. a permutation of the form $(a_1, a_2) \cdots (a_{2k-1}, a_{2k})$ which leaves one column unchanged).*

Then $\mathcal{C}'$ is permutation equivalent to a code $\mathcal{C}$ that has the reverse permutation R in its permutation group.

The lemma is proved simply by considering the permutation that sends column a_{2i-1} to column i and column a_{2i} to column $n + 1 - i$, for $1 \leq i \leq k$ (and a_{2k+1} to $k + 1$ if n is odd). The code $\mathcal{C}$ can be written as a disjoint union $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \mathcal{C}_2$, where $\mathcal{C}_0$ is the set of codewords fixed by R and $\mathcal{C}_1$, $\mathcal{C}_2$ are two sets that are interchanged by R. Either the set of codewords $\mathcal{C}_1$ or the set $\mathcal{C}_2$ can be chosen as a code that satisfies the reverse constraint and the Hamming distance constraint for the value of d prescribed for $\mathcal{C}'$.

Magma has facilities which can deal easily with the permutation group of the linear code. The Magma command *PermutationGroup(C)* is used to get the permutation group of the linear code C. When such a group has been constructed, a search through its elements can be made in order to find involutions (of order 2) and degree n in case of even length, or degree $n - 1$ in case of odd length. It might be hard to search through all permutations if the size of the permutation group is large, and sometimes the search may take a long time and end with no involution found. In fact the Magma command *ConjugacyClasses(PGC)*, constructs a set of representatives for the conjugacy classes of the permutation group *PGC*. The classes are returned as a sequence of triples containing the element order, the class length and a representative element for the class. This command makes it easier to find out whether there is an involution or not, which saves time for very large permutation groups. When an involution is found, Lemma 1 is applied to obtain the equivalent code that has the reverse permutation in its permutation group. The set of codewords that satisfy the reverse constraint together with the Hamming distance constraint is extracted by omitting the codewords that are unchanged by the reverse permutation (C_0) by applying the following proposition.

Proposition 2 *[16] If C is a code that is fixed by the reverse permutation R, then the subcode C_0 of C consisting of the codewords that are unchanged by R is obtained as the intersection of C and the code C_R which contains all possible reversible codewords .*

Given the code C (a code satisfying the reverse constraint and the Hamming distance constraint), it can be turned into a code satisfying the reverse-complement and Hamming distance constraints by applying the following lemma.

Lemma 2 *[16] Let n be even. For the code $\mathcal{C}_1$ replace each of the first $n/2$ coordinates by its Watson-Crick complement ($A \leftrightarrow T$, $C \leftrightarrow G$). The code $\mathcal{C}_3$ obtained satisfies the reverse-complement constraint and the Hamming distance constraint for the value of d prescribed for $\mathcal{C}'$.*

Note that the GC-content of codewords is unaffected by this operation on the code. For odd n the situation is slightly more complicated, as the operation in Lemma 1 can reduce the Hamming distance between a codeword of $\mathcal{C}_1$ and the reverse complement of a codeword of $\mathcal{C}_1$ by 1. Given the code $\mathcal{C}_1$ (which satisfies the reverse constraint and the Hamming distance constraint), it can be turned into a code $\mathcal{C}_3$ satisfying the reverse-complement constraint and the Hamming distance constraint by applying the following lemma.

Lemma 3 *[16] Let $n = 2k + 1$ be odd. For the code $\mathcal{C}_1$ replace each of the first $\lfloor n/2 \rfloor$ coordinates by its Watson-Crick complement ($A \leftrightarrow T$, $C \leftrightarrow G$). The code $\mathcal{C}_3$ obtained consists of four subcodes $\mathcal{C}_A$, $\mathcal{C}_C$, $\mathcal{C}_G$, $\mathcal{C}_T$ in which the coordinate $k + 1$ is A,C,G and T respectively. Then the two subcodes $\mathcal{C}_3 = \mathcal{C}_A \cup \mathcal{C}_C$ or $\mathcal{C}_4 = \mathcal{C}_G \cup \mathcal{C}_T$ both satisfy the reverse-complement constraint and the Hamming distance constraint for the value of d prescribed for $\mathcal{C}'$.*

As $|\mathcal{C}_A| + |\mathcal{C}_C| + |\mathcal{C}_G| + |\mathcal{C}_T| = |\mathcal{C}_1|$, one of the two codes has at least half as many codewords as $\mathcal{C}_1$. If coordinate $k + 1$ takes a constant value then one of the codes $\mathcal{C}_3$ or $\mathcal{C}_4$ has the same number of codewords as $\mathcal{C}_1$.

The operations in the lemmas can be applied to any linear or additive codes. Given the code $\mathcal{C}_3$ the GC-weight enumerator can be calculated. The largest set of words of constant GC-weight then gives a code which is a candidate for $\max_w(A_4^{GC,RC}(n, d, w))$. Different involutions can give different sizes for this set of words, either because of different sizes for $|\mathcal{C}_0|$, or from the nature of the construction when n is odd.

5.1.1 Cyclic and extended cyclic codes construction

In the case of cyclic (or extended cyclic) codes, ideally all generator polynomials should be examined (as in Chapter 4) as well as all possible involutions (which are easily generated by Magma). The methods just described are then applied to give a cyclic, (or extended cyclic) code satisfying the Hamming distance constraint and the reverse-complement constraint. Then the largest set of codewords of fixed GC-content is selected as a candidate for $\max_w(A_4^{GC,RC}(n, d, w))$. This was done for $n \leq 18$. For $19 \leq n \leq 30$ this did not prove feasible, and three restricted searches were used:

- All polynomials were generated as in Chapter 4 but the involution search was truncated after a fixed period of time and the best result obtained in this period was used.
- The polynomial that gave the best result for $\max_w(A_4^{GC}(n, d, w))$ in Chapter 4 was used, together with a search of all possible involutions. This was possible when the permutation group of the code was small enough.
- The polynomial that gave the best result for $\max_w(A_4^{GC}(n, d, w))$ in Chapter 4 was also used when the permutation group was too large for a complete search. The Magma command to generate representatives of con-

jugacy classes was used. Either a suitable involution was found, or its nonexistence was shown (in which case no candidate code was generated).

In the case of additive cyclic and additive extended cyclic codes, ideally all sets of three polynomials $p(x)$, $q(x)$, $r(x)$ should be examined (as in Chapter 4), as well as all possible involutions (if any suitable involutions exist). The methods described in 5.1 can then be applied. This only proved feasible for $n \leq 10$ and in some cases with $11 \leq n \leq 20$, when the permutation group is small. For larger permutation groups just a small number of involutions were selected. In other cases with $11 \leq n \leq 20$ the set of polynomials $\{p(x), q(x), r(x)\}$ that gave the best result for $\max_w(A_4^{GC}(n, d, w))$ in Chapter 4 was used. Then a search of all possible involutions was carried out when the permutation group of the code was small enough; otherwise the Magma command to generate representatives of conjugacy classes can be used, and either a suitable involution can be found, or its nonexistence shown.

5.2 Cosets

A coset of a cyclic or extended cyclic code satisfying the Hamming distance and reverse-complement constraints may have more codewords of fixed GC-content than the code itself. In order for the reverse-complement constraint to be satisfied the coset leader selected must be reversible (fixed by the reverse permutation R). The code $\mathcal{C}_1$ is then replaced by the coset given by the selected coset leader, before Lemma 2 or Lemma 3 is applied. The coset then satisfies the Hamming distance and reverse-complement constraints and a GC-weight enumerator can be computed as before.

It only proved feasible to consider all generator polynomials and all involutions for $n \leq 18$. In these cases between 30 and 40 random reversible coset leaders were selected (using the method in [16] to ensure the coset leaders are reversible). For $19 \leq n \leq 30$ the polynomial that gave the best result for $\max_w(A_4^{GC}(n, d, w))$ in Chapter 4 was used. For this single polynomial there were two methods that could be applied:

- For smaller permutation groups all involutions were considered, together with between 5 and 10 random reversible coset leaders.
- For larger permutation groups the Magma command to generate representatives of conjugacy classes was used. Either a suitable single involution was found, or its nonexistence was shown. For this involution 10 random reversible cosets were considered.

The method for cosets of additive cyclic and additive extended cyclic codes, follows that for cosets of a cyclic or extended cyclic code, this time with 20 randomly selected reversible coset leaders. Otherwise the searches are as for the code itself.

5.3 Shortening and puncturing

Shortening and puncturing must be done in a way which ensures that a reverse-complement constraint is satisfied. If *any* position is used for shortening or

puncturing then there may be no fixed point free involution (n even) or one-point fixed involution (n odd) in the permutation group of the code obtained. Long chains of shortened or punctured codes are not obtained and the benefits of a particularly good code are not inherited by shorter codes. A better approach is to proceed as follows. For a code $\mathcal{C}_3$ of even length, shorten or puncture in positions i and $n-i+1$ ($i \in \{1, 2, \ldots, n/2\}$). The pair $(i, n-i+1)$ is lost from the fixed point free involution of $\mathcal{C}_3$, but as the length of the code reduces by two a fixed point free involution remains. For a code $\mathcal{C}_3$ of odd length there are two options. Either shorten or puncture once in position $\lfloor (n+1)/2 \rfloor$ or shorten or puncture twice in positions i and $n-i+1$ ($i \in \{1, 2, \ldots, \lfloor (n-1)/2 \rfloor\}$. In both cases a suitable involution remains.

As was noted in 3.4, a nonlinear shortening operation sometimes gives more codewords. Given a code $\mathcal{C}$ of constant GC-content over $\{A, C, G, T\}$, compute the frequency of each letter A, C, G, T in each column i of the matrix of codewords. Choose the letter and column of the most frequent occurrence and select all codewords with the chosen letter in the chosen column. Delete this component from all selected codewords and a (normally nonlinear) code of constant GC-content is obtained. As the operation in nonlinear, it is only feasible for shorter codes. Again, as for linear shortening, in order to preserve the reverse-complement constraint it is necessary to select position $n/2$ (n odd) or to apply the operation twice in the positions i and $n-i+1$ ($i \in \{1, 2, \ldots, \lfloor (n-1)/2 \rfloor\}$. If the operation is applied twice, the same letter must be selected in both positions.

Normally shortening gives an unchanged Hamming distance, and puncturing reduces the minimum distance by 1. Sometimes, however, the minimum distance is greater than is anticipated. Thus it is necessary to assess all possible shortenings and puncturing of a given code using the options above to find the best codes.

5.4 Results

Tables 10.33 to 10.48 in Appendix 2 gives the best lower bounds for $A_4^{GC,RC}(n, d, w)$ for cyclic and extended cyclic codes over $GF(4)$, and for their cosets. They also give the best lower bounds for $A_4^{GC,RC}(n, d, w)$ for cyclic additive and extended cyclic additive codes over $GF(4)$ and for their cosets. These codes satisfy the Hamming distance constraint, GC-content constraint and a reverse complement constraint. It can be seen that this calculation of codes has produced many new bests, i.e. codes which are larger than the best codes currently known. As can be seen in Tables 10.33, 10.34, 10.35 and 10.36, the calculation of codes obtained from cyclic and extended cyclic codes over GF(4) has produced two new best cyclic codes and one new best extended cyclic code with $n \leq 20$. Tables 10.37, 10.38, 10.39, 10.40, give lower bounds for $A_4^{GC,RC}(n, d, w)$ for the best coset of a cyclic code and coset of an extended cyclic code over $GF(4)$. This calculation of codes obtained from cosets of cyclic codes over GF(4) has produced two new best codes and two other new best codes are obtained from cosets of extended cyclic codes over GF(4) with $n \leq 20$. In Tables 10.41, 10.42, the calculation of codes obtained from cyclic additive codes over GF(4) has produced good results with $n \leq 20$ for both mappings. There are 14 new best codes when pairing 0 with 1 to give the GC-weight and 9 new best codes when pairing 0 with ω or ω^2 to give the GC-weight. In Tables 10.43, 10.44, it can be seen that just five

new best codes with $n \leq 20$ are obtained in both cases of codes obtained from extended cyclic additive codes over GF(4). The calculation of codes obtained from cosets of cyclic additive codes over GF(4) produces 15 new best codes with $n \leq 20$ in the case of pairing 0 with 1 (see Table 10.45), and 10 new best codes with $n \leq 20$ in the case of pairing 1 with ω, (see Table 10.46). The calculation of codes obtained from cosets of extended cyclic additive codes over GF(4) has 2 new best codes (pairing 0 with 1) and 5 new best codes (pairing 1 with ω) with $n \leq 20$ as shown in Tables 10.47, 10.48. Tables 10.49 and 10.50 summarise the results for best codes with minimum Hamming distance d and constant GC content for all computations. The results in this table are improved after applying the techniques of shortening and puncturing, with 32 new bests with $n \leq 20$ and 125 codes with $21 \leq n \leq 30$. This is shown in Tables 10.51, 10.52. Cases with $n \geq 20$ have not been computed before and so are all new bests.

The work described in this chapter has been submitted for publication [1].

Chapter 6

Codes with an all 1s vector in the dual

6.1 Codes with an all 1s vector in the dual satisfying a Hamming distance constraint and with constant GC-content

The goal of this section is to search for codes that have an all-ones vector in their dual, aiming to find as many codewords as possible with fixed GC-weight [16]. In this case the code has only even GC-weights. This means that the set of all codewords is spread over a smaller number of possible weights. Thus, a larger number of codewords for a given choice of GC-weight is likely.

Proposition 3 *[16] Let C be a code over $GF(4)$. If the all-ones vector belongs to $C^{\perp}$, then the GC-weight enumerator of C is even (has all even weights).*

To construct such codes, start from best known codes over GF(4) after shortening and puncturing (Tables 10.25 and 10.26). To search for codes that have the all-ones vector in their dual, the following proposition is applied.

Proposition 4 *[16] Let C be a linear code over $GF(4)$ of length n. Suppose $C^{\perp}$ has a vector $c = (c_1, \ldots, c_n)$ of weight n. Then C is equivalent to a code that has the all-one vector in its dual.*

The equivalent code is obtained by selecting the codewords in $C^{\perp}$ that have weight n, using the Magma command $Words(Dual(C), n)$ which gives the set of codewords in the dual code $C^{\perp}$ of weight n. Notice that all vectors of length n in $C^{\perp}$ are considered, since different codewords may lead to different equivalent codes which may lead to different GC-weight enumerators. The inner product of these vectors and any codeword is zero. For every vector, each entry that is ω is replaced by 1 and the corresponding column of the generator matrix of the code C is multiplied by ω to get the same inner product. Similarly, each entry that is ω^2 is replaced by 1 and the corresponding column of the generator matrix of the code C is multiplied by ω^2 so the same inner product is obtained. Thus the equivalent code has the all ones vector in its dual and therefore has an

even GC-weight enumerator. After a GC-weight enumerator for the equivalent code is computed, the code of constant GC-weight can then be selected.

Notice that Proposition 3 is true in the case of pairing 0 with 1 for GC and pairing ω with ω^2 for AT. It is not true for all alternative pairings.

6.2 Results

Tables 10.59, 10.60 give lower bounds for $A_4^{GC}(n, d, w)$ for the codes with an all 1s vector in the dual over GF(4). The codes which are better than the best codes previously constructed are recorded. As can be seen, just one new best have been found for $n \leq 20$. All of the entries for $n > 20$ are new bests.

6.3 Codes with an all 1s vector in the dual satisfying a Hamming distance constraint, with constant GC-content and satisfying a reverse-complement constraint

The idea is to start from best known codes over GF(4) after shortening and puncturing (Tables 10.25 and 10.26), searching for codes whose dual contains the all-one vector and a fixed point free involution or a one point-fixed involution in their permutation group. Unfortunately no involution was found for any of these cases, so no new results were obtained.

Chapter 7

Using the best known linear codes

7.1 Best known linear codes satisfying a Hamming distance constraint and with constant GC-content

A linear code C is said to be a best known linear code if C has the highest dimension among all known linear codes of a given minimum weight. In fact, Magma has a database for best known linear codes over GF(4), which contains constructions of best codes of length up to 100. Many of the codes constructed in this database are improvements on the previously known bounds for best codes over GF(4).

The command $BestDimensionLinearCode(GF(4), n, d)$ is used to return a linear code over GF(4) with length n and minimum weight d which has the largest dimension among known codes. By using the command *SetVerbose(("BestCode"),true)* the verbose flag *BestCode* is set to true. The method by which the best code in the database is constructed is then printed. When the code is generated, its complete weight enumerator is computed and the GC-weight enumerator is determined. The techniques of shortening were repeatedly applied to a best linear code in order to find from it good shorter codes.

7.2 Results

Tables 10.61, 10.62 give lower bounds for $A_4^{GC}(n, d, w)$ for the best known linear codes over GF(4) or for shortening of these codes. Only the codes which are better than the best codes previously constructed are recorded. As can be seen, no new bests have been found for $n \leq 20$. All of the entries for $n > 20$ are new bests. The calculation of codes obtained from shortening the best known linear codes over GF(4) has not produced any new best codes. However, there are some improvement compared with the entry in the table of best lower bounds for $A_4^{GC}(n, d, w)$.

7.3 Best known linear codes satisfying a Hamming distance constraint, with constant GC-content and satisfying a reverse-complement constraint

For every best known linear code, the permutation group of the code is computed and a search is made for a fixed-point free involution (or one point fixed involution). By applying Lemma 1, and the other results of Chapter 5, codes are obtained with constant GC-content satisfying a Hamming distance constraint and a reverse-complement constraint.

7.4 Results

Tables 10.63, 10.64 give lower bounds for $A_4^{GC,RC}(n,d,w)$ for the best known linear codes over GF(4) or for shortening of these codes. Only the codes which are better than the best codes previously constructed are recorded. As can be seen, no new bests have been found for $n \leq 20$. All of the entries for $n > 20$ are new bests. The calculation of codes obtained from shortening the best known linear codes over GF(4) has not produced any new best codes with $n \leq 20$. However, there are some improvements compared with entries in Tables 10.51, 10.52 and some cases where an involution is found where none was found before.

Chapter 8

Conclusion

The thesis aimed to design DNA strand sets that are suitable for DNA computing and for other applications involving synthetic DNA. Techniques from coding theory were used to construct codes over an alphabet {A,C,G,T} relevant to the design of synthetic DNA strands used in DNA microarrays, as DNA tags in chemical libraries and in DNA computing. Previous approaches to the construction of these codes were extended in several ways. It was shown that in some cases codes are obtained which are larger than the best codes currently known with same minimum Hamming distance. Codes were obtained with length greater than twenty. Linear DNA codes of such lengths have not been constructed previously, so all are the best known results. The software system Magma was used for the implementation.

Firstly, codes which satisfy a Hamming distance constraint (a step towards avoiding unwanted hybridizations) and a GC-content constraint (which ensures uniform melting temperatures) were considered. In particular linear and additive cyclic and extended cyclic codes over $GF(4)$, together with linear cyclic and extended cyclic codes over Z_4 have been computed. The comprehensive approach presented here using cyclic and extended cyclic codes found many new best codes with $n \leq 20$ and extended known results to $n \leq 30$. Further improvements might be available if it became feasible to extend the computations for additive codes to $21 \leq n \leq 30$. Tables 10.25 and 10.26, summarise the results for best codes with minimum Hamming distance d and constant GC content for all computations. Those tables which give best lower bounds for $A_4^{GC}(n, d, w)$, are improved after applying the shortening and puncturing codes techniques which were applied to the best linear codes more than once. A nonlinear shortening operation sometimes gave more codewords. Normally shortening gives an unchanged Hamming distance, and puncturing reduces the minimum distance by 1. Sometimes, however, the minimum distance is greater than is anticipated. Thus all possible shortenings and puncturing of a given code in the table are assessed to find the best codes with length reduced by 1. The results obtained after shortening and puncturing are given in Tables 10.27 and 10.28, the polynomials used to construct the codes with subscripts cf and ef are given in Table 10.29 to 10.32.

Secondly, this comprehensive approach was applied to codes which also sat-

isfy the reverse complement constraint (to further prevent unwanted hybridizations.). The codes were derived from linear codes over $GF(4)$, additive codes over $GF(4)$, and their cosets. This constraint was studied using involutions in a permutation group, as had been done previously by Gaborit and King [16]. Detailed attention to the option for constructing DNA codes from cyclic and extended cyclic codes, in combination with the use of involutions pioneered by Gaborit and King, produced many new best codes with $n \leq 20$. Further codes were obtained with length greater than twenty. Linear DNA codes of such lengths had not been constructed previously, so all are the best known results. Tables 10.49 and 10.50 summarise the results for best codes satisfying a minimum Hamming distance constraint, a fixed GC-content constraint and a reverse complement constraint for all computations. Shortening and puncturing the best lower codes for $A_4^{GC,RC}(n, d, w)$, produced improvements to the results. A nonlinear shortening operation sometimes gave more codewords. Normally shortening gives an unchanged Hamming distance, and puncturing reduces the minimum distance by 1. Sometimes, however, the minimum distance is greater than is anticipated. Thus shortenings and puncturings of codes with two consecutive values of d (one position) or three consecutive values of d (two positions) were assessed. The results obtained after shortening and puncturing are given in Tables 10.51 and 10.52; the generator polynomials and the coset leaders for cosets of linear codes are given in Tables 10.53 to 10.58.

Thirdly, the database for best known linear codes over GF(4) in Magma was applied. Many of the codes constructed in this database are improvements on the previously known bounds for best codes over GF(4). Lower bounds for $A_4^{GC}(n, d, w)$ and $A_4^{GC,RC}(n, d, w)$ for the best known linear codes over GF(4) or for shortening of these codes are computed. Tables 10.61,10.62, 10.63 and 10.64, summarise the results for the best known linear codes $A_4^{GC}(n, d, w)$ and $A_4^{GC,RC}(n, d, w)$ respectively. Some improvement have been made compared with the entry in the tables of best lower bounds for $A_4^{GC}(n, d, w)$ and $A_4^{GC,RC}(n, d, w)$. Linear DNA codes of lengths > 20 have not been constructed previously, so all are the best known results.

Fourthly, searching for linear codes that have an all ones vector in the dual as Gaborit and King [16] recommended was carried out. The approach has generated some improvements to the results in Tables 10.27, 10.28 when finding lower bounds for $A_4^{GC}(n, d, w)$. However, when the reverse-complement constraint was added, it was found that either the dual code had no words with an all 1s vector, or the permutation group of the equivalent code had no fixed point free involution (n even) or no one-point fixed involution (n odd). Thus no improvements to $A_4^{GC,RC}(n, d, w)$ were obtained.

Tables 10.65 and 10.66 give the summary of best codes satisfying a Hamming distance constraint and a GC-content constraint ($A_4^{GC}(n, d, w)$), whereas Tables 10.67 and 10.68 give the summary of best codes satisfying a Hamming distance constraint, a GC-content and a reverse complement constraint. Entries with subscript a are best codes obtained after applying the techniques of shortening and puncturing, entries with subscript b are the best codes obtained from best linear codes, and entries with subscript c are best codes obtained from codes

with an all 1s vector in the dual. Entries that improve results in the online table of Gaborit and King[1] for $n \leq 20$ are marked in bold; entries that equal these results are marked in italic. Note that there are five entries in the table for $A_4^{GC}(n, d, w)$ in [17] which correspond to bold entries in Table 10.65 and have been updated since this part of the work was completed. Two of these are results from [31] or [32] which improve the result given here and three of them are results from [31] or [32] which equal the result given here. For $n \geq 21$ the entries are all new bests and are not marked. Files of codewords for these codes (when the code is best known and the number of codewords does not exceed 50000) are maintained at two web sites[2]. The polynomials and coset leaders for cosets of linear codes, used to construct codes satisfying a Hamming distance constraint and a GC-content constraint are presented in Tables 10.30 to 10.32. Cases with fewer than 4 codewords are omitted.

Finally, the best known lower bounds for $\max_w(A_4^{GC}(n, d, w))$ are reported in Tables 10.69 and 10.70 and the best known lower bounds for $\max_w(A_4^{GC,RC}(n, d, w))$ are reported in Tables 10.71 and 10.72. Entries with the subscript gk are taken from [16], [23] or the authors' updated online table[3], where details of the individual constructions used can be found. Entries with the subscript $m1$ are variable neighbourhood search results taken from [31] and entries with the subscript $m2$ are variable neighbourhood search results or simulated annealing results taken from [32]. Entries with the subscript $m3$ are results taken from [33]. Entries with subscripts $L1$ and $L2$ are taken from best codes generated in Chapter 7.

In conclusion, it can be observed that the use of linear cyclic and extended cyclic codes (and codes derived from them) is very successful. The comprehensive approach presented here using cyclic and extended cyclic codes has found many new best codes with $n \leq 20$ and extended known results to $n \leq 30$. It has been shown that additive codes often have more codewords satisfying the constraints than linear codes with the same minimum distance. This is possibly because there are more additive codes than linear codes for the same code length. Further improvements might be available if it became feasible to extend the computations for additive codes to $21 \leq n \leq 30$. Although a cyclic code over Z_4 with constant GC-content and minimum distance d sometimes has more codewords than any linear cyclic code over $GF(4)$ with constant GC-content and minimum distance d, no code over Z_4 appears as the best of all candidates in the final table after all constructions are considered. Some do give lower bounds that equal the bound in the table, but a $GF(4)$ construction is then always referenced. Methods based on linear codes and algorithmic methods are complementary; neither dominates the other. Algorithmic methods generally work better when d is close to n, linear code methods are better (or essential) when d is much smaller than n. On the other hand, it has been shown that the computer algebra system Magma saves both time and effort in developing the software, and is very convenient for the construction of linear codes. It has many built in facilities which aid the construction of linear codes. In terms of

[1] http://llama.med.harvard.edu/ king/dnacodes.html
[2] http://data.research.glam.ac.uk/projects/ ; http://www.idsia.ch/∼roberto/DNA10.zip ;
[3] http://llama.med.harvard.edu/∼king/dnacodes.html

running time, it usually took a few seconds for small n up to a few minutes to a few weeks for $n \geq 20$, In the case of linear additive codes it was not possible to construct codes with $n \geq 20$. The computation would have taken months to complete.

It might be possible to generate further improvements by a detailed study of linear codes in the literature that are not equivalent to a cyclic code or an extended cyclic code. It would also be possible to consider linear codes with random generator matrices. Finally, it would be possible to undertake a detailed study of non-linear codes in the literature. It seems certain that the computational effort required for these last two options would be very large.

Bibliography

[1] 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*, submitted for publication.

[2] Adleman, L.M. *Molecular computation of solutions to combinatorial problems*. Science. 266(1994) 1021-1024.

[3] Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z., Shapiro, E. *DNA molecule provides a computing machine with both data and fuel*. Proceedings of the National Academies of Sciences. 100(2003) 2191-2196.

[4] Bosma, W., Cannon, J. *Discovering mathematics with Magma*, Algorithms and Computation in Mathematics, vol. 19, Springer-Verlag, Berlin, (2006).

[5] Brenner, S. *Methods for sorting polynucleotides using oligonucleotide tags*. US Patent Number. 5604097, (1997).

[6] Brenner, S., Lerner, R.A. *Encoded combinatorial chemistry*. Proc. Natl. Acad. Sci. U. S. A. 89(1992) 5381-5383.

[7] Brouwer, A. B. *Bounds on the minimum distance of linear codes*. URL:http://www.win.tue.nl/ aeb/voorlincod.html. [Accessed 15 November 2010].

[8] Calderbank, A.R., Rains, E.M., Shor, P.W., Sloane, N.J.A. *Quantum error correction via codes over GF(4)*, IEEE Trans. Inform. Theory 44(1998) 1369-1387.

[9] Calderbank, A.R., Sloane, N.J.A. *Modular and p-adic cyclic codes*. Designs, Codes and Cryptography. 6(1995) 21-35.

[10] Cannon, J., Bosma, W. (eds.) *Handbook of Magma Functions*, version 2.13. The University of Sydney. (2006).

[11] Cannon, J. J., Souvignier, B. *On the computation of conjugacy classes in permutation groups*. In Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, pages 392-399. Association for Computing Machinery, 1997. Maui, July 21-23, (1997).

[12] Deaton, R., Garzon, M., Murphy, R.C., Rose, J.A., Franceschetti, D.R., Stevens Jr, S.E. *Genetic search of reliable encodings for DNA-based computation*. Proc. 1st Ann. Conf. Genetic Programming (1996).

[13] Deaton, R., Murphy, R.C., Garzon, M., Franceschetti, D.R., and Stevens Jr., S.E. *Good encodings for DNA based solutions to combinatorial problems*. Proc. DNA Based Computers II, DIMACS Workshop, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 44(1996) 247-258.

[14] Fodor, S., Read, J., Pirrung, M., Stryer, L., Lu, A., Solas, D. *Light-directed, spatially addressable parallel chemical synthesis*. Science 251(1991) 767-773.

[15] Frutos, A.G., Liu, Q., Thiel, A.J., Sanner, A.M.W., Condon, A.E., Smith, L.M., Corn, R.M. *Demonstration of a word design strategy for DNA computing on surfaces*, Nucleic Acids Res. 25(1997) 4748-4757.

[16] Gaborit, P., King, O.D. *Linear constructions for DNA codes*. Theoretical Computer Science. 334(2005) 99-113.

[17] Gaborit, P., King, O.D. *Tables of lower bounds for DNA codes with constant GC-content*, Available at: URL: <http://llama.mshri.on.ca/∼king/dnacodes.html> [accessed 15 November 2008]

[18] Gathen, J., Gerhard, J. *Modern Computer Algebra*. Cambridge University Press, Cambridge, (1999).

[19] Grassl, M. *Code tables: Bounds on the parameters of various types of codes*, Available at: URL: <http://www.codetables.de/> [accessed 06 Sept. 2010].

[20] Heck, A. *Introduction to Maple*. Second Edition. New York: Springer-Verlag (1997).

[21] Heal, K.M., Hansen, M.L., Rickard, K.M. *Maple V Learning Guide*. New York: Springer-Verlag (1998).

[22] Holt, D,. KBMAG *Knuth-Bendix in Monoids and Automatic Groups*. University of Warwick, (1997).

[23] King, O.D. *Bounds for DNA codes with constant GC-content*. Electron. J. Comb. 10, R33 (2003).

[24] Klima, R.E. *Application of Abstract Algebra with Maple*. Boca Racon, Florida: CRC Press. (1999).

[25] Knuth, D. E. *The Art of Computer Programming*, volume 2. Addison Wesley, Reading, Massachusetts, 3rd edition, (1997).

[26] Lipton, R. J. *DNA solution of hard computational problems*. Science, 268(1995) 542-545.

[27] Li, M., Lee, H.J., Condon, A.E., Corn, R.M. *DNA word design strategy for creating sets of non-interacting oligonucleotides for DNA microarrays*. Langmuir. 18(2002) 805-812.

[28] MacWilliams, F.J., Sloane, N.J.A. *The Theory of Error-Correcting Codes*. Amsterdam, Netherlands: North-Holland. (1997).

[29] Maplesoft. *Maple Product History* [online]. Canada: Maplesoft. Available at: URL: <http://www.maplesoft.com/company/about/index.aspx> [Accessed 5 January 2008].

[30] Marathe, A., Condon, A.E. Corn, R.M. *On combinatorial DNA word design*, J. Comput. Biol. 8(2001) 201-219.

[31] Montemanni, R., Smith, D.H. *Construction of constant GC-content DNA codes via a variable neighbourhood search algorithm*, J. Math. Model. Algorithms 7(2008) 311-326.

[32] Montemanni, R,. Smith, D. H. *Metaheuristics for the construction of constant GC-content DNA codes*, MIC 2009: The VIII Metaheuristics International Conference, Hamburg, July 13 - 16, 2009.

[33] Montemanni, R,. Smith, D. H,. Koul, N. *Three metaheuristics for the construction of constant GC-content DNA codes*, in Post-proceedings of the VIII Metaheuristic International Conference, Chapter 14, (eds. S. Voss and M. Caserta), Springer, to appear.

[34] Rykov,V., Macula, A. J., Torney, D., White, P., *DNA sequences and quaternary cyclic codes*, IEEE Isit, Washington, pp. (2001) 248-248.

[35] Shoemaker, D., Lashkari, D.A., Morris, D., Mittmann, M., Davis, R.W. *Quantitative phenotypic analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy*. Nat. Genet. 14(1996) 450-456.

[36] Slonim, D. K., Tamayo, P., Mesirov, J. P., Golup, T. R., Lander, E. S. *Class prediction and discovery using gene expression data*. In Proceedings of the Fourth Annual International Conference on Research in Computational Molecular Biology(RECOMB), 263-272, 2000.

[37] Smith, D. H., Aboluion, N., Montemanni, R., Perkins, S. *Linear and nonlinear constructions of DNA codes with Hamming distance d and constant GC-content*, Discrete Mathematics, in press: http://dx.doi.org/10.1016/j.disc.2010.03.005

[38] Tulpan, D. C. *Effective Heuristic Methods for DNA Strand Design*. Ph.D. thesis. The University of British Columbia. (2006).

[39] Tulpan, D. C., Hoos, H. H., Condon, A. E. *Stochastic local search algorithms for DNA word design*. In DNA Computing: 8th International Workshop on DNA-Based Computers (editors M. Hagiya and A. Ohuchi), Springer LNCS. 2568(2003) 229-241.

[40] Tulpan, D. C., Hoos, H. H. *Hybrid randomised neighbourhoods improve stochastic local search for DNA Code design*. In Advances in Artificial Intelligence: 16th Conference of the Canadian Society for Computational Studies of Intelligence (editors Y. Xiang and B. Chaib-draa), Springer LNCS vol. 2671(2003) 418-433.

[41] The University of Sydney. *MAGMA Computational Algebra System* [online]. Sydney: Computer Algebra Group. Available at: URL: <http://magma.maths.usyd.edu.au/magma/> [Accessed 4 November 2008].

[42] University of Waterloo. *History of Maple* [online]. Canada. Available at: URL:<http://www.scg.uwaterloo.ca/history.shtml> [Accessed 5 January 2008].

[43] Van Lint, J. H. *Coding theory*. Berlin, Heidelberg, New York. Springer-Verlag. (1970).

Appendix I

The construction of binary codes using the software packages Maple and Magma

This appendix demonstrates the advantages of Magma over Maple for this work.

Constructing linear codes in Maple from a generator matrix

This section shows how Maple can be used to construct a linear code from its generator matrix.

Suppose that the following generator matrix is given:

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

To construct the codes using Maple, the Maple linear algebra package *linalg* should be included at the beginning. The generator matrix G used to construct the code is then entered:

```
with(linalg):
G:=array(1..4,1..15,[[0,1,1,1,0,0,1,0,1,0,0,0,1,1,1],
[1,0,1,1,0,1,1,1,0,0,0,1,0,1,0],[1,1,0,1,1,1,0,0,1,0,1,0,0,1,0],
[1,1,1,0,1,0,0,1,0,1,0,0,0,1,1]]);
```

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

It is clear that if G is a generator matrix of the code C and given a vector $u = (u_1, u_2, \ldots, u_k)$ where $u \in F_2^k$ then $v = uG$ is a codeword in C. Conversely, any $v \in C$ can be written as $v = uG$ where $u = (u_1, \ldots, u_k)$, $u \in F_2^k$.

The construction of the linear code can be started by first building the vector space F_2^k. Create an array A whose rows are the binary representations for the numbers $0, 1, 2, \ldots, 2^m - 1$, where m is the number of columns of G. The array itself is obtained by using the command:

```
A:=array(1..2^(rowdim(G))-1,1..rowdim(G)):
```

Next, by using the *convert* command, the binary representations of the numbers $0, 1, 2, \ldots, 2^m - 1$ can be obtained and to ensure that every vector is converted to a binary vector of length m, the array *bv* is defined. All its element are initially zero, then by using a *for loop* command, each vector *cb* becomes the array *bv* in turn. These vectors form the rows of A.

```
for i from 1 to 2^(rowdim(G)) do
cb:=convert (i-1,base,2);
d:=vectdim(cb);
```

```
bv:=array(1..rowdim(G)):
for k from 1 to rowdim(G) do
  bv[k]:=0:
od:
for j from 1 to d do
bv[rowdim(G)-j+1]:=cb[j]:
od:
for w from 1 to rowdim(G) do
A[i,w]:=bv[w]:
od:
od:
A:=eval(A);
```

Finally, by multiplying A by G, and using a *map* command, the codewords are reduced mod 2 and a matrix in which the rows are the codewords of the linear code is obtained.

```
C:=multiply(A, G ):
Code:=map( irem, C, 2);
```

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

In the above example the generator matrix can be stored in a text file:

```
fd := fopen("niemadata7.txt", WRITE):
for i1 from 1 to m do
  for i2 from 1 to 2^m-1 do
    fprintf(fd,"%1d ",G[i1,i2]):
  od:
  fprintf(fd,"\n"):
od:
fclose(fd);
```

It can be read in again from the text file into Maple if required.

```
G:=array(1..m,1..2^m-1):
fd := fopen("niemadata6.txt", READ):
```

```
G:=readdata(fd,[integer,integer,integer,integer,integer,integer,integer,integer,
integer,integer,integer,integer,integer,integer,integer]);
```

Constructing Hamming codes in Maple

In this section construction of Hamming codes using Maple is illustrated.

Firstly, the parity check matrix H of the Hamming code is constructed column by column. These columns are binary representations of the numbers $1, 2, \ldots, 2^m - 1$, where m is the number of rows of H; this is obtained as shown before. For example, in the case $m = 3$ there are seven possible columns constructed as follows:

```
with(linalg):
m:=3;
H:=array(1..m,1..2^m-1):
for j from 1 to (2^m)-1 do
cb:=convert(j,base,2);
d:=vectdim(cb);
bv:=array(1..m):
for k from 1 to m do
  bv[k]:=0:
od:
for i from 1 to d do
bv[m-i+1]:=cb[i]:
od:
for l from 1 to m do
H[l,j]:=bv[l]:
od:
od:
H:=evalm(H);
```

As a result the parity check matrix of $[7, 4]$ Hamming code is obtained.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Next a generator matrix G for the Hamming code is constructed. To do this, the Maple *Nullspace* command is used. This command gives a set of rows which form a basis for the nullspace of H over the binary field. By using *for loop* and *stackmatrix* commands, the set of rows are arranged in a matrix G which forms the generator matrix of the Hamming code.

```
GG:=Nullspace(H) mod 2 ;
G:=[ ]:
for b from 1 to (2^(m)-1)-m do
G:=stackmatrix(op(G),GG[b]);
od:
G:=evalm(G);
```

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Finally, to obtain the Hamming code from its generator matrix, the same process is used as shown in constructing linear codes in Maple from a generator matrix.

Constructing Reed-Muller codes in Maple

The construction of *Reed-Muller codes* using Maple is illustrated in this section. Begin by entering the values of m and r; after that an array is created for the values of V. The rows of the array are the binary representations of length m for the numbers $0, 1, \ldots, 2^m - 1$. These become headings $(v_1, v_2, \ldots, v_m)$ for the columns of the generator matrix of the code $R(r, m)$.

```
restart:
##Reed_Muller code length 2^m Order r
with(linalg):
m:=4;r:=3;
##GENERATE COLUMN HEADINGS
colheadings:array(1..2^m,1..m):
for j from 0 to 2^m-1 do :
  cb:=convert(j,base,2):
  d:=vectdim(cb):
  bv:=array(1..m):
  for k from 1 to m do:
    bv[k]:=0:
  od:
  for i from 1 to d do:
    bv[m-i+1]:=cb[i]:
  od:
  for i from 1 to m do:
    colheadings[j+1,i]:=bv[i]:
  od:
od:
```

Next, the Boolean functions used to construct the generator matrix are constructed, using binary representations of length m for the numbers $1, 2, \ldots, 2^m$. A vector in a row in this array corresponds to one Boolean function. For example, the vector 0011 corresponds to the Boolean function $v_3 v_4$.

```
bools:array(1..2^m,1..m):bvfn:=array(1..m):
 numberbools:=0:
 for jfn from 1 to 2^m-1 do
   cbfn:=convert(jfn,base,2):
   dfn:=vectdim(cbfn):
   for kfn from 1 to m do:
   bvfn[kfn]:=0:
   od:
```

```
   for ifn from 1 to dfn do:
     bvfn[m-ifn+1]:=cbfn[ifn]:
   od:
   degreecount:=0:
   for i from 1 to m do
     if bvfn[i]=1 then degreecount:=degreecount+1: fi:
   od:
   if degreecount<=r then numberbools:=numberbools+1:
for i from 1 to m do
 bools[numberbools,i]:=bvfn[i]: od: fi:
od:
```

Finally, the generator matrix G is constructed. Every row in the matrix G represents the set of values of one of the Boolean functions, each value depending on the values of the Boolean variables in the column headings. For example, the third row of G will represent the values of the Boolean function v_3v_4. The way that each (non-zero) function is evaluated for the values in the column heading can be seen in lines 7–11 of the Maple code below:

```
G:=array(1..numberbools+1,1..2^m):
for w from 1 to 2^m do
G[1,w]:=1:
od:
for z from 1 to numberbools do
for q from 1 to 2^m do
K:=1;
for s from 1 to m do
if  bools[z,s] = 1 then
K:=K*(colheadings[q,s]);
fi:
 od:
G[z+1,q]:=K;
od;
od;
```

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The codewords of the Reed-Muller code can be constructed from the generator matrix as shown in constructing linear codes in Maple from a generator matrix.

Constructing Cyclic codes in Maple

This section shows how Maple can be used to construct cyclic codes.

Note that in a cyclic code, if n is odd the zeros of $x^n - 1$ are called "primitive n'th roots of unity" and lie in the "splitting field" GF(2^m). There are then n distinct roots and no repeated factors in this case. Full details can be found on page 196 of [28]. It follows that in the n even case the multiplicities of all the factors are even.

Suppose that all cyclic codes C of length $n = 14$ are to be constructed. Begin by including the *linalg* package and entering the polynomial $f(x) = x^{14} - 1 \in GF(2)[x]$ before factorising it by using the Maple *Factor* command. The factors and products of factors of $f(x)$ will give various possibilities for constructing cyclic codes.

```
with(linalg):
n:=14:
f:=x->x^n-1;
factf:=Factor(f(x))mod 2;
```

$$f := x \to x^{14} - 1$$

$$factf := (x+1)^2(x^3+x+1)^2(x^3+x^2+1)^2$$

The following Maple code is used to ensure that if $x^n - 1$ is a perfect square, then repeated factors are counted and extracted correctly.

```
nn:=n:
while nn=2*(floor(nn/2)) do
nn:=nn/2;
```

```
repeated:=repeated*2;
f:=x->x^nn-1:
factf:=Factor(f(x))mod 2;
od;
```

Next, the rows of a matrix A are used to get all possibilities for the generator polynomial of a cyclic code C. To do this, first an *if* statement is used to ensure that factors of $f(x) \neq x^i - 1$ for $i = 1 \ldots n$, i.e to exclude trivial factors. After that, subsequent loops are used to build the rows of A; these rows are the binary representations of length equal to number of factors of $f(x)$. To convert these rows, the process used in the construction of linear codes in Maple from a generator matrix is followed.

```
trivial:=false;
for i from 1 to n do
if factf = x^i+1 then trivial:=true; fi:
od:
if trivial=false then
numfacts:=nops(factf);
A:=array(1..(repeated+1)^(numfacts)-2,1..numfacts):
for u from 1 to (repeated+1)^(numfacts)-2 do
cb:=convert (u,base,(repeated+1));
d:=vectdim(cb);
bv:=array(1..numfacts):
for k from 1 to numfacts do
  bv[k]:=0:
od:
for i from 1 to d do
bv[numfacts-i+1]:=cb[i]:
od:
for w from 1 to numfacts do
A[u,w]:=bv[w]:
od:
od:
evalm(A);
```

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \\ 0 & 2 & 2 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 0 & 0 \\ 2 & 0 & 1 \\ 2 & 0 & 2 \\ 2 & 1 & 0 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

In order to construct all generator matrices corresponding to every generator polynomial, several commands are used, included further loops. Start by extracting factors of $f(x)$ using an *op* command, after that, a *for loop* is used to evaluate the generator polynomial which corresponds to each row in A. The *expand* command which appears in this loop is used to obtain the products of polynomials.

```
for z from 1 to numfacts do
f[z]:=op(z,factf);
od;
for j from 1 to (repeated+1)^(numfacts)-2 do
Fprod:=x^0:
for h from 1 to numfacts do
if A[j,h]>0 then
 Fprod:=expand((Fprod)*(f[h])^A[j,h])mod 2 ;
fi:
od:
print(Fprod);
```

The *if* statements which appear in the next *for loop* are used to examine whether the generator polynomial obtained in the last step is trivial or not. If it is not trivial then further *for loops* are used to build the generator matrix corresponding to each generator polynomial. Then the generator matrix is printed.

```
trivial:=false;
```

```
for i from 1 to n do
if Fprod = x^i+1 then trivial:=true; fi:
od:
if trivial=false then
deg:=degree(Fprod,x);
print(deg);
G:=array(1..n,1..n);
for l from 1 to n do
G[1,n+1-l]:=coeff((Fprod,x,l-1));
od:
for p from 1 to n-deg-1 do
si:=rem(expand((x^p)*(Fprod)),x^n-1,x);
print (si);
for v from 1 to n do
G[p+1,n+1-v]:=(coeff(si,x,v-1)):
od:
od:
for k1 from 1 to n-deg do
for k2 from 1 to n do
printf(%d,G[k1,k2]);
od;
printf("\n");
od:
else printf("No non trivial factors");
fi;
od;
else printf("No non trivial factors");
fi;
```

The Maple finite field package

Maple has a package *GF* for constructing finite (*Galois*) fields. It contains some commands used to construct the non zero elements in a finite field $GF(p^m)$. This section shows how this package is used to construct the finite field. Suppose the finite field $GF(2^4)$ is to be constructed. Firstly, the field $GF(2^4)$ needs to be defined and this can be done as follows:

```
p:=2:
r:=4:
G[p^r] := GF(p,r,alpha^r+alpha+1);
```

The command *GF* allows the arithmetic to be carried out over the Galois field $GF(2^4)$, the primitive irreducible polynomial used to construct the field is $\alpha^4 + \alpha + 1$. Note that this command only works for those r for which there is an irreducible polynomial of the form $\alpha^r + \alpha + 1$. For other r a suitable irreducible polynomial must be input instead. After that, elements from $GF(2^4)$ are converted to Maple sums of products using the *ConvertIn* command.

```
a := G[p^r]:-ConvertIn(alpha);
```

In order to construct and display all of the non zero elements in $GF(2^4)$ as powers of α, a Maple *for* loop is used.

```
for i from 1 to p^(r)-1 do
f[i]:=G[p^r]:-‘^‘(a,(i-1));
od;
f0:=G[p^r][‘-‘](a,a);
```

Finally, the *output* command can be used to convert the polynomials to corresponding integers in the range $0 \ldots 2^4 - 1$. Conversely, the *convert out* command converts the integers to the corresponding polynomials.

```
for j from 1 to p^(r)-1 do
integarf[j]:=G[p^r]:-output(f[j]):
ordinaryMaplePolynomialf[j]:=G[p^r]:-ConvertOut(f[j]):
od;
```

Constructing BCH codes in Maple

This section shows the construction of the *BCH* code using Maple. Begin by representing the codewords of the parity check matrix of the *BCH* code as polynomials in a finite field $GF(2^m)$ before expanding as codewords. To do this, first create arrays $H1$ and H with 2 rows and $2^m - 1$ rows respectively. The first one is used to store the polynomials and the second one to store the vectors corresponding to the polynomials. After that, the primitive polynomial $f(x) = \alpha^m + \alpha + 1 \in GF(2)$ is entered in order to construct the parity check matrix, as has been seen in the finite field section.

```
with(linalg):
p:=2:
r:=4:
H1:=array(1..p,1..p^(r)-1):
H:=array(1..2*r,1..p^(r)-1):
G[p^r] := GF(p,r,alpha^r+alpha+1);
a := G[p^r]:-ConvertIn(alpha);
for i from 1 to p^(r)-1 do
f1[i]:=G[p^r]:-‘^‘(a,(i-1));
f2[i]:=G[p^r]:-‘^‘(a,(3*(i-1)));
od;
f0:=G[p^r][‘-‘](a,a);
for j from 1 to p^(r)-1 do
integarf1[j]:=G[p^r]:-output(f1[j]):
print(integarf1[j]);
ordinaryMaplePolynomialf1[j]:=G[p^r]:-ConvertOut(f1[j]):
H1[1,j]:=ordinaryMaplePolynomialf1[j]:
cc:=convert(integarf1[j],base,2):
z:=vectdim(cc);
bv:=array(1..r):
for k from 1 to r do
  bv[k]:=0:
od;
for w from 1 to z do
bv[w]:=cc[w]:
od:
```

```
eval(bv);
integarf2[j]:=G[p^r]:-output(f2[j]):
ordinaryMaplePolynomialf2[j]:=G[p^r]:-ConvertOut(f2[j]);
H1[2,j]:=ordinaryMaplePolynomialf2[j]:
ccc:=convert(integarf2[j],base,2):
zz:=vectdim(ccc);
bvv:=array(1..r):
for kk from 1 to r do
  bvv[kk]:=0:
od;
for ww from 1 to zz do
bvv[ww]:=ccc[ww]:
od:
eval(bvv);
for q from 1 to r do
H[q,j]:=bv[q]:
H[q+r,j]:=bvv[q]:
od:
od;
eval(H1);
type(H1, 'array'(polynom));
H:=eval(H);
```

The command

```
f1[i]:=G[p^r]:-`^`(a,(i-1));
```

which appears in the first *for* loop, constructs the first row of the parity check matrix $H1$, whereas the command

```
f2[i]:=G[p^r]:-`^`(a,(3*(i-1)));
```

constructs the second row. The Maple *output* command causes the conversion, for every polynomial in the rows of $H1$, to the corresponding integer. Conversely, the Maple *Convertout* command is used to convert the integers to polynomials. Next, the generator matrix G of the *BCH* code is constructed. To do this, firstly, the Maple *Nullspace* command is used to find a basis for the null space of H as follows:

```
G:=Nullspace(H) mod 2 ;
```

A Maple *for* loop is used after that to place the set of vectors G as rows in the array GG, which forms a generator matrix for the *BCH* code.

```
GG:=[]:
for b from 1 to coldim(H)-rowdim(H) do
GG:=stackmatrix(op(GG),G[b]);
od:
GG:=evalm(GG);
```

Finally, the same process used to construct the code from its generator matrix is used to construct the *BCH* code.

```
A:=array(1..2^rowdim(GG),1..rowdim(GG)):
for jj from 1 to 2^rowdim(GG) do
cccb:=convert(jj-1,base,2);
ddd:=vectdim(cccb);
bvvv:=array(1..rowdim(GG)):
for kkk from 1 to rowdim(GG) do
  bvvv[kkk]:=0:
od:
for iii from 1 to ddd do
bvvv[rowdim(GG)er which have generator matrix o-iii+1]:=cccb[iii]:
od:
for lll from 1 to rowdim(GG) do
A[jj,lll]:=bvvv[lll]:
od:
od:
BCH:=map(irem,multiply(A,GG),2);
rowdim(BCH);
map( irem, multiply(GG,transpose(H) ), 2);
```

Using Maple to compute the MacWilliams identity

The MacWilliams identity can be obtained easily using Maple. This section illustrates how this can be done. To calculate the weight enumerator of a [7,4,3] Hamming code with Maple, the weight enumerator of its dual code and the MacWilliams identity are entered first.

```
B:=y->1+7*y^4;
A:=(z,q,k,n)->(q^k)*(((1+(q-1)*z)/q)^n)*B((1-z)/(1+(q-1)*z));
```

$$B := y \to 1 + 7y^4$$

$$A := (z, q, k, n) \to q^k \left(\frac{1 + (q-1)z}{q}\right)^7 B\left(\frac{1-z}{1+(q-1)z}\right)$$

After that, a Maple *series* command is used to generate a truncated series expansion of the function A, with respect to the variable z about the point 0. In fact, Maple performs series calculations up to order 6. In order to use a different order, a large upper limit is specified, and the final term can be ignored.

```
A(z):=series(A(z,2,4,7),z=0,999);
```

$$A(z) := 1 + 7z^3 + 7z^4 + z^7 + O(z^{999});$$

Alternatively, the series can be converted to a polynomial by using a Maple *convert* command, which leads to the order term being dropped.

```
A:=convert(A(z),polynom);
```

$$A := 1 + 7z^3 + 7z^4 + z^7$$

Another option is to convert this polynomial to a functional operator by using an *unapply* command.

```
unapply(A,z);
```

$$z- > 1 + 7z^3 + 7z^4 + z^7$$

As another example, the weight enumerator of the dual code of a [15,7] BCH code can be computed using the MacWilliams identities as follows. After constructing the BCH code as shown before, the weight enumerator of the BCH code is calculated.

```
R:=array(1..coldim(BCH)):
for q from 1 to coldim(BCH) do
R[q]:=0:
od:
s:=array(1..rowdim(BCH)):
for t from 1 to rowdim(BCH) do
s[t]:=0:
for tt from 1 to coldim(BCH) do
if BCH[t,tt] = 1 then s[t]:=s[t]+1:
fi:
od:
s[t]:=(s[t]):
od:
eval(s);
for p from 1 to rowdim(BCH) do:
for pp from 1 to coldim(BCH) do :
if s[p]=pp then R[pp]:=R[pp]+1:
fi:
od:
od:
eval(R);
B:=y^0:
for ppp from  1 to coldim(GG) do
B:=B+R[ppp]*y^(ppp):
od:
B:=eval(B);
BBB:= unapply(B,y);
```

Lines 5 to 21 of the Maple code are used to compute the weight of every codeword in the BCH code and then store them in an array called s. The array s is used to get the array R, which contains the numbers of codewords of each weight. This is used to build the weight enumerator of the code. Next, the Maple *unapply* command is used to convert the weight enumerator to a functional operator.

$$BBB := y \to 1 + 18y^5 + 30y^6 + 15y^7 + 15y^8 + 30y^9 + 18y^{10} + y^{15}$$

Finally, the same process as in the last example is used to get the weight enumerator of the dual BCH code using the MacWilliams identities.

```
A:=(zw,p,k,n)->(p^k)*(((1+(p-1)*zw)/p)^n)*BBB((1-zw)/(1+(p-1)*zw));
A(zw):=series(A(zw,2,rowdim(H),coldim(H)),zw=0,999);
```

$$zw \to 1 + 15zw^4 + 100zw^6 + 75zw^8 + 60zw^{10} + 5zw^{12} + O(zw^{999})$$

In general, the same process can be used to apply the MacWilliams identity to any linear codes.

Constructing linear codes in Magma

The constructions of four very important families of linear codes: *Hamming Codes, BCH Codes, Reed Muller Codes* and *Cyclic Codes* using *Maple* were illustrated. However, Magma software has a Linear Codes package included, which makes the construction of such families of codes much easier. In this section the construction of these families of codes using Magma is presented.

Hamming codes constructed with Magma

This section shows how Magma constructs the $[n, k, d]$ *Hamming codes*. Firstly, the generator matrix of the *Hamming code* is constructed using the following commands:

```
r:=3;
H:=HammingCode(GF(2),r);
H;
[7, 4, 3] Hamming code (r = 3) over GF(2)
Generator matrix:
[1 0 0 0 0 1 1]
[0 1 0 0 1 1 0]
[0 0 1 0 1 0 1]
[0 0 0 1 1 1 1]
```

As can seen, by using the command

```
HammingCode(K,n)
```

the r^{th} order *Hamming code* over Finite Field K is obtained. Next, to obtain the codewords of the Hamming code from its generator matrix, a *for* loop statement is used.

```
for z in H do
z;
end for;
```

This loop produces each codeword z in the Hamming code H and prints it out as follows:

```
(0 0 0 0 0 0 0)
(1 0 0 0 1 1 0)
(1 1 0 0 1 0 1)
(0 1 0 0 0 1 1)
(0 1 1 0 1 0 0)
(1 1 1 0 0 1 0)
(1 0 1 0 0 0 1)
(0 0 1 0 1 1 1)
(0 0 1 1 0 1 0)
(1 0 1 1 1 0 0)
(1 1 1 1 1 1 1)
(0 1 1 1 0 0 1)
(0 1 0 1 1 1 0)
(1 1 0 1 0 0 0)
(1 0 0 1 0 1 1)
(0 0 0 1 1 0 1)
16
```

Finally, by using the command

```
Dual(H);
```

The parity check matrix of the Hamming code is obtained as follows:

```
[1 0 1 0 1 1 0]
[0 1 1 0 0 1 1]
[0 0 0 1 1 1 1]
```

To obtain the dual code, the same process is used as for the original Hamming code H:

```
(0 0 0 0 0 0 0)
(1 0 0 1 0 1 1)
(1 1 0 0 1 0 1)
(0 1 0 1 1 1 0)
(0 1 1 1 0 0 1)
(1 1 1 0 0 1 0)
(1 0 1 1 1 0 0)
(0 0 1 0 1 1 1)
8
```

BCH codes constructed with Magma

This section shows how Magma can be used to construct the BCH code and its dual. As for the Hamming code, Magma has a special command that can be used to construct the BCH code. In fact the command:

```
BCHCode(K,n,d)
```

gives the generator matrix of a BCH code of length n over the finite field K and minimum distance d. For example, the $[15, 7]$ BCH code can be constructed as follows:

```
n:=15;
d:=5;
C:=BCHCode(GF(2),n,d);
C;
```

The output will be in the form:

```
[15, 7, 5] "BCH code (d = 5, b = 1)" Linear Code over GF(2)
Generator matrix:
[1 0 0 0 0 0 0 1 0 0 0 1 0 1 1]
[0 1 0 0 0 0 0 1 1 0 0 1 1 1 0]
[0 0 1 0 0 0 0 0 1 1 0 0 1 1 1]
[0 0 0 1 0 0 0 1 0 1 1 1 0 0 0]
[0 0 0 0 1 0 0 0 1 0 1 1 1 0 0]
[0 0 0 0 0 1 0 0 0 1 0 1 1 1 0]
[0 0 0 0 0 0 1 0 0 0 1 0 1 1 1]
```

Next, the same processes as shown before are used to construct the code from its generator matrix, and to construct the parity check matrix and the dual code.

```
for z in C do
z;
end for;
D:=Dual(C);
D;
for H in D do
H;
end for;
```

In Magma the output can be printed to file by using the command:

```
PrintFile(F,X);
```

This command prints X to the file specified by the string F. For example, the output in the above BCH code can be printed to the file C:/AMAGNA/bch_out.cod as follows:

```
n:=15;
d:=5;
C:=BCHCode(GF(2),n,d);
PrintFile("C:/AMAGNA/bch_out.cod",C);
PrintFile("C:/AMAGNA/bch_out.cod","  ");
PrintFile("C:/AMAGNA/bch_out.cod","List of codewords:");
for z in C do
PrintFile("C:/AMAGNA/bch_out.cod",z);
end for;
PrintFile("C:/AMAGNA/bch_out.cod",#C);
"BCH code found";
read string;
D:=Dual(C);
PrintFile("C:/AMAGNA/bch_out.cod",D);
PrintFile("C:/AMAGNA/bch_out.cod","  ");
```

```
PrintFile("C:/AMAGNA/bch_out.cod","List of codewords:");
for H in D do
PrintFile("C:/AMAGNA/bch_out.cod",H);
end for;
PrintFile("C:/AMAGNA/bch_out.cod",#D);
"Dual BCH found";
```

Reed Muller Codes with Magma

To obtain the generator matrix of r^{th} order binary*Reed Muller code* of length $n = 2^m$, the command

```
ReedMullerCode(r,m)
```

is used. For example, $(2, 4)$ Reed Muller code can be obtained follows:

```
r:=2;
m:=4;
R:=ReedMullerCode(r,m);
R;
for z in R do
z;
end for;
D:=Dual(R);
D;
for s in D do
s;
end for;
```

As can be seen, the first four commands are used to construct the generator matrix of the $(1, 4)$ ReedMuller code and the result will be as follows:

```
[16, 11, 4] "Reed-Muller Code (r = 2, m = 4)" Linear Code over GF(2)
Generator matrix:
[1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1]
[0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0]
[0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0]
[0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1]
[0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0]
[0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1]
[0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1]
[0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0]
[0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1]
[0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1]
[0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1]
```

After that the same procedures as shown before are used to obtain the Reed Muller code and its dual.

Cyclic Codes with Magma

Recall that all factors of $x^n - 1$ have the same multiplicities.

Suppose that all cyclic codes C of length $n = 14$ are to be constructed. Begin by factorising the polynomial $f(x) = x^{14} - 1 \in GF(2)[x]$. Products of factors will generate a nested chain of cyclic codes of length $n = 14$.

```
>P<x>:= PolynomialRing(GF(2));
>n:=14;
>F:= Factorization(x^n-1);
>F;
>s:=#F;
>s;

[
    <x + 1, 2>,
    <x^3 + x + 1, 2>,
    <x^3 + x^2 + 1, 2>
]
3
```

To ensure that if $x^n - 1$ is a perfect square, then repeated factors are counted and extracted correctly, the following Magma code is used.

```
>nn:=n;
>while nn eq 2*(nn div 2) do
>nn:=nn div 2;
>r:=r*2;
end while;
>F:=Factorization(x^(nn)-1);
>F;

[
    <x + 1, 1>,
    <x^3 + x + 1, 1>,
    <x^3 + x^2 + 1, 1>
]
```

Next, the cartesian power set $L := A^s$ is constructed to get all possibilities for the generator polynomial of a cyclic code C.

```
A:={0..r};
L:=CartesianPower(A,#F);
```

A *for* loop is used to evaluate the generator polynomial which corresponds to every cardinality in L. After that the command *CyclicCode(n, K)* constructs the cyclic code C over K of length n corresponding to each generator polynomial.

```
for u in L do
if u ne Rep(L) then
Gens:=x^0;
u;
read string;
for i:=1 to #F do
u[i];
```

```
Gens:= Gens *(F[i][1])^(u[i]);
end for;
Gens;
C:=CyclicCode(n, Gens);
end if;
end for;
```

The fact that the multiplicities of all factors of $x^n - 1$ are the same can be confirmed by the following Magma code.

```
P<x>:= PolynomialRing(GF(2));
k:=1000;
for n:= 1 to k do
F:= Factorization(x^n-1);
for i:= 1 to #F do
if i ne #F then
for j:=i+1 to #F do
if F[i][2] ne F[j][2] then
F;
else
end if;
end for;
else
end if;
end for;
end for;
```

Using Magma to compute the MacWilliams identities

It is easy to compute the Hamming weight enumerator and complete weight enumerator, and to apply the MacWilliams identities of a linear code C in Magma by using the following commands:

- *WeightEnumerator(C):* Gives the Hamming weight enumerator $W_C(x, y)$ of the linear code C.
- *CompleteWeightEnumerator(C):* Gives the complete weight enumerator $W_C(z_0, \ldots, z_{q-1})$ of the linear code C.
- *MacWilliamsTransform(n,k,K,w):* Applies the MacWilliams transform to W to obtain the complete weight enumerator $W^{'}$ of the dual code of C.

For example, the weight distribution of a $[7, 4, 3]$ Hamming code can be obtained as follows:

```
r:=3;
H:=HammingCode(GF(2),r);
H;
for z in H do
z;
end for;
#H;
```

```
W<x,y>:=WeightEnumerator(H);
W;
CW<u,v>:=CompleteWeightEnumerator(H);
CW;
Ma:=MacWilliamsTransform(7,4,GF(2),CW);
Ma;
```

```
[7, 4, 3] "Hamming code (r = 3)" Linear Code over GF(2)
Generator matrix:
[1 0 0 0 1 1 0]
[0 1 0 0 0 1 1]
[0 0 1 0 1 1 1]
[0 0 0 1 1 0 1]

(0 0 0 0 0 0 0)
(1 0 0 0 1 1 0)
(1 1 0 0 1 0 1)
(0 1 0 0 0 1 1)
(0 1 1 0 1 0 0)
(1 1 1 0 0 1 0)
(1 0 1 0 0 0 1)
(0 0 1 0 1 1 1)
(0 0 1 1 0 1 0)
(1 0 1 1 1 0 0)
(1 1 1 1 1 1 1)
(0 1 1 1 0 0 1)
(0 1 0 1 1 1 0)
(1 1 0 1 0 0 0)
(1 0 0 1 0 1 1)
(0 0 0 1 1 0 1)

16

x^7 + 7*x^4*y^3 + 7*x^3*y^4 + y^7
u^7 + 7*u^4*v^3 + 7*u^3*v^4 + v^7
u^7 + 7*u^3*v^4
```

CPU time comparisons for Maple and Magma

Both Maple and Magma software are provided with a timing facility, where the CPU time (in seconds) used since the start of the Maple or Magma session is returned. In Maple the *time()* command returns the total CPU time since the beginning of the Maple session. In Magma, the *Cputime()* command returns the CPU time used since the beginning of the Magma session. To illustrate the difference between CPU time in both Maple and Magma, consider cyclic codes of lengths 60 to 73. These codes can be constructed using Maple and Magma by following the same processes as in the construction of linear codes using Maple and Magma. In addition, to compute and record the CPU time used to construct each set of cyclic codes, the commands are used in the following way. In Maple:

n	Magma time	Maple time
60	3.346	129.222
61	0.003	0.007
62	2.548	99.559
63	9.989	533.960
64	0.070	0.003
65	0.159	5.804
66	0.303	11.844
67	0.003	0.0125
68	0.163	5.440
69	0.087	2.902
70	1.053	39.592
71	0.010	0.249
72	1.099	37.826
73	0.852	0.703

Table 9.1: **CPU time comparisons for Maple and Magma**

```
st:= time():
```

is added to the beginning of the Maple code, and the command

```
time() - st;
```

is added at the end in order to return the time taken to evaluate the result.

Similarly, using Magma, the commands are:

```
tt:=Cputime();
```

at the beginning of the Magma code and the command

```
Cputime(tt);
```

at the end, in order to return the time taken to evaluate the result. It was found that more consistent CPU times were obtained if the computation was repeated many times and the total CPU time obtained was then divided by the number of repetitions.

In Table 9.1 the CPU times taken to construct each set of cyclic codes in Maple and Magma are shown. It can be seen that in most cases Magma is faster than Maple, sometimes much faster.

It is clear from the description in this appendix that Magma has many advantages over Maple for the construction of DNA codes.

Appendix II

Tables presented here give the best lower bounds for $A_4^{GC}(n,d,w)$ and $A_4^{GC,RC}(n,d,w)$ from all codes constructed. Many new best codes are shown.

1. Entries in bold face denote new best lower bounds.
2. Codes with length greater than 20 are all new.
3. Entries in italics are codes which match the best known values as given (for $n \leq 20$) at http://llama.med.harvard.edu/∼king/dnacodes.html

The labels in the tables have the following meaning:

Subscripts		Superscripts	
cf	cyclic linear code over GF(4)	co	coset of code
ef	extended cyclic linear code over GF(4)		
ca	cyclic additive code over GF(4)	i	position for shortening
ea	extended cyclic additive code over GF(4)		or puncturing
pr	linear puncturing of the code below and one position to the right		
pb	puncturing of the code below	Xi	letter and position
sb	linear shortening of the code below		for nonlinear
nb	nonlinear shortening of the code below		shortening
st	linear shortening of the code two positions below		
pt	puncturing of the code two positions below and two positions to the right		
nt	nonlinear shortening of the code two positions below		
$pt1$	linear puncturing twice of the code two positions below and one position to the right		
sl2	linear shortening twice of the code two positions below and two positions to the left		

The labels in the Tables 10.65 to 10.72 have the following meaning:

Subscripts	
a	best codes obtained after shortening and puncturing
b	best codes obtained from best linear codes
c	best codes obtained from codes with an all 1s vector in the dual
gk	results taken from [16]
$m1$	variable neighbourhood search results taken from [31]
$m2$	variable neighbourhood search results or simulated annealing results taken from [32]
$m3$	results taken from [33]
$L1$	best codes generated in Chapter 7
$L2$	best codes from shortening best linear codes generated in Chapter 7

Cyclic codes over GF(4) have been computed for $4 \leq n \leq 30$.
Extended cyclic codes over GF(4) have been computed for $4 \leq n \leq 30$.
Additive cyclic codes over GF(4) have been computed for $4 \leq n \leq 19$.
Additive extended cyclic codes over GF(4) have been computed for $4 \leq n \leq 20$.
Cyclic codes over Z_4 have been computed for $4 \leq n \leq 24$.

Extended cyclic codes over Z_4 have been computed for $4 \leq n \leq 24$.
Cosets of Cyclic codes over GF(4) have been computed for $4 \leq n \leq 20$.
Cosets of extended cyclic codes over GF(4) have been computed for $4 \leq n \leq 20$.
Cosets of additive cyclic codes over GF(4) have been computed for $4 \leq n \leq 14$.
Cosets of additive extended cyclic codes over GF(4) have been computed for $4 \leq n \leq 15$.
Cosets of cyclic codes over Z_4 have been computed for $4 \leq n \leq 20$.
Cosets of extended cyclic codes over Z_4 have been computed for $4 \leq n \leq 20$.

n/d	3	4	5	6	7	8	9	10	11
4	-	2	-	-	-	-	-	-	-
5	20	*10*	2	-	-	-	-	-	-
6	36	12	-	3	-	-	-	-	-
7	112	56	-	-	2	-	-	-	-
8	-	48	-	-	-	2	-	-	-
9	504	-	-	12	-	-	3	-	-
10	1008	**1760**	128	20	-	10	-	2	-
11	-	-	924	*462*	-	-	-	-	2
12	29568	8064	-	192	-	12	-	-	-
13	12	-	3432	1716	-	-	-	-	-
14	25600	103936	-	512	8	56	-	-	-
15	*6589440*	*1647360*	55680	**27840**	**6960**	**1600**	120	**120**	30
16	-	8960	-	-	-	48	-	-	-
17	-	12446720	43520	21760	*48620*	*24310*	170	-	-
18	22468608	423936	-	6624	-	-	36	-	-
19	-	-	-	-	184756	92378	-	-	-
20	47297536	376832000	196416	98208	-	23040	-	768	-
21	5778898944	360062976	90295296	5625984	64512	516096	32256	672	8064
22	-	331319296	2821728	1293292	-	-	-	8192	8
23	-	-	-	-	5275648	2637824	-	-	-
24	22152445952	5569511424	-	1636352	-	8064	-	-	-
25	166409600	83204800	320	-	-	-	-	160	-
26	-	19782483968	41602400	19315400	-	-	-	32768	-
27	5134924800	-	-	32256	-	-	504	-	-
28	43536875520	5251000172544	-	16220160	6144	7712768	-	-	-
29	-	-	-	-	-	-	-	-	155117520
30	1219947795578880	304986948894720	161533132800	298023321600	18626457600	4665323520	18163200	18224640	1417920

Table 10.1: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cyclic codes over $GF(4)$ for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	**15**	-	-	*3*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	85	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
18	12	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-
20	20	-	-	-	10	-	-	-	2	-	-	-	-	-	-	-	-	-	-
21	2240	-	12	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-
22	462	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-
24	192	-	-	-	12	-	-	-	-	-	-	-	3	-	-	-	-	-	-
25	-	-	-	20	-	-	-	-	10	-	-	-	-	2	-	-	-	-	-
26	1716	8	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-
27	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-	3	-	-	-
28	3072	-	48	-	56	-	-	-	-	-	-	-	-	-	-	-	2	-	-
29	77558760	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-
30	565440	-	32768	2048	1600	-	120	-	120	-	30	-	15	-	-	-	-	-	3

Table 10.2: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cyclic codes over GF(4) for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	12	2		-	-	-	-	-	-
5	-	2	-	-	-	-	-	-	-
6	-	30	-	2	-	-	-	-	-
7	140	44	-	3	-	-	-	-	-
8	-	224	-	-	-	2	-	-	-
9	-	48	-	-	-	2	-	-	-
10	1680	24	-	12	6	-	3	2	-
11	-	1848	-	168	-	10	-	2	-
12	-	-	-	*1848*	-	-	-	-	-
13	109824	28416	-	192	-	12	-	-	-
14	-	-	-	6006	-	-	-	-	-
15	-	103936	-	736	-	56	-	-	-
16	*26357760*	*6589440*	-	*111360*	**27840**	6300	120	120	96
17	-	8960	-	-	-	48	-	-	-
18	-	22404096	-	78336	-	*87516*	-	306	-
19	23648768	433664	-	6776	80	-	36	32	-
20	-	-	-	-	-	**369512**	-	-	-
21	-	376832000	-	360256	-	23040	-	768	-
22	21189296128	1325449216	90295296	20672512	1292704	516096	32256	20608	8064
23	-	331319296	-	5408312	-	-	-	10040	-
24	-	-	-	-	-	10551296	-	-	-
25	85201715200	21315911680	-	1636352	-	8064	-	-	-
26	-	309046400	-	320	-	-	-	160	-
27	-	19782483968	-	80233200	-	-	-	39632	-
28	20539699200	64512	-	32256	1008	-	504	24	-
29	-	5251000172544	-	26771456	-	7712768	-	-	-
30	-	-	-	-	-	-	-	-	-

Table 10.3: **Best lower bounds for $A_4^{GC}(n,d,w)$ from extended cyclic codes over GF(4) for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	*60*	-	-	3	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	85	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
19	12	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
21	20	-	-	-	10	-	-	-	2	-	-	-	-	-	-	-	-	-	-
22	5082	-	12	6	-	-	-	-	-	3	2	-	-	-	-	-	-	-	-
23	462	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-
25	192	-	-	-	12	-	-	-	-	-	-	-	3	-	-	-	-	-	-
26	-	-	-	-	20	-	-	-	10	-	-	-	-	-	2	-	-	-	-
27	1716	-	8	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-
28	-	-	-	-	-	-	12	6	-	-	-	-	-	-	-	3	2	-	-
29	3072	48	-	-	56	-	-	-	-	-	-	-	-	-	-	-	2	-	-
30	290845350	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2

Table 10.4: **Best lower bounds for $A_4^{GC}(n,d,w)$ from extended cyclic codes over GF(4) for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	20	10	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	40	16	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	112	56	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	64	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
9	504	-	-	12	-	-	3	-	-	-	-	-	-	-	-	-	-	-
10	1008	**2016**	128	64	-	16	-	4	-	-	-	-	-	-	-	-	-	-
11	-	-	924	*462*	-	-	-	-	2	-	-	-	-	-	-	-	-	-
12	30208	8704	-	192	-	16	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	3432	1716	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	25600	118784	-	512	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	*6589440*	*1647360*	55680	**27840**	**6960**	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	16384	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	12446720	51200	-	*48620*	-	-	-	-	-	-	-	-	-	-	-	-	-
18	24893440	507904	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.5: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cosets of cyclic codes over GF(4) for** $3 \leq d \leq 20$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	20	10	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	40	16	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	112	56	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	16	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
9	504	-	-	12	-	-	3	-	-	-	-	-	-	-	-	-	-	-
10	1008	**2016**	128	64	-	16	-	*4*	-	-	-	-	-	-	-	-	-	-
11	-	-	924	*462*	-	-	-	-	2	-	-	-	-	-	-	-	-	-
12	30208	8704	-	192	-	16	-	-	-	*4*	-	-	-	-	-	-	-	-
13	-	-	3432	1716	-	-	-	-	-	-	2	-	-	-	-	-	-	-
14	25600	118784	-	512	8	48	-	-	-	-	-	2	-	-	-	-	-	-
15	*6589440*	*1647360*	55680	**27840**	**6960**	**1920**	144	**124**	30	10	-	-	*3*	-	-	-	-	-
16	-	16384	-	-	-	32	-	-	-	-	-	-	-	*4*	-	-	-	-
17	-	12446720	51200	25600	*48620*	24310	200	-	-	100	-	-	-	-	2	-	-	-
18	24893440	507904	-	6240	-	-	40	-	-	16	-	-	-	-	-	*4*	-	-
19	-	-	-	-	184756	*92378*	-	-	-	-	-	-	-	-	-	-	2	-
20	47297536	**378896384**	196416	98208	-	25056	-	384	-	32	-	-	-	10	-	-	-	*4*

Table 10.6: **Best lower bounds for $A_4^{GC}(n, d, w)$ from random cosets of cyclic codes over GF(4) for** $3 \leq d \leq 20$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	*40*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	140	44	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	224	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	64	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
10	2016	48	-	12	12	-	3	-	-	-	-	-	-	-	-	-	-	-
11	-	2016	-	168	-	16	-	*4*	-	-	-	-	-	-	-	-	-	-
12	-	-	-	*1848*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	109824	28416	-	192	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	6864	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	118784	-	736	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	*26357760*	*6589440*	-	*111360*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	24893440	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.7: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cosets of extended cyclic codes over GF(4) for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	3	*4*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	*40*	-	*4*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	140	44	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	128	-	-	-	*4*	-	-	-	-	-	-	-	-	-	-	-	-
9	-	32	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
10	2016	48	-	12	12	-	3	*4*	-	-	-	-	-	-	-	-	-	-
11	-	2016	-	168	-	16	-	*4*	-	-	-	-	-	-	-	-	-	-
12	-	-	-	1848	-	-	-	-	-	4	-	-	-	-	-	-	-	-
13	109824	28416	-	192	-	16	-	-	-	4	-	-	-	-	-	-	-	-
14	-	-	-	6864	-	-	-	-	-	-	-	*4*	-	-	-	-	-	-
15	-	118784	-	640	-	48	-	-	-	-	-	*4*	-	-	-	-	-	-
16	*26357760*	*6589440*	-	*111360*	25740	6480	144	124	104	32	-	-	3	2	-	-	-	-
17	-	8192	-	-	-	64	-	-	-	-	-	-	-	*4*	-	-	-	-
18	-	24893440	-	102400	-	97240	-	400	-	100	-	-	-	-	-	*4*	-	-
19	24893440	507904	-	7936	140	-	40	28	-	16	-	-	-	-	-	4	-	-
20	-	-	-	-	-	**369512**	-	-	-	-	-	-	-	-	-	-	-	*4*

Table 10.8: **Best lower bounds for $A_4^{GC}(n,d,w)$ from random cosets of extended cyclic codes over GF(4) for** $3 \leq d \leq 20$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	6	2	-	-	-	-	–	-	-	-	-	-	-	-	-	-	-	-
5	10	*10*	2	-	-	-	–	-	-	-	-	-	-	-	-	-	-	-
6	40	32	-	3	-	-	–	-	-	-	-	-	-	-	-	-	-	-
7	**280**	70	14	*7*	2	-	-	-	-	-	-	-	-	-	-	-	-	-
8	304	176	44	12	-	2	-	-	-	-	-	-	-	-	-	-	-	-
9	504	504	126	27	-	-	*3*	-	-	-	-	-	-	-	-	-	-	-
10	1760	**1760**	252	110	-	10	-	2	-	-	-	-	-	-	-	-	-	-
11	462	462	462	*462*	-	-	-	-	2	-	-	-	-	-	-	-	-	-
12	59136	*29568*	1848	1808	168	48	12	-	-	2	-	-	-	-	-	-	-	-
13	1716	1716	1716	1716	-	-	-	-	-	-	2	-	-	-	-	-	-	-
14	777728	*384384*	14848	**6496**	**1624**	203	-	**49**	-	7	-	2	-	-	-	-	-	-
15	*6589440*	*1647360*	**102960**	**27840**	**6960**	**1600**	200	**120**	30	**15**	-	-	*3*	-	-	-	-	-
16	828160	828160	51760	51760	13080	1804	-	44	12	-	-	-	-	2	-	-	-	-
17	6223360	6223360	48620	48620	24310	*24310*	170	170	-	85	-	-	-	-	2	-	-	-
18	22468608	22468608	1400256	1400256	**87768**	22032	**5508**	414	-	32	-	-	-	-	-	3	-	-
19	92378	92378	92378	92378	92378	*92378*	-	-	-	-	-	-	-	-	-	-	2	-
20	-	-	1478048	1478048	739024	369008	**24552**	2880	-	192	-	-	-	-	-	-	-	-

Table 10.9: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cyclic additive codes over GF(4) pairing 0 with 1 for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	*12*	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	20	*10*	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	60	32	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	**280**	70	14	*7*	1	-	-	-	-	-	-	-	-	-	-	-	-	-
8	560	176	44	12	-	2	-	-	-	-	-	-	-	-	-	-	-	-
9	1008	504	126	27	-	-	3	-	-	-	-	-	-	-	-	-	-	-
10	1760	**1760**	**440**	110	-	10	-	2	-	-	-	-	-	-	-	-	-	-
11	924	924	924	*462*	-	-	-	-	1	-	-	-	-	-	-	-	-	-
12	*118272*	*29568*	**3696**	*1848*	168	48	12	-	-	3	-	-	-	-	-	-	-	-
13	3432	3432	3432	**3432**	-	-	-	-	-	-	1	-	-	-	-	-	-	-
14	777728	*384384*	14848	**6496**	**1624**	203	49	-	-	7	-	2	-	-	-	-	-	-
15	*6589440*	*1647360*	**102960**	**27840**	**6960**	**1600**	200	**120**	30	**15**	-	-	*3*	-	-	-	-	-
16	1647360	1647360	102960	102960	13080	1804	-	44	-	12	-	-	-	2	-	-	-	-
17	12446720	12446720	48620	48620	*48620*	*24310*	170	170	-	85	-	-	-	-	1	-	-	-
18	22468608	22468608	1400256	1400256	**87768**	22032	**5508**	414	-	32	-	-	-	-	-	3	-	-
19	184756	184756	184756	184756	184756	*92378*	-	-	-	-	-	-	-	-	-	-	1	-
20	-	-	2956096	1478048	*1478048*	369008	23000	2880	-	192	-	-	-	-	-	-	-	-

Table 10.10: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cyclic additive codes over GF(4) pairing 1 with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	10	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	15	15	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	70	44	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	560	224	-	*28*	-	2	-	-	-	-	-	-	-	-	-	-	-	-
9	504	252	44	12	-	2	-	-	-	-	-	-	-	-	-	-	-	-
10	840	840	252	108	6	-	3	2	-	-	-	-	-	-	-	-	-	-
11	1760	1760	462	168	-	10	-	2	-	-	-	-	-	-	-	-	-	-
12	924	924	924	924	-	-	-	-	-	2	-	-	-	-	-	-	-	-
13	109824	29568	3432	1808	168	48	12	-	-	3	-	-	-	-	-	-	-	-
14	3003	3003	3003	3003	-	-	-	-	-	-	-	2	-	-	-	-	-	-
15	823680	427520	26720	13360	1624	596	-	49	-	7	-	2	-	-	-	-	-	-
16	13178880	3294720	*411840*	*111360*	**27840**	6300	210	120	96	*60*	-	-	3	2	-	-	-	-
17	1555840	1555840	97240	97240	24310	1804	-	44	-	12	-	-	-	2	-	-	-	-
18	11202048	11202048	43758	87516	43758	-	306	306	-	85	-	-	-	-	-	2	-	-
19	22468608	22468608	1478048	1478048	184756	23210	5832	620	-	32	-	-	-	-	-	3	-	-
20	184756	184756	184756	184756	184756	184756	2	-	-	-	-	-	-	-	-	-	-	-

Table 10.11: **Best lower bounds for $A_4^{GC}(n,d,w)$ from extended cyclic additive codes over GF(4) pairing 0 with 1 for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	6	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	12	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	20	20	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	60	44	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	560	224	-	*28*	-	1	-	-	-	-	-	-	-	-	-	-	-	-
9	560	304	44	12	-	2	-	-	-	-	-	-	-	-	-	-	-	-
10	1008	1008	252	108	6	-	3	1	-	-	-	-	-	-	-	-	-	-
11	1760	1760	440	168	-	10	-	2	-	-	-	-	-	-	-	-	-	-
12	924	924	924	924	-	-	-	-	-	1	-	-	-	-	-	-	-	-
13	118272	29568	3696	1848	168	48	12	-	-	3	-	-	-	-	-	-	-	-
14	3432	3432	3432	3432	-	-	-	-	-	-	-	1	-	-	-	-	-	-
15	777728	427520	26720	13360	1624	596	-	49	-	7	-	2	-	-	-	-	-	-
16	13178880	3294720	403200	*111360*	**27840**	6300	210	120	96	*60*	-	-	3	1	-	-	-	-
17	1647360	1647360	102960	102960	25880	1804	-	44	-	12	-	-	-	2	-	-	-	-
18	12446720	12446720	48620	78336	48620	48620	306	306	-	85	-	-	-	-	-	1	-	-
19	22468608	22468608	1400256	1400256	175032	23210	5832	620	-	32	-	-	-	-	-	3	-	-
20	184756	184756	184756	184756	184756	184756	-	-	-	-	-	-	-	-	-	-	-	1

Table 10.12: **Best lower bounds for $A_4^{GC}(n,d,w)$ from extended cyclic additive codes over GF(4) pairing 1 with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	6	*4*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	10	*10*	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	40	*40*	-	*4*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	**280**	70	*14*	*7*	2	-	-	-	-	-	-	-	-	-	-	-	-	-
8	304	70	44	8	-	2	-	-	-	-	-	-	-	-	-	-	-	-
9	504	504	132	33	-	-	*3*	-	-	-	-	-	-	-	-	-	-	-
10	2016	**2016**	252	**126**	-	*16*	-	*4*	-	-	-	-	-	-	-	-	-	-
11	462	462	462	*462*	-	-	-	-	2	-	-	-	-	-	-	-	-	-
12	60416	*29568*	1888	**1888**	118	64	6	-	-	4	-	-	-	-	-	-	-	-
13	1716	1716	1716	1716	-	-	-	-	-	-	2	-	-	-	-	-	-	-
14	878592	**439296**	13728	**6864**	**1716**	232	-	28	-	6	-	2	-	-	-	-	-	-
15	*6589440*	*1647360*	**102960**	*25740*	6435	**1620**	**240**	**120**	31	10	-	-	*3*	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.13: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cosets of cyclic additive codes over GF(4) pairing 1 with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	6	*4*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	20	*10*	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	80	*40*	-	*4*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	**280**	70	14	*7*	1	-	-	-	-	-	-	-	-	-	-	-	-	-
8	560	152	38	8	-	*4*	-	-	-	-	-	-	-	-	-	-	-	-
9	1008	504	132	33	-	-	3	-	-	-	-	-	-	-	-	-	-	-
10	2016	**2016**	**504**	110	-	*16*	-	*4*	-	-	-	-	-	-	-	-	-	-
11	924	924	924	*462*	-	-	-	-	1	-	-	-	-	-	-	-	-	-
12	*118272*	25344	**3168**	*1848*	108	30	6	-	-	3	-	-	-	-	-	-	-	-
13	3432	3432	3432	**3432**	-	-	-	-	-	-	1	-	-	-	-	-	-	-
14	878592	**439296**	14848	**6864**	**1716**	256	-	28	-	6	-	2	-	-	-	-	-	-
15	*6589440*	*1647360*	**103680**	*25740*	**6960**	1620	**240**	124	31	10	-	-	*3*	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.14: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cosets of cyclic additive codes over GF(4) pairing 0 with 1 for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	12	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	20	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	80	44	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	560	224	-	*28*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	560	304	44	8	-	4	-	-	-	-	-	-	-	-	-	-	-	-
10	1008	1008	264	**132**	12	-	3	-	-	-	-	-	-	-	-	-	-	-
11	2016	2016	504	126	-	16	-	*4*	-	-	-	-	-	-	-	-	-	-
12	924	924	924	924	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	118272	30208	3776	1888	132	40	8	-	-	*4*	-	-	-	-	-	-	-	-
14	3432	3432	3432	3432	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	878592	439296	26720	13728	1856	596	-	50	-	4	-	3	-	-	-	-	-	-
16	13178880	3294720	**414720**	*111360*	25600	6480	240	116	104	48	-	-	3	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.15: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cosets of extended cyclic additive codes over GF(4) pairing** 0 **with** 1 **for** $3 \leq d \leq 20$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	10	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	20	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	70	44	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	560	140	-	*28*	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	504	252	44	16	-	4	-	-	-	-	-	-	-	-	-	-	-	-
10	1008	1008	264	**132**	12	-	3	-	-	-	-	-	-	-	-	-	-	-
11	2016	2016	462	126	-	16	-	3	-	-	-	-	-	-	-	-	-	-
12	924	924	924	924	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	109824	29568	3432	1776	136	40	10	-	-	4	-	-	-	-	-	-	-	-
14	3432	3432	3432	3432	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	878592	439296	26720	13728	1856	424	-	50	-	8	-	*4*	-	-	-	-	-	-
16	13178880	3294720	**411840**	111360	25600	6400	240	108	104	48	-	-	3	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.16: **Best lower bounds for $A_4^{GC}(n,d,w)$ from cosets of extended cyclic additive codes over GF(4) pairing 1 with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
4	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	8	9	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	70	35	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	56	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	24	-	-	9	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	110	8	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
12	1224	4832	-	36	-	9	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
14	13728	54264	-	288	8	49	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-
15	823680	231660	2820	1215	300	150	-	-	9	-	-	-	2	-	-	-	-	-	-	-	-
16	-	10992	-	-	-	56	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-
17	-	-	49980	13600	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-
18	1280	225954	-	320	-	-	8	-	-	9	-	-	-	-	-	2	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-
20	-	12944192	96	-	-	110	-	36	-	-	-	-	-	-	-	-	-	2	-	-	-
21	722327424	183333696	2828280	1055964	798	90888	70	294	-	35	-	9	-	-	-	-	-	-	2	-	-
22	-	323554	-	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-	-	2	-
23	-	-	-	-	2704156	1352078	-	2	-	-	-	-	-	-	-	-	-	-	-	-	2

Table 10.17: **Lower bounds for $A_4^{GC}(n,d,w)$ from all cyclic codes over Z_4 with one invertible element and one non-invertible element for $3 \leq d \leq 23$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
4	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	8	12	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	112	56	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	48	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	24	-	-	12	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	160	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
12	1536	6144	-	48	-	12	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
14	25600	103936	-	512	8	56	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-
15	890880	445440	3840	1920	480	240	-	12	-	-	-	-	2	-	-	-	-	-	-	-	-
16	-	8960	-	-	-	48	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-
17	-	-	43520	21760	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-
18	1280	423936	-	480	-	-	8	-	-	12	-	-	-	-	-	2	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-
20	-	23592960	96	-	-	160	-	48	-	-	-	-	-	-	-	-	-	2	-	-	-
21	712900608	356450304	4128768	2064384	1344	143360	112	672	-	56	-	12	-	-	-	-	-	-	2	-	-
22	-	473088	-	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-	-	2	-
23	-	-	-	-	5275648	2637824	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2

Table 10.18: **Lower bounds for $A_4^{GC}(n,d,w)$ from all cyclic codes over Z_4 with two invertible elements for $3 \leq d \leq 23$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
4	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	9	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	140	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	56	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	12	-	9	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	110	-	6	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
13	-	4832	-	24	-	9	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
15	-	54264	-	391	-	49	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
16	-	926640	-	4290	-	480	-	9	-	-	-	-	-	2	-	-	-	-	-	-	-	-
17	-	10992	-	-	-	56	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-
18	-	-	-	49980	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-
19	-	225954	-	320	-	-	-	6	-	9	-	-	-	-	-	2	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-
21	-	12944192	-	60	-	110	-	24	-	-	-	-	-	-	-	-	-	2	-	-	-	-
22	-	722327424	-	2828280	-	90888	-	294	-	35	-	9	-	-	-	-	-	-	-	1	-	-
23	-	323554	-	-	-	-	-	-	-	6	-	-	-	-	-	-	-	-	-	2	-	-
24	-	-	-	-	-	5408312	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2

Table 10.19: **Lower bounds for $A_4^{GC}(n,d,w)$ from all extended cyclic codes over Z_4 with one invertible element and one non-invertible element for $3 \leq d \leq 24$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
4	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	12	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	224	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	48	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	24	-	12	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	160	-	8	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
13	-	6144	-	48	-	12	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
15	-	103936	-	736	-	56	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
16	-	1781760	-	6144	-	960	-	12	-	-	-	-	-	2	-	-	-	-	-	-	-	-
17	-	8960	-	-	-	48	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-
18	-	-	-	78336	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-
19	-	423936	-	480	-	-	-	8	-	12	-	-	-	-	-	2	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-
21	-	23592960	-	96	-	160	-	48	-	-	-	-	-	-	-	-	-	2	-	-	-	-
22	-	1323040768	-	5275648	-	143360	-	672	-	56	-	12	-	-	-	-	-	-	-	2	-	-
23	-	473088	-	-	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-	2	-	-
24	-	-	-	-	-	10551296	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2

Table 10.20: **Lower bounds for $A_4^{GC}(n,d,w)$ from all extended cyclic codes over Z_4 with two invertible elements for $3 \leq d \leq 24$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	8	16	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	112	56	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	64	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
9	24	-	-	12	-	-	2	-	-	-	-	-	-	-	-	-	-	-
10	-	256	8	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-
12	1536	8704	-	48	-	16	-	-	-	4	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-
14	25600	118784	-	512	8	64	-	-	-	-	-	4	-	-	-	-	-	-
15	890880	445440	4608	1984	480	240	-	12	-	-	-	-	2	-	-	-	-	-
16	-	16384	-	-	-	64	-	-	-	-	-	-	-	4	-	-	-	-
17	-	-	51200	25600	-	-	-	-	-	-	-	-	-	-	2	-	-	-
18	1280	507904	-	640	-	-	8	-	-	16	-	-	-	-	-	4	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-
20	-	25657344	96	-	-	256	-	48	-	-	-	-	-	-	-	-	-	4

Table 10.21: **Lower bounds for $A_4^{GC}(n,d,w)$ from random cosets of cyclic codes over Z_4 with two invertible elements for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	8	16	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	70	35	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	64	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
9	24	-	-	9	-	-	2	-	-	-	-	-	-	-	-	-	-	-
10	-	128	8	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-
12	1224	4832	-	36	-	16	-	-	-	4	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-
14	13728	62336	-	288	8	49	-	-	-	-	-	4	-	-	-	-	-	-
15	823680	231660	3308	1240	300	150	-	9	-	-	-	-	2	-	-	-	-	-
16	-	10992	-	-	-	56	-	-	-	-	-	-	-	4	-	-	-	-
17	-	-	49980	13600	-	-	-	-	-	-	-	-	-	-	2	-	-	-
18	1280	244200	-	320	-	-	8	-	-	16	-	-	-	-	-	4	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-
20	-	12944192	96	-	-	128	-	36	-	-	-	-	-	-	-	-	-	4

Table 10.22: **Lower bounds for $A_4^{GC}(n, d, w)$ from random cosets of cyclic codes over Z_4 with one invertible element and one non-invertible element for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	10	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	140	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
9	-	56	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
10	-	36	-	9	-	-	-	4	-	-	-	-	-	-	-	-	-	-
11	-	126	-	9	-	-	-	4	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-
13	-	4912	-	28	-	10	-	-	-	4	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-
15	-	62336	-	391	-	49	-	-	-	-	-	4	-	-	-	-	-	-
16	-	926640	-	4302	-	480	-	9	-	-	-	-	-	4	-	-	-	-
17	-	10992	-	-	-	56	-	-	-	-	-	-	-	4	-	-	-	-
18	-	-	-	52140	-	-	-	-	-	-	-	-	-	-	-	4	-	-
19	-	243320	-	400	-	-	-	9	-	10	-	-	-	-	-	3	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4

Table 10.23: **Lower bounds for $A_4^{GC}(n,d,w)$ from random cosets of extended cyclic codes over Z_4 with one invertible element and one non-invertible element for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	16	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	224	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
9	-	64	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-
10	-	48	-	12	-	-	-	4	-	-	-	-	-	-	-	-	-	-
11	-	256	-	12	-	-	-	4	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-
13	-	8704	-	48	-	16	-	-	-	4	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-
15	-	118784	-	736	-	64	-	-	-	-	-	4	-	-	-	-	-	-
16	-	1781760	-	6656	-	960	-	12	-	-	-	-	-	4	-	-	-	-
17	-	16384	-	-	-	64	-	-	-	-	-	-	-	4	-	-	-	-
18	-	-	-	102400	-	-	-	-	-	-	-	-	-	-	-	4	-	-
19	-	507904	-	640	-	-	-	12	-	16	-	-	-	-	-	4	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4

Table 10.24: **Lower bounds for $A_4^{GC}(n,d,w)$ from random cosets of extended cyclic codes over Z_4 with two invertible elements for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11
4	12_{ef}	4^{co}_{cf}	-	-	-	-	-	-	-
5	20_{cf}	10_{cf}	2_{cf}	-	-	-	-	-	-
6	60_{ca}	40^{co}_{ef}	-	4^{co}_{cf}	-	-	-	-	-
7	$\mathbf{280}_{ca}$	70_{ca}	14_{ca}	7_{ca}	2_{cf}	-	-	-	-
8	560_{ca}	224_{ef}	44_{ca}	28_{ea}	-	4^{co}_{cf}	-	-	-
9	1008_{ca}	504_{ca}	132^{co}_{ca}	33^{co}_{ca}	-	4^{co}_{ef}	3_{cf}	-	-
10	2016^{co}_{ef}	$\mathbf{2016}^{co}_{cf}$	$\mathbf{504}^{co}_{ca}$	$\mathbf{132}^{co}_{ea}$	12^{co}_{ef}	16^{co}_{cf}	3_{ef}	4^{co}_{cf}	-
11	2016^{co}_{ea}	2016^{co}_{ef}	924_{cf}	462_{cf}	-	16^{co}_{ef}	-	4^{co}_{ef}	2_{cf}
12	118272_{ca}	29568_{ca}	$\mathbf{3696}_{ca}$	$\mathbf{1888}^{co}_{ca}$	168_{ca}	48_{ca}	12_{ca}	-	-
13	118272_{ea}	30208^{co}_{ea}	3776^{co}_{ea}	$\mathbf{3432}_{ca}$	168_{ea}	48_{ea}	12_{ea}	-	-
14	878592^{co}_{ca}	$\mathbf{439296}^{co}_{ca}$	14848_{ca}	$\mathbf{6864}^{co}_{ef}$	$\mathbf{1624}_{ca}$	256^{co}_{ca}	-	$\mathbf{49}_{ca}$	-
15	6589440_{cf}	1647360_{cf}	$\mathbf{102960}_{ca}$	$\mathbf{27840}_{cf}$	$\mathbf{6960}_{cf}$	$\mathbf{1920}^{co}_{cf}$	200_{ca}	$\mathbf{124}^{co}_{cf}$	30_{cf}
16	26357760_{ef}	6589440_{ef}	411840_{ea}	111360_{ef}	$\mathbf{27840}_{ef}$	6480^{co}_{ef}	210_{ea}	124^{co}_{ef}	104^{co}_{ef}
17	12446720_{ca}	12446720_{cf}	102960_{ea}	102960_{ea}	48620_{cf}	24310_{cf}	200^{co}_{cf}	170_{ca}	-
18	24893440^{co}_{cf}	24893440^{co}_{ef}	1400256_{ca}	1400256_{ca}	$\mathbf{87768}_{ca}$	87516_{ef}	$\mathbf{5508}_{ca}$	414_{ca}	-
19	24893440^{co}_{ef}	22468608_{ea}	1478048_{ea}	1478048_{ea}	184756_{cf}	92378_{cf}	5832_{ea}	620_{ea}	-
20	47297536_{cf}	$\mathbf{378896384}^{co}_{cf}$	196416_{cf}	184756_{ea}	184756_{ea}	$\mathbf{369512}_{ef}$	2_{ea}	768_{cf}	-
21	5778898944_{cf}	376832000_{ef}	90295296_{cf}	5625984_{cf}	64512_{cf}	516096_{cf}	32256_{cf}	768_{ef}	8064_{cf}
22	21189296128_{ef}	1325449216_{ef}	90295296_{ef}	20672512_{ef}	1292704_{ef}	516096_{ef}	32256_{ef}	20608_{ef}	8064_{ef}
23	-	331319296_{ef}	-	5408312_{ef}	5275648_{cf}	2637824_{cf}	-	10040_{ef}	-
24	22152445952_{cf}	5569511424_{cf}	-	1636352_{cf}	-	10551296_{ef}	-	-	-
25	85201715200_{ef}	21315911680_{ef}	320_{cf}	1636352_{ef}	-	8064_{ef}	-	160_{cf}	-
26	-	19782483968_{cf}	41602400_{cf}	19315400_{cf}	-	-	-	32768_{cf}	-
27	5134924800_{cf}	19782483968_{ef}	-	80233200_{ef}	-	-	504_{cf}	39632_{ef}	-
28	43536875520_{cf}	5251000172544_{cf}	-	16220160_{cf}	6144_{cf}	7712768_{cf}	504_{ef}	24_{ef}	-
29	-	5251000172544_{ef}	-	26771456_{ef}	-	7712768_{ef}	-	-	155117520_{cf}
30	1219947795578880_{cf}	304986948894720_{cf}	161533132800_{cf}	298023321600_{cf}	18626457600_{cf}	4665323520_{cf}	18163200_{cf}	18224640_{cf}	1417920_{cf}

Table 10.25: **Best lower bounds for $A_4^{GC}(n,d,w)$ before shortening or puncturing** $3 \leq d \leq 11$

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	4^{co}_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	4^{co}_{ef}	2_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	7_{ca}	-	4^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	$\mathbf{15}_{cf}$	-	4^{co}_{ef}	3_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	60_{ef}	-	-	3_{ef}	4^{co}_{cz}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	100^{co}_{cf}	-	-	-	4^{co}_{ef}	2_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-
18	85_{ef}	-	-	-	-	-	4^{co}_{cf}	-	-	-	-	-	-	-	-	-	-	-	-
19	32_{ea}	-	-	-	-	-	3_{ef}	2_{cf}	-	-	-	-	-	-	-	-	-	-	-
20	32^{co}_{cf}	-	-	-	10_{cf}	-	-	-	4^{co}_{cf}	-	-	-	-	-	-	-	-	-	-
21	2240_{cf}	-	12_{cf}	-	10_{ef}	-	-	-	2_{ef}	3_{cf}	-	-	-	-	-	-	-	-	-
22	5082_{ef}	-	12_{ef}	6_{ef}	-	-	-	-	-	3_{ef}	2_{cf}	-	-	-	-	-	-	-	-
23	462_{ef}	-	-	-	-	-	-	-	-	-	2_{ef}	2_{cf}	-	-	-	-	-	-	-
24	192_{cf}	-	-	-	12_{cf}	-	-	-	-	-	-	-	3_{cf}	-	-	-	-	-	-
25	192_{ef}	-	-	20_{cf}	12_{ef}	-	-	-	10_{cf}	-	-	-	3_{ef}	2_{cf}	-	-	-	-	-
26	1716_{cf}	8_{cf}	-	-	20_{ef}	-	-	-	10_{ef}	-	-	-	-	-	2_{cf}	-	-	-	-
27	1716_{ef}	-	8_{ef}	-	-	-	12_{cf}	-	-	-	-	-	-	-	2_{ef}	3_{cf}	-	-	-
28	3072_{cf}	-	48_{cf}	-	56_{cf}	-	12_{ef}	6_{ef}	-	-	-	-	-	-	-	3_{ef}	2_{cf}	-	-
29	77558760_{cf}	-	48_{ef}	-	56_{ef}	-	-	-	-	-	-	-	-	-	-	-	2_{ef}	2_{cf}	-
30	290845350_{ef}	-	32768_{cf}	2048_{cf}	1600_{cf}	-	120_{cf}	-	120_{cf}	-	30_{cf}	-	15_{cf}	-	-	-	-	-	3_{cf}

Table 10.26: **Best lower bounds for $A_4^{GC}(n, d, w)$ before shortening or puncturing** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	$\mathit{12}_{ef}$	$\mathit{4}^{co}_{cf}$	-	-	-	-	-	-	-
5	20_{cf}	$\mathit{10}_{cf}$	2_{cf}	-	-	-	-	-	-
6	80^{co}_{ca}	$\mathit{40}^{co}_{ef}$	4^{7}_{pr}	$\mathit{4}^{co}_{cf}$	-	-	-	-	-
7	$\mathbf{280}_{ca}$	70_{ca}	14_{ca}	$\mathit{7}_{ca}$	2_{cf}	-	-	-	-
8	560_{ca}	224_{ef}	44_{ca}	$\mathit{28}_{ea}$	3^{G3}_{nb}	$\mathit{4}^{co}_{cf}$	-	-	
9	2016^{10}_{sb}	504_{ca}	132^{co}_{ca}	$\mathbf{42}^{C5}_{nb}$	3^{A1}_{nb}	4^{co}_{ef}	$\mathit{3}_{cf}$	-	-
10	6720^{11}_{sb}	$\mathbf{2016}^{co}_{cf}$	$\mathbf{504}^{co}_{ca}$	$\mathbf{144}^{C2}_{nb}$	12^{co}_{ef}	$\mathit{16}^{co}_{cf}$	3_{ef}	$\mathit{4}^{co}_{cf}$	-
11	29568^{12}_{sb}	$\mathit{7392}^{12}_{sb}$	924^{1}_{sb}	$\mathbf{472}^{A1}_{nb}$	42^{12}_{sb}	16^{co}_{ef}	3^{12}_{sb}	$\mathit{4}^{co}_{ef}$	2_{cf}
12	$\mathit{118272}^{13}_{sb}$	$\mathit{29568}_{ca}$	$\mathbf{3696}_{ca}$	$\mathbf{1888}^{co}_{ca}$	168_{ca}	64^{co}_{ca}	12_{ca}	4^{2A}_{nb}	2^{A2}_{nb}
13	439296^{14}_{sb}	30208^{co}_{ea}	7232^{A2}_{nb}	$\mathbf{3432}_{ca}$	$\mathbf{464}^{A1}_{nb}$	128^{A1}_{nb}	28^{14}_{pr}	11^{14}_{sb}	4^{14}_{pr}
14	$\mathit{1537536}^{15}_{sb}$	$\mathbf{439296}^{co}_{ca}$	$\mathbf{28032}^{A2}_{nb}$	$\mathbf{7424}^{T1}_{nb}$	$\mathbf{1856}^{A1}_{nb}$	$\mathbf{512}^{C1}_{nb}$	64^{C1}_{nb}	$\mathbf{49}_{ca}$	10^{C2}_{nb}
15	$\mathit{6589440}_{cf}$	$\mathit{1647360}_{cf}$	$\mathbf{103680}^{co}_{ca}$	$\mathbf{27840}_{cf}$	$\mathbf{6960}_{cf}$	$\mathbf{1920}^{co}_{cf}$	$\mathbf{240}^{co}_{ca}$	$\mathbf{124}^{co}_{cf}$	31^{co}_{ca}
16	$\mathit{26357760}_{ef}$	$\mathit{6589440}_{ef}$	$\mathbf{414720}^{co}_{ea}$	$\mathit{111360}_{ef}$	$\mathbf{27840}_{ef}$	6480^{co}_{ef}	423^{10}_{sb}	124^{co}_{ef}	104^{co}_{ef}
17	24893440^{17}_{sb}	12446720_{cf}	777920^{18}_{pr}	388960^{18}_{sb}	$\mathit{48620}_{cf}$	$\mathit{24310}_{cf}$	$\mathbf{1530}^{18}_{sb}$	170_{ca}	100^{18}_{pr}
18	89616384^{18}_{sb}	24893440^{co}_{ef}	1400256_{ca}	1400256_{ca}	$\mathbf{97520}^{G8}_{nb}$	$\mathit{87516}_{ef}$	$\mathbf{5508}_{ca}$	414_{ca}	132^{19}_{sb}
19	378380288^{19}_{sb}	22468608_{ea}	5912192^{20}_{sb}	1478048_{ea}	369512^{1}_{sb}	$\mathit{92378}_{cf}$	6138^{1}_{sb}	1056^{20}_{sb}	536^{14}_{sb}
20	1513521152^{20}_{sb}	$\mathbf{378896384}^{co}_{cf}$	23648768^{21}_{sb}	1479168^{20}_{sb}	$\mathit{1478048}_{ca}$	$\mathbf{369512}_{ef}$	$\mathbf{24552}_{ca}$	4224^{21}_{pr}	$\mathbf{2112}^{21}_{sb}$
21	5778898944_{cf}	1443692544^{22}_{sb}	90295296_{cf}	5641216^{21}_{sb}	707168^{21}_{pr}	516096_{cf}	44764^{19}_{pr}	11344^{16}_{sb}	8064_{cf}
22	21204434944^{23}_{sb}	5301108736^{23}_{sb}	90295296_{ef}	20672512_{ef}	2821728^{23}_{pr}	707168^{22}_{pr}	177352^{2}_{sb}	45296^{7}_{pr}	11324^{7}_{sb}
23	88640061440^{24}_{sb}	22160015360^{24}_{sb}	174125056^{24}_{sb}	21883904^{2}_{sb}	5275648_{cf}	2705248^{21}_{sb}	676956^{23}_{pr}	170044^{23}_{pr}	42826^{23}_{sb}
24	354439135232^{25}_{sb}	88609783808^{25}_{sb}	696500224^{13}_{sb}	87005184^{18}_{sb}	5437824^{18}_{sb}	10816624^{25}_{pr}	2704156^{25}_{sb}	676956^{25}_{sb}	170044^{25}_{sb}
25	1362985222144^{26}_{sb}	340746305536^{26}_{sb}	2670166016^{14}_{sb}	333684736^{21}_{sb}	20855296^{21}_{sb}	5244416^{14}_{sb}	10400600^{26}_{pr}	2600612^{26}_{sb}	651028^{25}_{sb}
26	5064454307840^{24}_{sb}	1266113576960^{25}_{sb}	10650386432^{3}_{pb}	1237975040^{23}_{sb}	77373440^{23}_{sb}	19415040^{15}_{sb}	168960^{15}_{pr}	10400600^{26}_{pr}	2599688^{27}_{sb}
27	21034728161280^{26}_{sb}	5258682040320^{24}_{sb}	1026883584^{18}_{sb}	5131714560^{28}_{sb}	320732160^{28}_{sb}	80056320^{18}_{sb}	639232^{26}_{sb}	319616^{26}_{sb}	10028292^{28}_{sb}
28	84130607923200^{28}_{sb}	21032651980800^{28}_{sb}	4104939110^{29}_{sb}	2052469555^{29}_{sb}	1282793472^{29}_{sb}	319975424^{29}_{sb}	2520576^{27}_{sb}	1260288^{27}_{sb}	40116600^{28}_{pr}
29	3252906224517^{1}_{sb}	813226556129^{1}_{sb}	158945771520^{1}_{pr}	79360425984^{1}_{sb}	4960026624^{1}_{sb}	1237057536^{1}_{sb}	9719808^{1}_{pr}	4859904^{1}_{sb}	330848^{1}_{sb}
30	1219947795578880_{cf}	304986948894720_{cf}	161533132800_{cf}	29802332160_{cf}	18626457600_{cf}	4665323520_{cf}	18163200_{cf}	18224640_{cf}	1417920_{cf}

Table 10.27: **Best lower bounds for $A_4^{GC}(n,d,w)$ after shortening or puncturing $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	4^{co}_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	4^{co}_{ef}	2^{1}_{sb}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	7_{ca}	2^{7}_{sb}	4^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	$\mathbf{15}_{cf}$	2^{14}_{sb}	4^{co}_{ef}	3_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	60_{ef}	2^{17}_{sb}	2^{13}_{sb}	3_{ef}	4^{co}_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	100^{co}_{cf}	8^{3}_{sb}	2^{18}_{sb}	3^{14}_{sb}	4^{co}_{ef}	2_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-
18	$\mathbf{180}^{7}_{sb}$	8^{12}_{sb}	8^{12}_{pr}	3^{19}_{sb}	2^{15}_{sb}	2^{18}_{pr}	4^{co}_{cf}	-	-	-	-	-	-	-	-	-	-	-	-
19	180^{14}_{sb}	24^{13}_{pr}	11^{20}_{sb}	12^{13}_{sl}	2^{20}_{sb}	1^{12}_{sb}	3_{ef}	2_{cf}	-	-	-	-	-	-	-	-	-	-	-
20	$\mathbf{520}^{21}_{sb}$	40^{11}_{pr}	44^{6}_{sl}	8^{17}_{sl}	10_{cf}	1^{17}_{sb}	2^{13}_{sb}	2^{20}_{pr}	4^{co}_{ef}	-	-	-	-	-	-	-	-	-	-
21	2240_{cf}	88^{17}_{pr}	32^{22}_{sb}	2^{21}_{sb}	16^{G3}_{nb}	1^{22}_{sb}	2^{18}_{sb}	1^{14}_{sb}	2_{ef}	3_{cf}	-	-	-	-	-	-	-	-	-
22	5856^{T7}_{nb}	88^{18}_{pr}	128^{12}_{pb}	6_{ef}	16^{A12}_{nb}	6^{23}_{pr}	2^{23}_{sb}	1^{19}_{sb}	2^{15}_{sb}	3_{ef}	3^{C19}_{nb}	-	-	-	-	-	-	-	-
23	21434^{23}_{sb}	84^{18}_{pr}	160^{8}_{sl}	24^{13}_{sb}	16^{A1}_{nb}	6^{3}_{sb}	6^{24}_{pr}	1^{24}_{sb}	2^{20}_{sb}	2^{14}_{sb}	3^{C12}_{nb}	2_{cf}	-	-	-	-	-	-	-
24	85652^{25}_{sb}	320^{21}_{pb}	128^{8}_{sl}	24^{25}_{sb}	64^{C17}_{nb}	6^{16}_{sb}	6^{11}_{sb}	6^{25}_{pr}	2^{25}_{sb}	2^{21}_{sb}	3^{C8}_{nb}	2^{9}_{sb}	3_{cf}	-	-	-	-	-	-
25	326088^{26}_{sb}	512^{26}_{sb}	256^{A1}_{nb}	96^{5}_{sb}	64^{C5}_{nb}	12^{13}_{sb}	6^{24}_{sb}	6^{13}_{sb}	10_{cf}	2^{25}_{sb}	3^{A26}_{nb}	2^{14}_{sb}	3_{ef}	2_{cf}	-	-	-	-	-
26	1208316^{27}_{sb}	2048^{14}_{sb}	1024^{A14}_{nb}	96^{19}_{sb}	144^{10}_{sb}	12^{26}_{sb}	20^{10}_{sb}	6^{26}_{sb}	10_{ef}	2^{15}_{pr}	3^{A13}_{nl}	2^{19}_{sb}	3^{15}_{sb}	-	2_{cf}	-	-	-	-
27	5013288^{28}_{sb}	4736^{28}_{pr}	2368^{28}_{sb}	288^{24}_{pr}	144^{24}_{sb}	40^{24}_{pr}	20^{24}_{sb}	20^{24}_{pr}	10^{24}_{sb}	5^{G12}_{nb}	2^{28}_{sb}	2^{24}_{sb}	3^{20}_{sb}	-	2_{ef}	3_{cf}	-	-	-
28	20056584^{28}_{sb}	4736^{29}_{sb}	9472^{15}_{sl}	288^{25}_{sb}	480^{15}_{sb}	20^{29}_{sb}	64^{15}_{pr}	20^{25}_{sb}	30^{15}_{sb}	5^{27}_{sb}	8^{15}_{sb}	2^{29}_{sb}	3^{25}_{sb}	-	-	3_{ef}	2_{cf}	-	-
29	77558760_{cf}	18944^{1}_{pr}	8192^{1}_{sb}	960^{1}_{pr}	480^{1}_{sb}	72^{1}_{pr}	36^{1}_{sb}	64^{1}_{pr}	30^{1}_{sb}	18^{1}_{pr}	8^{1}_{sb}	8^{1}_{pr}	3^{1}_{sb}	-	-	-	-	2_{cf}	-
30	290845350_{ef}	-	32768_{cf}	2048_{cf}	1600_{cf}	-	120_{cf}	-	120_{cf}	-	30_{cf}	-	15_{cf}	-	-	-	-	-	3_{cf}

Table 10.28: **Best lower bounds for $A_4^{GC}(n,d,w)$ after shortening or puncturing** $12 \leq d \leq 30$

(n,d)	generator polynomials
(30,3)	$g(x) = x^4 + x^3 + \omega^2 x^2 + x + \omega$
(30,4)	$g(x) = x^5 + \omega^2 x^4 + x^3 + \omega^2$
(30,5)	$g(x) = x^{10} + \omega x^9 + x^8 + x^4 + \omega x^3 + \omega x + 1$
(30,6)	$g(x) = x^{10} + \omega^2 x^9 + x^6 + x^5 + \omega^2 x^4 + \omega x^3 + x^2 + \omega$
(30,7)	$g(x) = x^{12} + \omega^2 x^{11} + \omega^2 x^{10} + \omega x^9 + x^8 + x^7 + x^5 + \omega^2 x^4 + x^3 + x^2 + 1$
(30,8)	$g(x) = x^{13} + x^{11} + \omega x^8 + \omega^2 x^7 + x^6 + \omega^2 x^4 + \omega x^3 + \omega^2 x^2 + x + \omega^2$
(30,9)	$g(x) = x^{17} + x^{14} + \omega x^{13} + x^{12} + x^{11} + \omega^2 x^8 + \omega x^7 + \omega x^5 + x^4 + x^3 + \omega x^2 + x + \omega$
(30,10)	$g(x) = x^{17} + \omega x^{15} + \omega^2 x^{13} + x^{12} + \omega^2 x^{11} + \omega^2 x^{10} + \omega x^9 + x^8 + x^7 + x^6 + x^5 + \omega x^4 + \omega^2 x^3 + \omega x^2 + x + 1$
(30,11)	$g(x) = x^{19} + \omega^2 x^{18} + \omega x^{17} + x^{16} + x^{15} + \omega x^{14} + x^{13} + x^{12} + \omega x^{11} + \omega^2 x^{10} + x^7 + x^5 + \omega^2 x^4 + \omega x^3 + \omega x^2 + \omega^2 x + \omega^2$
(30,12)	$g(x) = x^{14} + \omega x^{13} + \omega x^{11} + \omega^2 x^{10} + x^9 + \omega^2 x^8 + \omega x^7 + \omega^2 x^6 + x^5 + \omega^2 x^4 + \omega x^3 + \omega x + 1$
(30,14)	$g(x) = x^{22} + \omega x^{21} + \omega x^{19} + x^{18} + x^{16} + \omega^2 x^{15} + \omega x^{14} + \omega x^{13} + \omega x^{12} + \omega x^{11} + \omega x^{10} + \omega x^9 + \omega x^8 + \omega^2 x^7 + \omega x^5 + \omega^2 x^3 + \omega x^2 + \omega^2 x + 1$
(30,15)	$g(x) = x^{24} + x^{22} + x^{21} + \omega x^{20} + x^{19} + \omega^2 x^{18} + x^{17} + x^{16} + \omega x^{14} + \omega x^{13} + \omega x^{12} + \omega x^{11} + \omega x^{10} + \omega^2 x^9 + \omega x^8 + \omega^2 x^7 + \omega^2 x^6 + \omega^2 x^4 + x^3 + \omega^2 x^2 + \omega^2 x + \omega$
(30,16)	$g(x) = x^{24} + \omega x^{23} + x^{22} + x^{20} + \omega x^{19} + \omega^2 x^{17} + \omega^2 x^{16} + x^{15} + x^9 + \omega x^8 + x^7 + x^5 + \omega x^4 + \omega^2 x^2 + \omega^2 x + 1$
(30,18)	$g(x) = x^{26} + \omega x^{25} + \omega x^{24} + x^{23} + \omega^2 x^{22} + x^{21} + \omega x^{19} + x^{18} + \omega^2 x^{17} + \omega x^{15} + x^{11} + \omega x^{10} + \omega x^9 + x^8 + \omega^2 x^7 + x^6 + \omega x^4 + x^3 + \omega^2 x^2 + \omega$
(30,20)	$g(x) = x^{26} + \omega x^{25} + \omega^2 x^{23} + \omega^2 x^{22} + x^{21} + x^{20} + \omega^2 x^{19} + \omega^2 x^{18} + \omega x^{16} + x^{15} + x^{11} + \omega x^{10} + \omega^2 x^8 + \omega^2 x^7 + x^6 + x^5 + \omega^2 x^4 + \omega^2 x^3 + \omega x + 1$
(30,22)	$g(x) = x^{27} + x^{25} + \omega x^{24} + x^{23} + \omega x^{21} + \omega x^{20} + \omega x^{19} + x^{18} + x^{17} + \omega x^{16} + \omega^2 x^{15} + x^{12} + x^{10} + \omega x^9 + x^8 + \omega x^6 + \omega x^5 + \omega x^4 + x^3 + x^2 + \omega x + \omega^2$
(30,24)	$g(x) = x^{28} + \omega^2 x^{27} + x^{26} + x^{25} + \omega^2 x^{23} + \omega x^{22} + \omega^2 x^{21} + \omega^2 x^{20} + \omega x^{18} + x^{17} + \omega x^{16} + \omega x^{15} + x^{13} + \omega^2 x^{12} + x^{11} + x^{10} + \omega^2 x^8 + \omega x^7 + \omega^2 x^6 + \omega^2 x^5 + \omega x^3 + x^2 + \omega x + \omega$
(29,12)	$g(x) = x^{15} + \omega^2 x^{14} + \omega x^{13} + \omega x^{12} + x^{11} + \omega x^{10} + \omega x^9 + x^8 + x^7 + \omega x^6 + \omega x^5 + x^4 + \omega x^3 + \omega x^2 + \omega^2 x + 1$
(26,20)	$g(x) = x^{23} + \omega^2 x^{22} + \omega^2 x^{21} + x^{20} + x^{18} + \omega^2 x^{17} + \omega^2 x^{16} + x^{15} + x^{13} + \omega^2 x^{12} + \omega^2 x^{11} + x^{10} + x^8 + \omega^2 x^7 + \omega^2 x^6 + x^5 + x^3 + \omega^2 x^2 + \omega^2 x + 1$
(25,20)	$g(x) = x^{23} + \omega^2 x^{22} + \omega^2 x^{21} + x^{20} + x^{18} + \omega^2 x^{17} + \omega^2 x^{16} + x^{15} + x^{13} + \omega^2 x^{12} + \omega^2 x^{11} + x^{10} + x^8 + \omega^2 x^7 + \omega^2 x^6 + x^5 + x^3 + \omega^2 x^2 + \omega^2 x + 1$
(23,7)	$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$
(22,5)	$g(x) = x^7 + \omega^2 x^6 + x^4 + x^3 + \omega^2 x + 1$
(22,6)	$g(x) = x^8 + \omega^2 x^7 + \omega x^6 + x^5 + x^4 + x^3 + \omega x^2 + 1$
(22,15)	$g(x) = x^{19} + \omega x^{18} + x^{16} + \omega x^{15} + x^{13} + \omega x^{12} + x^{10} + \omega x^9 + x^7 + \omega x^6 + x^4 + \omega x^3 + x + \omega$
(21,3)	$g(x) = x^4 + x^3 + \omega x^2 + \omega^2 x + 1$
(21,5)	$g(x) = x^7 + \omega^2 x^6 + x^4 + x^3 + \omega^2 x + 1$
(21,8)	$g(x) = x^{11} + \omega x^{10} + x^8 + \omega^2 x^7 + \omega x^6 + x^5 + \omega x^4 + x^3 + \omega x^2 + x + \omega$
(21,11)	$g(x) = x^{14} + \omega x^{13} + \omega^2 x^{12} + x^{10} + x^8 + x^7 + \omega x^6 + x^4 + \omega x^2 + \omega x + \omega$
(21,12)	$g(x) = x^{15} + x^{14} + \omega x^{13} + \omega x^{12} + x^{11} + \omega^2 x^{10} + x^9 + \omega x^8 + x^7 + x^6 + x^5 + \omega^2 x^4 + \omega x^3 + \omega^2 x^2 + \omega^2 x + 1$

Table 10.29: **Generator polynomials for codes with more than 4 codewords,** $21 < n \leq 30$ **.**

(n,d)	generator polynomials and coset leaders $\mathbf{L}$
$(20,4)$	$g(x) = x^5+\omega^2x^4+\omega x^3+\omega x^2+\omega^2x+1$
	$\mathbf{L} = (1,\omega^2,1,\omega,\omega,\omega,1,\omega^2,1,\omega^2,1,\omega^2,\omega^2,0,\omega,0,1,\omega^2,\omega,1)$
$(20,7)$	$\omega p(x)+q(x) = x^{13}+x^{10}+x^7+x^5+x^4+x^2+\omega x+\omega^2$
	$r(x) = x^{17}+x^{16}+x^{13}+x^{12}+x^9+x^8+x^5+x^4+x+1$
$(20,8)$	$g(x) = x^9+\omega x^8+\omega x^6+\omega x^5+\omega^2x^4+\omega^2x^3+\omega^2x+1$
$(20,9)$	$\omega p(x)+q(x) = x^{13}+x^{11}+x^9+x^7+x^6+\omega^2x^4+\omega^2x^3+\omega^2x^2+\omega x+\omega$
	$r(x) = x^{20}+1$
$(20,16)$	$g(x) = x^{18}+\omega^2x^{17}+\omega^2x^{16}+x^{15}+x^{13}+\omega^2x^{12}+\omega^2x^{11}+x^{10}+x^8+\omega^2x^7+\omega^2x^6$
	$+x^5+x^3+\omega^2x^2+\omega^2x+1$
$(19,4)$	$\omega p(x)+q(x) = x^7+x^5+x^4+x^3+\omega x^2+x+\omega^2$
	$r(x) = x^8+x^6+x^5+x^3+x^2+1$
$(19,6)$	$\omega p(x)+q(x) = x^{10}+x^8+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x) = x^{13}+x^{12}+x^7+x^6+x+1$
$(19,8)$	$g(x) = x^{10}+\omega x^9+\omega^2x^8+\omega^2x^7+x^5+\omega x^3+\omega x^2+\omega^2x+1$
$(18,4)$	$g(x) = x^4+x^3+\omega x^2+x+1$
	$\mathbf{L} = (1,1,1,0,\omega^2,1,0,1,\omega,1,0,\omega^2,1,\omega^2,1,\omega,\omega,\omega^2)$
$(18,5)$	$\omega p(x)+q(x) = x^{12}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x) = x^{13}+x^{12}+x^7+x^6+x+1$
$(18,6)$	$\omega p(x)+q(x) = x^{10}+x^5+x^4+x^3+\omega^2x+\omega^2$
	$r(x) = x^{13}+x^{12}+x^7+x^6+x+1$
$(18,8)$	$g(x) = x^8+\omega^2x^7+x^6+x^5+\omega x^4+x^3+x^2+\omega^2x+1$
$(18,9)$	$\omega p(x)+q(x) = x^{17}+x^{16}+x^{15}+x^{12}+x^{11}+x^{10}+x^9+x^7+x^6+x^5+\omega x^4+\omega^2x^3+x^2+\omega^2x+\omega^2$
	$r(x) = x^{18}+1$
$(18,10)$	$\omega p(x)+q(x) = x^{17}+x^{12}+x^9+\omega x^8+x^7+\omega x^6+\omega^2x^5+x^4+\omega^2x^3+\omega^2x^2+x+\omega^2$
	$r(x) = x^{18}+1$
$(17,4)$	$g(x) = x^4+\omega^2x^3+x^2+\omega^2x+1$
$(17,7)$	$g(x) = x^8+\omega^2x^7+\omega^2x^5+\omega^2x^4+\omega^2x^3+\omega^2x+1$
$(17,8)$	$g(x) = x^9+\omega^2x^8+\omega x^7+\omega x^6+\omega x^3+\omega x^2+\omega^2x+1$
$(17,10)$	$\omega p(x)+q(x) = x^{12}+x^{10}+\omega x^9+\omega x^8+\omega x^6+x^5+x^4+\omega^2x^3+x^2+\omega^2x+\omega^2$
	$r(x) = x^{16}+x^{15}+x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3$
	$+x^2+x+1$
$(17,12)$	$g(x) = x^{13}+\omega^2x^{12}+\omega^2x^{11}+\omega x^{10}+x^9+\omega^2x^7+\omega^2x^6+x^4+\omega x^3+\omega^2x^2+\omega^2x+1$
	$\mathbf{L} = (0,\omega^2,0,\omega,0,\omega,\omega,\omega^2,\omega^2,1,0,1,1,\omega^2,1,0,1)$
$(16,3)$	$g(x) = x^2+x+\omega$
$(16,4)$	$g(x) = x^3+\omega^2x^2+\omega^2$
$(16,5)$	$\omega p(x)+q(x) = x^6+x^5+x^3+\omega x^2+\omega^2x+\omega^2$
	$r(x) = x^8+x^7+x^6+x^4+1$
	$\mathbf{L} = (\omega,\omega^2,\omega^2,0,\omega^2,0,0,\omega^2,1,0,0,1,1,\omega^2,\omega,\omega^2)$
$(16,6)$	$g(x) = x^6+\omega x^5+x^4+x^3+\omega^2x^2+\omega^2x+1$
$(16,7)$	$g(x) = x^7+\omega x^5+\omega^2x^4+x^3+\omega x^2+\omega$
$(16,8)$	$g(x) = x^8+\omega x^6+\omega x^5+x^4+\omega x^3+\omega x^2+1$
	$\mathbf{L} = (\omega,1,0,\omega^2,\omega,\omega^2,0,1,\omega,1,\omega,0,0,\omega,\omega,\omega)$
$(16,10)$	$g(x) = x^{11}+x^{10}+\omega x^9+x^8+\omega^2x^7+\omega^2x^6+x^4+\omega^2x^3+\omega x^2+\omega$
	$\mathbf{L} = (1,\omega,\omega^2,1,1,0,\omega^2,\omega,\omega^2,\omega^2,0,1,\omega^2,\omega^2,1,1)$
$(16,11)$	$g(x) = x^{11}+x^{10}+\omega x^8+\omega^2x^7+\omega^2x^6+x^5+\omega^2x^4+x^3+x+\omega^2$
	$\mathbf{L} = (\omega^2,\omega^2,\omega,1,1,1,\omega,\omega^2,1,\omega^2,0,0,\omega^2,\omega,0,1)$
$(16,12)$	$g(x) = x^{12}+\omega x^{11}+\omega^2x^{10}+\omega x^9+\omega x^8+x^7+\omega^2x^6+\omega^2x^4+\omega^2x^3+x^2+\omega$
$(15,3)$	$g(x) = x^3+\omega x^2+\omega^2$
$(15,4)$	$g(x) = x^4+\omega x^3+\omega^2x^2+\omega^2x+\omega^2$

Table 10.30: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $15 \leq n \leq 20$ **.**

(n,d)	generator polynomials and coset leaders **L**
$(15,5)$	$\omega p(x)+q(x) = x^4+\omega x^3+x^2+x+\omega^2$
	$r(x) = x^9+x^7+x^6+x^3+x^2+1$
	$\mathbf{L} = (0,1,1,\omega,\omega^2,0,\omega,1,0,\omega,\omega^2,0,\omega,\omega,\omega)$
$(15,6)$	$g(x) = x^7+\omega x^6+\omega x^4+\omega^2 x^3+\omega x^2+\omega^2 x+\omega^2$
$(15,7)$	$g(x) = x^8+\omega^2 x^6+\omega x^5+\omega^2 x^4+\omega^2 x^3+\omega x^2+\omega$
$(15,8)$	$g(x) = x^9+\omega^2 x^8+\omega^2 x^7+\omega x^5+x^4+x^2+\omega x+1$
	$\mathbf{L} = (1,1,\omega^2,1,0,1,\omega,\omega,\omega^2,1,\omega,\omega,0,1,1)$
$(15,9)$	$\omega p(x)+q(x) = x^{11}+x^{10}+x^9+\omega x^7+x^6+\omega^2 x^3+x^2+\omega^2 x+\omega^2$
	$r(x) = x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	$\mathbf{L} = (1,1,0,1,1,\omega^2,\omega,0,\omega,\omega^2,\omega,0,\omega,\omega^2,1)$
$(15,10)$	$g(x) = x^{11}+\omega^2 x^{10}+\omega x^8+\omega x^7+x^6+x^5+\omega x^4+\omega x^3+\omega^2 x+1$
	$\mathbf{L} = (\omega^2,\omega^2,\omega^2,\omega^2,\omega,0,\omega,\omega,0,\omega^2,1,1,\omega,\omega,0)$
$(15,11)$	$\omega p(x)+q(x) = x^{14}+x^{13}+x^{12}+\omega^2 x^9+\omega x^7+\omega x^6+x^5+\omega x^3+\omega^2 x^2+x+\omega^2$
	$r(x) = x^{15}+1$
	$\mathbf{L} = (1,\omega^2,\omega^2,\omega^2,\omega^2,\omega^2,1,1,0,\omega,1,0,0,\omega,0)$
$(15,12)$	$g(x) = x^{13}+\omega^2 x^{12}+x^{11}+x^{10}+\omega^2 x^8+\omega x^7+\omega^2 x^6+\omega^2 x^5+\omega x^3+x^2+\omega x+\omega$
$(14,4)$	$\omega p(x)+q(x) = x^4+\omega^2 x^3+x^2+\omega x+\omega$
	$r(x) = x^5+x^2+x+1$
	$\mathbf{L} = (0,0,0,1,1,1,1,\omega^2,\omega,0,\omega,0,\omega,0)$
$(14,10)$	$\omega p(x)+q(x) = x^{11}+x^{10}+\omega x^8+\omega x^6+x^5+\omega^2 x^4+x^3+x^2+x+\omega^2$
	$r(x) = x^{14}+1$
$(14,12)$	$\omega p(x)+q(x) = x^{12}+\omega x^{11}+\omega x^{10}+\omega^2 x^9+x^8+\omega^2 x^7+x^5+\omega x^4+\omega x^3+\omega^2 x^2+x+\omega^2$
	$r(x) = x^{14}+1$
$(13,4)$	$\omega p(x)+q(x) = x^4+x^3+\omega^2 x^2+\omega^2$
	$r(x) = x^6+x^5+x^3+x+1$
	$\mathbf{L} = (\omega^2,\omega,\omega^2,1,\omega^2,1,1,0,0,\omega^2,1,\omega,\omega^2)$
$(13,6)$	$\omega p(x)+q(x) = x^{11}+x^8+x^3+x^2+\omega^2 x+\omega^2$
	$r(x) = x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
$(12,4)$	$\omega p(x)+q(x) = x^6+x^3+x^2+\omega^2 x+\omega^2$
	$r(x) = x^7+x^6+x+1$
$(12,5)$	$\omega p(x)+q(x) = x^9+x^7+x^6+x^5+x^4+x^3+x^2+\omega^2 x+\omega^2$
	$r(x) = x^{10}+x^8+x^6+x^4+x^2+1$
$(12,6)$	$\omega p(x)+q(x) = x^9+x^7+x^5+\omega^2 x^2+\omega x+\omega^2$
	$r(x) = x^{10}+x^9+x^7+x^6+x^4+x^3+x+1$
	$\mathbf{L} = (0,0,0,0,\omega^2,\omega,0,0,0,1,\omega,1)$
$(12,7)$	$\omega p(x)+q(x) = x^{10}+x^9+x^7+\omega x^5+\omega x^4+\omega^2 x^3+\omega^2 x^2+\omega^2 x+\omega^2$
	$r(x) = x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
$(12,8)$	$\omega p(x)+q(x) = x^{11}+x^7+\omega^2 x^6+\omega x^5+\omega^2 x^4+\omega^2 x^2+\omega x+\omega^2$
	$r(x) = x^{12}+1$
	$\mathbf{L} = (1,\omega^2,\omega^2,\omega,\omega^2,\omega^2,0,1,1,\omega^2,1,\omega)$
$(12,9)$	$\omega p(x)+q(x) = x^{10}+\omega^2 x^9+\omega^2 x^8+x^6+\omega^2 x^5+\omega^2 x^4+x^2+\omega^2 x+\omega^2$
	$r(x) = x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
$(11,8)$	$g(x) = x^8+\omega^2 x^7+\omega^2 x^6+x^5+x^3+\omega^2 x^2+\omega^2 x+1$
	$\mathbf{L} = (1,\omega,1,\omega,1,0,0,0,\omega^2,\omega^2,1)$
$(10,4)$	$g(x) = x^4+\omega x^3+\omega x+1$
	$\mathbf{L} = (\omega^2,\omega,\omega,1,\omega^2,0,\omega^2,0,0,\omega^2)$
$(10,5)$	$\omega p(x)+q(x) = x^8+x^6+x^5+x^4+x^3+x^2+\omega^2 x+\omega^2$
	$r(x) = x^{10}+1$
	$\mathbf{L} = (0,0,1,1,1,1,0,1,0,0)$
$(10,7)$	$g(x) = x^7+\omega x^6+x^4+\omega x^3+x+\omega$
	$\mathbf{L} = (0,\omega,\omega,1,0,\omega,\omega^2,0,0,\omega)$

Table 10.31: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $10 \leq n \leq 15$ **.**

(n,d)	generator polynomials and coset leaders **L**
$(10,8)$	$g(x) = x^8 + \omega^2 x^7 + \omega^2 x^6 + x^5 + x^3 + \omega^2 x^2 + \omega^2 x + 1$
	$\mathbf{L} = (1, \omega, 1, 0, 0, 0, \omega^2, 0, \omega, \omega)$
$(9,4)$	$\omega p(x) + q(x) = x^6 + x^4 + \omega^2 x + \omega^2$
	$r(x) = x^7 + x^6 + x^4 + x^3 + x + 1$
$(9,5)$	$\omega p(x) + q(x) = x^6 + x^5 + x^4 + \omega x^3 + x^2 + x + \omega^2$
	$r(x) = x^7 + x^6 + x^4 + x^3 + x + 1$
	$\mathbf{L} = (0, 1, 0, \omega, 0, 0, \omega^2, 0, \omega)$
$(8,3)$	$\omega p(x) + q(x) = x^2 + \omega^2 x + \omega^2$
	$r(x) = x^5 + x^4 + x + 1$
$(8,4)$	$g(x) = x^3 + x + 1$
$(8,5)$	$\omega p(x) + q(x) = x^5 + \omega x^3 + \omega^2 x^2 + \omega^2 x + \omega^2$
	$r(x) = x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
$(8,6)$	$\omega p(x) + q(x) = x^5 + \omega x^3 + x^2 + \omega^2 x + \omega^2$
	$r(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
$(7,3)$	$\omega p(x) + q(x) = x^3 + x^2 + \omega^2 x + \omega^2$
	$r(x) = x^4 + x^3 + x^2 + 1$
$(7,4)$	$\omega p(x) + q(x) = x^5 + x^3 + x^2 + \omega^2 x + \omega^2$
	$r(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
$(7,5)$	$\omega p(x) + q(x) = x^5 + \omega x^4 + \omega x^3 + \omega^2 x^2 + x + \omega^2$
	$r(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
$(7,6)$	$\omega p(x) + q(x) = x^5 + \omega x^4 + \omega x^3 + \omega^2 x^2 + x + \omega^2$
	$r(x) = x^7 + 1$
$(6,3)$	$\omega p(x) + q(x) = x^3 + \omega^2 x + \omega^2$
	$r(x) = x^4 + x^2 + 1$
	$\mathbf{L} = (\omega^2, 0, \omega^2, \omega, \omega^2, \omega^2)$
$(6,4)$	$g(x) = x^2 + \omega x + 1$
	$\mathbf{L} = (1, 0, \omega^2, \omega^2, 0, 0)$
$(5,3)$	$g(x) = x^2 + \omega^2 x + 1$
$(5,4)$	$g(x) = x^3 + \omega x^2 + \omega x + 1$
$(4,3)$	$g(x) = x + \omega$

Table 10.32: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $4 \leq n \leq 10$ **.**

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	4	*3*	-	-	-	-	-	-	-
6	16	5	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	20	-	-	-	-	-	-	-
9	5	-	-	3	-	-	-	-	-
10	504	**860**	64	8	-	4	-	-	-
11	-	-	-	-	-	-	-	-	-
12	14624	4032	-	80	-	5	-	-	-
13	-	-	848	434	-	-	-	-	-
14	12800	51664	-	256	4	22	-	-	-
15	205780	201480	12835	6435	780	**394**	4	30	-
16	-	4432	-	-	-	20	-	-	-
17	-	3111120	10840	5440	12120	*6060*	40	-	-
18	11230272	211744	-	3240	-	-	16	-	-
19	-	-	-	-	-	-	-	-	-
20	23646752	188416000	98056	48952	-	11520	-	368	-
21	45145632-tr	2623072	22573824	1406496	10179	-	5103	-	634
22	-	165423104	1410864	646184	-	-	-	4096	4
23	-	-	-	-	-	-	-	-	-
24	11076104704	2784697344	-	815488	-	4032	-	-	-
25	-	2599688	-	-	-	-	-	-	-
26	-	9891187072	20801200	9655984	-	-	-	16384	-
27	-	-	-	8064	-	-	-	-	-
28	21768306688	2625500086272	-	8106752	3008	3853632	-	-	-
29	-	-	-	-	-	-	-	-	38777664
30	609973884610560	152493461268480	80766566400	149011451520	9313176480	2332609440	9080016	9110544	708168

Table 10.33: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cyclic codes over GF(4) for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	8	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	331	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	80	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	848	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-
28	1536	-	20	-	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	19388832	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	281928	-	16384	1024	784	-	48	-	58	-	12	-	6	-	-	-	-	-	-

Table 10.34: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cyclic codes over GF(4) for $12 \leq d \leq 30$**

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	35	11	-	-	-	-	-	-	-
8	-	108	-	-	-	-	-	-	-
9	-	20	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	860	-	42	-	4	-	-	-
12	-	-	-	*924*	-	-	-	-	-
13	27456	7104	-	80	-	5	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	51664	-	184	-	22	-	-	-
16	*13174400*	*3293600*	-	*55376*	**13808**	2560	-	-	48
17	-	4432	-	-	-	20	-	-	-
18	-	-	-	-	-	-	-	-	-
19	11230272	211744	-	3240	20	-	12	8	-
20	-	-	-	-	-	*184756*	-	-	-
21	-	188416000	-	90064	-	11520	-	368	-
22	-	-	-	-	-	-	-	-	-
23	-	165423104	-	1352078	-	-	-	2510	-
24	-	-	-	-	-	-	-	-	-
25	21300369664	5328918784	-	815488	-	4000	-	-	-
26	-	-	-	-	-	-	-	-	-
27	-	9891187072	-	20057442	-	-	-	9908	-
28	-	-	-	-	-	-	-	-	-
29	-	2625500086272	-	6691200	-	3853632	-	-	-
30	-	-	-	-	-	-	-	-	-

Table 10.35: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from extended cyclic codes over GF(4) for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	*24*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	8	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	80	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	848	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	1536	-	20	-	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.36: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from extended cyclic codes over GF(4) for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	4	3	-	-	-	-	-	-	-
6	16	5	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	24	-	-	-	-	-	-	-
9	5	-	-	3	-	-	-	-	-
10	504	860	64	12	-	4	-	-	-
11	-	-	-	-	-	-	-	-	-
12	14784	4352	-	80	-	6	-	-	-
13	-	-	848	434	-	-	-	-	-
14	12800	51664	-	256	4	22	-	-	-
15	205780	207200	12835	6435	800	**406**	4	31	-
16	-	8064	-	-	-	24	-	-	-
17	-	3111120	12768	6400	12120	*6060*	48	-	-
18	11230272	211744	-	3240	-	-	12	-	-
19	-	-	-	-	-	-	-	-	-
20	23646752	**189432064**	98056	48952	-	12452	-	368	-
21	45145632	2885152	22572816	1412068	11169	-	5601	-	697
22	-	165423104	1410864	646184	-	-	-	4096	4
23	-	-	-	-	-	-	-	-	-
24	11076104704	2784697344	-	809216	-	4672	-	-	-
25	-	20800276	-	-	-	-	-	-	-
26	-	9891187072	20801200	9655984	-	-	-	16384	-
27	-	-	-	5376	-	-	-	-	-
28	20546738176	-	-	7518464	2272	2768896	-	-	-
29	-	-	-	-	-	-	-	-	38777664
30	-	-	79386640384	148913258496	9313176480	2332609440	9053760	9077760	569600

Table 10.37: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of cyclic codes over GF(4) for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	16	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	364	-	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	64	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	768	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	1280	-	20	-	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	19388832	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	145664	-	15	1024	1112	-	48	-	63	-	12	-	5	-	-	-	-	-	-

Table 10.38: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of cyclic codes over GF(4) for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	35	11	-	-	-	-	-	-	-
8	-	108	-	-	-	-	-	-	-
9	-	24	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	860	-	42	-	4	-	-	-
12	-	-	-	*924*	-	-	-	-	-
13	27456	7104	-	80	-	6	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	51664	-	184	-	22	-	-	-
16	-	-	-	*55376*	**13856**	**3776**	-	-	68
17	-	4432	-	-	-	24	-	-	-
18	-	-	-	-	-	-	-	-	-
19	5615136	211744	-	3240	35	-	12	11	-
20	-	-	-	-	-	*184756*	-	-	-
21	-	94716032	-	90064	-	12452	-	368	-
22	-	-	-	-	-	-	-	-	-
23	-	-	-	1351847	-	-	-	2510	-
24	-	-	-	-	-	-	-	-	-
25	-	5328918784	-	407744	-	6080	-	-	-
26	-	-	-	-	-	-	-	-	-
27	-	-	-	20057442	-	-	-	9908	-
28	-	-	-	-	-	-	-	-	-
29	-	-	-	6003840	-	1384448	-	-	-
30	-	-	-	-	-	-	-	-	-

Table 10.39: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of extended cyclic codes over GF(4) for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	16	-	-	-	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	96	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	752	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	192	-	20	-	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.40: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of extended cyclic codes over GF(4) for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	*6*	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	27	*16*	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	16	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	256	88	16	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
9	246	126	12	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	800	**860**	**210**	50	-	4	-	1	-	-	-	-	-	-	-	-	-	-
11	226	226	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	**59136**	**14784**	**1848**	*924*	84	24	6	-	-	-	-	-	-	-	-	-	-	-
13	848	848	848	434	-	-	-	-	-									
14	387744	*192192*	7424	**3192**	**784**	84	-	21	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	819200	819200	51200	51200	6464	832	-	16	-	-	-	-	-	-	-	-	-	-
17	-	3111120	-	-	-	6060	-	-	-	22	-	-	-	-	-	-	-	-
18	11218176	**11232288**	**699624**	**699624**	**43632**	**10890**	**2691**	144	-	16	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	738520	738520	**369008**	98056	6012	1440	32	-	-	-	-	-	-	-	-	-

Table 10.41: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cyclic additive codes over GF(4) pairing 0 with 1 for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	3	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	30	*16*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	10	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	128	80	16	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
9	126	126	12	7	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	800	**860**	126	50	-	4	-	1	-	-	-	-	-	-	-	-	-	-
11	126	126	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	29408	7352	924	904	84	24	6	-	-	-	-	-	-	-	-	-	-	-
13	462	462	434	434	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	387744	191912	7424	**3192**	**784**	84	-	21	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	409600	409600	25600	25600	6464	832	-	16	-	6	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	11218176	**11232288**	**699624**	**699624**	**43632**	**10890**	**2691**	144	-	16	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	738520	738520	368504	92252	6012	1440	-	32	-	-	-	-	-	-	-	-

Table 10.42: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cyclic additive codes over GF(4) pairing 1 with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	4	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	15	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	108	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	128	84	16	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	800	860	110	42	-	4	-	1	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	30208	7552	944	472	64	16	4	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	387744	213584	6680	6624	784	149	-	21	-	-	-	-	-	-	-	-	-	-
16	-	-	-	55376	13808	-	-	-	48	24	-	-	-	-	-	-	-	-
17	409600	409600	25600	25600	6464	832	-	16	-	6	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	5609088	5616144	350568	350568	43758	**11542**	2863	155	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.43: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from extended cyclic additive codes over GF(4) pairing 0 with 1 for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	4	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	32	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	108	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	240	120	16	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	800	860	226	42	-	4	-	1	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	54752	7552	1696	472	64	16	4	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	411560	213584	6680	6624	784	149	-	21	-	-	-	-	-	-	-	-	-	-
16	-	221600	-	55376	13808	2560	-	-	48	24	-	-	-	-	-	-	-	-
17	775680	775680	48480	48480	12120	832	-	16	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	5609088	5616144	**738772**	**738772**	**92252**	**11542**	1458	155	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.44: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from extended cyclic additive codes over GF(4) pairing 1 with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	*6*	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	4	*3*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	27	*16*	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	16	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	280	88	22	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
9	246	126	15	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	800	**860**	**210**	50	-	4	-	1	-	-	-	-	-	-	-	-	-	-
11	226	226	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	**59136**	**14784**	**1848**	*924*	84	24	4	-	-	-	-	-	-	-	-	-	-	-
13	848	848	848	434	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	387744	*192192*	7424	**3216**	**796**	84	-	**21**	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	823680	823680	51480	51480	6540	902	-	20	-	26	-	-	-	-	-	-	-	-
17	-	-	-	-	-	6060	-	-	-	16	-	-	-	-	-	-	-	-
18	11218176	**11232288**	**699624**	**699624**	**43632**	**10890**	**2691**	144	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	738520	738520	**369008**	92816	6012	1440	-	32	-	-	-	-	-	-	-	-

Table 10.45: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of cyclic additive codes over GF(4) pairing 0 with 1 for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	2	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	3	*3*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	20	*16*	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	10	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	152	88	22	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
9	126	126	15	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	800	**860**	126	50	-	4	-	1	-	-	-	-	-	-	-	-	-	-
11	126	126	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	29408	7352	904	452	44	8	2	-	-	-	-	-	-	-	-	-	-	-
13	462	462	434	434	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	387744	191912	7424	**3216**	**796**	84	-	12	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	414080	414080	25880	25880	6476	902	-	24	-	6	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	11218176	**11232288**	**699120**	**699120**	**43632**	**6160**	**2691**	144	-	16	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	738520	738520	**368504**	92252	6012	1440	-	32	-	-	-	-	-	-	-	-

Table 10.46: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of cyclic additive codes over GF(4) pairing** 1 **with** ω **for** $3 \leq d \leq 20$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	4	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	22	22	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	108	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	152	88	22	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	800	860	110	50	-	4	-	1	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	30208	7552	944	880	64	16	4	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	387744	213632	6680	**6648**	796	98	-	12	-	-	-	-	-	-	-	-	-	-
16	-	-	-	55376	13856	-	-	-	68	24	-	-	-	-	-	-	-	-
17	414080	414080	25880	25880	6540	902	-	20	-	6	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	5609088	5609088	351072	351072	43758	**11542**	2893	161	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.47: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of extended additive cyclic codes over GF(4) pairing** 0 **with** 1 **for** $3 \leq d \leq 20$

n/d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	4	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	32	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	108	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	240	120	22	6	-	1	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	800	860	226	50	-	4	-	1	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	54752	7552	1696	472	64	16	4	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	411560	213632	6680	**6648**	796	98	-	21	-	-	-	-	-	-	-	-	-	-
16	-	221600	-	55424	13856	2560	-	-	68	24	-	-	-	-	-	-	-	-
17	775680	775680	48480	48480	12120	902	-	24	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	5609088	5616144	**738772**	**738772**	**92252**	**11542**	1458	161	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.48: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ from cosets of extended additive cyclic codes over GF(4) pairing** 1 **with ω for $3 \leq d \leq 20$**

n/d	3	4	5	6	7	8	9	10	11
4	$\mathit{6}_{ca}$	1_{ca}	-	-	-	-	-	-	-
5	4_{cf}	$\mathit{3}_{cf}$	-	-	-	-	-	-	-
6	27_{ca}	$\mathit{16}_{ca}$	-	1_{ca}	-	-	-	-	-
7	35_{ef}	22_{ea}	-	1^{co}_{ea}	-	-	-	-	-
8	280^{co}_{ca}	108_{ef}	16_{ca}	$\mathit{12}_{ea}$	-	1_{ca}	-	-	-
9	246_{ca}	126_{ca}	22^{co}_{ea}	8_{ca}	-	1_{ea}	-	-	-
10	800_{ca}	$\mathbf{860}_{cf}$	$\mathbf{210}_{ca}$	50_{ca}	-	4_{cf}	-	1_{ca}	-
11	800_{ea}	860_{ef}	226_{ea}	50^{co}_{ea}	-	4_{ef}	-	1_{ea}	-
12	$\mathbf{59136}_{ca}$	$\mathbf{14784}_{ca}$	$\mathbf{1848}_{ca}$	$\mathit{924}_{ca}$	84_{ca}	24_{ca}	6_{ca}	-	-
13	54752_{ea}	7552_{ea}	1696_{ea}	880^{co}_{ea}	64_{ea}	16_{ea}	4_{ea}	-	-
14	387744_{ca}	$\mathit{192192}_{ca}$	7424_{ca}	$\mathbf{3216}^{co}_{ca}$	$\mathbf{796}^{co}_{ca}$	84_{ca}	-	$\mathbf{21}_{ca}$	-
15	411560_{ea}	213632^{co}_{ea}	12835_{cf}	$\mathbf{6648}^{co}_{ea}$	800^{co}_{cf}	$\mathbf{406}^{co}_{cf}$	4_{cf}	31^{co}_{cf}	-
16	$\mathit{1317440}_{ef}$	$\mathit{3293600}_{ef}$	51480^{co}_{ca}	$\mathbf{55424}^{co}_{ea}$	$\mathbf{13856}^{co}_{ef}$	$\mathbf{3776}^{co}_{ef}$	-	20^{co}_{ca}	$\mathbf{68}^{co}_{ef}$
17	775680_{ea}	3111120_{cf}	48480_{ea}	48480_{ea}	12120_{cf}	$\mathit{6060}_{cf}$	48^{co}_{cf}	24^{co}_{ea}	-
18	11230272_{cf}	$\mathbf{11232288}_{ca}$	$\mathbf{699624}_{ca}$	$\mathbf{699624}_{ca}$	$\mathbf{43632}_{ca}$	$\mathbf{10890}_{ca}$	$\mathbf{2691}_{ca}$	144_{ca}	-
19	11230272_{ef}	5616144_{ea}	$\mathbf{738772}_{ea}$	$\mathbf{738772}_{ea}$	$\mathbf{92252}_{ea}$	$\mathbf{11542}_{ea}$	2893^{co}_{ea}	161^{co}_{ea}	-
20	23646752_{cf}	$\mathbf{189432064}^{co}_{cf}$	738520_{ca}	738520_{ca}	$\mathbf{369008}_{ca}$	$\mathit{184756}_{ef}$	6012_{ca}	1440_{ca}	-
21	45145632_{cf}	188416000_{ef}	22573824_{cf}	1412068^{co}_{cf}	11169^{co}_{cf}	12452^{co}_{ef}	5601^{co}_{cf}	368_{ef}	697^{co}_{cf}
22	-	165423104_{cf}	1410864_{cf}	646184_{cf}	-	-	-	4096_{cf}	4_{cf}
23	-	165423104_{ef}	-	1352078_{ef}	-	-	-	2510_{ef}	-
24	11076104704_{cf}	2784697344_{cf}	-	815488_{cf}	-	4672^{co}_{cf}	-	-	-
25	21300369664_{ef}	5328918784_{ef}	-	815488_{ef}	-	6080^{co}_{ef}	-	-	-
26	-	9891187072_{cf}	20801200_{cf}	9655984_{cf}	-	-	-	16384_{cf}	-
27	-	9891187072_{ef}	-	20057442_{ef}	-	-	-	9908_{ef}	-
28	21768306688_{cf}	2625500086272_{cf}	-	8106752_{cf}	3008_{cf}	3853632_{cf}	-	-	-
29	-	2625500086272_{ef}	-	6691200_{ef}	-	3853632_{ef}	-	-	38777664_{cf}
30	609973884610560_{cf}	152493461268480_{cf}	80766566400_{cf}	149011451520_{cf}	9313176480_{cf}	2332609440_{cf}	9080016_{cf}	9110544_{cf}	708168_{cf}

Table 10.49: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ before shortening or puncturing for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	3_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	24_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	26^{co}_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	16_{ca}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	32_{ca}	-	-	-	4_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	364^{co}_{cf}	-	3_{cf}	-	6^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	80_{cf}	-	-	-	5_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	96^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	848_{cf}	4_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	848_{ef}	-	3^{co}_{ef}	-	-	-	3_{cf}	-	-	-	-	-	-	-	-	-	-	-	-
28	1536_{cf}	-	20_{cf}	-	22_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	19388832_{cf}	-	20_{ef}	-	24^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	281928_{cf}	-	16384_{cf}	1024_{cf}	1104^{co}_{cf}	-	48_{cf}	-	58_{cf}	-	12_{cf}	-	6_{cf}	-	-	-	-	-	-

Table 10.50: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ before shortening or puncturing for $12 \leq d \leq 30$**

n/d	3	4	5	6	7	8	9	10	11
4	$\mathit{6}_{ca}$	1_{ca}	-	-	-	-	-	-	-
5	4_{cf}	$\mathit{3}_{cf}$	-	-	-	-	-	-	-
6	27_{ca}	$\mathit{16}_{ca}$	-	1_{ca}	-	-	-	-	-
7	35_{ef}	22_{ea}	-	1_{ea}^{co}	-	-	-	-	-
8	280_{ca}^{co}	108_{ef}	22_{nb}^{T5}	$\mathit{12}_{ea}$	-	1_{ca}	-	-	-
9	252_{st}^{3}	126_{ca}	22_{ea}^{co}	8_{ca}	-	1_{ea}	-	-	-
10	3280_{st}^{6}	$\mathbf{860}_{cf}$	$\mathbf{210}_{ca}$	50_{ca}	5_{st}^{6}	4_{cf}	-	2_{nt}^{C6}	-
11	3656_{st}^{6}	920_{st}^{3}	226_{ea}	56_{nt}^{C2}	5_{st}^{4}	4_{ef}	-	$\mathit{2}_{nt}^{C5}$	-
12	$\mathbf{59136}_{ca}$	$\mathbf{14784}_{ca}$	$\mathbf{1848}_{ca}$	$\mathit{924}_{ca}$	84_{ca}	24_{ca}	6_{ca}	-	$\mathit{2}_{nt}^{C4}$
13	54752_{ea}	13520_{st}^{5}	1696_{ea}	880_{ea}^{co}	68_{st}^{3}	30_{st}^{6}	4_{ea}	3_{nt}^{A1}	-
14	$\mathbf{768768}_{st}^{8}$	$\mathit{192192}_{ca}$	7424_{ca}	$\mathbf{3712}_{nb}^{C8}$	$\mathbf{796}_{ca}^{co}$	$\mathbf{208}_{nt}^{A1}$	12_{st}^{6}	$\mathbf{21}_{ca}$	4_{nt}^{A1}
15	411560_{ea}	213584_{ea}	12835_{cf}	$\mathbf{6648}_{ea}^{co}$	802_{st}^{8}	$\mathbf{410}_{st}^{8}$	15_{nt}^{C2}	31_{cf}^{co}	3_{st1}^{5}
16	$\mathit{13174400}_{ef}$	$\mathit{3293600}_{ef}$	51608_{sb}^{9}	$\mathbf{55424}_{ea}^{co}$	$\mathbf{13856}_{ef}^{co}$	$\mathbf{3776}_{ef}^{co}$	194_{st}^{9}	20_{ca}^{co}	$\mathbf{68}_{ef}^{co}$
17	777640_{st}^{9}	3111120_{cf}	$\mathbf{97520}_{st}^{3}$	48550_{st}^{9}	12120_{cf}	$\mathit{6060}_{cf}$	196_{nt}^{A2}	28_{st}^{7}	5_{st}^{9}
18	$\mathbf{44933184}_{st}^{10}$	$\mathbf{11266112}_{sb}^{10}$	$\mathbf{699624}_{ca}$	$\mathbf{699624}_{ca}$	$\mathbf{43632}_{sb}^{10}$	$\mathbf{10908}_{st}^{10}$	$\mathbf{2691}_{ca}$	144_{ca}	28_{st}^{10}
19	11824384_{pt}^{10}	11266112_{st}^{10}	$\mathbf{1477796}_{st}^{10}$	$\mathbf{738772}_{ea}$	$\mathbf{92252}_{ea}$	$\mathbf{11542}_{ea}$	2893_{ea}^{co}	198_{st}^{10}	53_{st}^{5}
20	755728384_{st}^{11}	$\mathbf{189432064}_{cf}^{co}$	$\mathbf{11822368}_{sb}^{11}$	738520_{ca}	$\mathbf{369008}_{ca}$	$\mathit{184756}_{ef}$	6012_{ca}	1592_{pr}^{11}	400_{st}^{2}
21	90291264_{st}^{11}	188416000_{ef}	22573824_{cf}	1412068_{cf}^{co}	176772_{st}^{8}	45112_{pt}^{8}	11148_{pt}^{2}	2926_{st}^{2}	847_{st}^{4}
22	10602158336_{st}^{12}	2650495232_{st}^{12}	22607872_{st}^{12}	5643456_{pt}^{12}	1410864_{st}^{12}	353496_{pt}^{1}	88424_{sb}^{12}	22088_{st}^{11}	5522_{st}^{3}
23	1384513088_{st}^{12}	670222080_{st}^{12}	43264648_{pt}^{12}	10816624_{pt}^{12}	2703694_{st}^{12}	676312_{st}^{11}	169182_{st}^{6}	42968_{pt}^{3}	10701_{st}^{3}
24	177279886336_{st}^{13}	44319794176_{st}^{13}	346436544_{pt1}^{12}	43355616_{st}^{12}	21631400_{sb}^{13}	5406464_{pt}^{13}	1351616_{st}^{13}	338016_{st}^{13}	84964_{st}^{3}
25	21300369664_{ef}	11204500480_{st}^{2}	10399676_{pt1}^{13}	1299844_{st}^{13}	41600552_{pt}^{13}	10399676_{pt}^{13}	2599688_{st}^{13}	649922_{st}^{13}	162986_{st}^{3}
26	2532157069312_{st}^{14}	633038608384_{st}^{14}	326893568_{st}^{14}	618544192_{st}^{14}	38656528_{st}^{14}	83204800_{pt}^{14}	20801200_{sb}^{14}	5199376_{st}^{14}	1299844_{st}^{14}
27	-	158680788992_{st}^{14}	-	20057442_{ef}	-	300224_{st}^{14}	40114884_{pt}^{14}	10029150_{pt}^{14}	2506644_{st}^{14}
28	$420617052487 68_{st}^{15}$	10515426312192_{st}^{15}	5154680832_{st}^{15}	10262347776_{st}^{15}	641396736_{st}^{15}	159987712_{st}^{15}	627776_{st}^{10}	80226336_{p1}^{15}	20056584_{sb}^{15}
29	-	2625500086272_{ef}	-	6691200_{ef}	-	3853632_{ef}	-	-	38777664_{cf}
30	609973884610560_{cf}	152493461268480_{cf}	80766566400_{cf}	149011451520_{cf}	9313176480_{cf}	2332609440_{cf}	9080016_{cf}	9110544_{cf}	708168_{cf}

Table 10.51: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ after shortening or puncturing for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	1^{7}_{stl}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	3_{cf}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	$\mathit{24}_{ef}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	26^{co}_{cf}	-	-	-	$\mathit{2}^{G1}_{nt}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	16_{ca}	-	-	-	$\mathit{2}^{A4}_{nt}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	28^{8}_{st}	-	-	-	2^{A6}_{nt}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	$\mathit{179}^{9}_{st}$	-	-	5^{1}_{pt}	$\mathbf{8}^{C8}_{nt}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	357^{5}_{st}	-	3_{cf}	-	6^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	2546^{12}_{sb}	4^{8}_{pt}	4^{12}_{pt1}	-	8^{A2}_{nt}	4^{7}_{pt1}	-	-	-	-	-	-	-	-	-	-	-	-	-
23	5312^{11}_{st}	-	-	-	2^{A1}_{nt}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	42796^{13}_{sb}	-	-	4^{11}_{pt}	32^{G12}_{nt}	-	4^{13}_{pt}	-	-	-	-	-	-	-	-	-	-	-	-
25	81772^{3}_{st}	-	-	-	8^{C4}_{nt}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	603734^{14}_{sb}	4_{cf}	-	-	80^{A8}_{nt}	4^{14}_{pr}	8^{4}_{st}	-	6^{14}_{pt}	-	-	-	-	-	-	-	-	-	-
27	1253105^{14}_{st}	-	3^{co}_{ef}	-	8^{A11}_{nt}	-	3_{cf}	-	-	-	-	-	-	-	-	-	-	-	-
28	10026576^{15}_{sb}	-	1104^{15}_{pt1}	-	320^{A4}_{nt}	-	28^{15}_{pt}	-	8^{15}_{pt}	-	4^{15}_{pt}	-	-	-	-	-	-	-	-
29	19388832_{cf}	-	20_{ef}	-	24^{co}_{ef}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	281928_{cf}	-	16384_{cf}	1024_{cf}	1104^{co}_{cf}	-	48_{cf}	-	58_{cf}	-	12_{cf}	-	6_{cf}	-	-	-	-	-	-

Table 10.52: **Best lower bounds for $A_4^{GC,RC}(n,d,w)$ after shortening or puncturing for $12 \leq d \leq 30$**

(n,d)	generator polynomials and involutions
(30,3)	$g(x) = x^4 + x^3 + \omega^2 x^2 + x + \omega$
	invol=(1, 18)(2, 25)(3, 16)(4, 19)(5, 12)(6, 23)(7, 30)(8, 21)(9, 24)(10, 17)(11, 28)(13, 26)(14, 29)(15, 22)(20, 27)
(30,4)	$g(x) = x^5 + \omega^2 x^4 + x^3 + \omega^2$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,5)	$g(x) = x^{10} + \omega x^9 + x^8 + x^4 + \omega x^3 + \omega x + 1$
	invol=(1, 16)(2, 17)(3, 18)(4, 19)(5, 20)(6, 21)(7, 22)(8, 23)(9, 24)(10, 25)(11, 26)(12, 27)(13, 28)(14, 29)(15, 30)
(30,6)	$g(x) = x^{10} + \omega^2 x^9 + x^6 + x^5 + \omega^2 x^4 + \omega x^3 + x^2 + \omega$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,7)	$g(x) = x^{12} + \omega^2 x^{11} + \omega^2 x^{10} + \omega x^9 + x^8 + x^7 + x^5 + \omega^2 x^4 + x^3 + x^2 + 1$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,8)	$g(x) = x^{13} + x^{11} + \omega x^8 + \omega^2 x^7 + x^6 + \omega^2 x^4 + \omega x^3 + \omega^2 x^2 + x + \omega^2$
	invol=(1, 10)(2, 29)(3, 18)(4, 7)(5, 26)(6, 15)(8, 23)(9, 12)(11, 20)(13, 28)(14, 17)(16, 25)(19, 22)(21, 30)(24, 27)
(30,9)	$g(x) = x^{17} + x^{14} + \omega x^{13} + x^{12} + x^{11} + \omega^2 x^8 + \omega x^7 + \omega x^5 + x^4 + x^3 + \omega x^2 + x + \omega$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,10)	$g(x) = x^{17} + \omega x^{15} + \omega^2 x^{13} + x^{12} + \omega^2 x^{11} + \omega^2 x^{10} + \omega x^9 + x^8 + x^7 + x^6 + x^5 + \omega x^4 + \omega^2 x^3 + \omega x^2 + x + 1$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,11)	$g(x) = x^{19} + \omega^2 x^{18} + \omega x^{17} + x^{16} + x^{15} + \omega x^{14} + x^{13} + x^{12} + \omega x^{11} + \omega^2 x^{10} + x^7 + x^5 + \omega^2 x^4 + \omega x^3 + \omega x^2 + \omega^2 x + \omega^2$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,12)	$g(x) = x^{20} + \omega x^{16} + x^{15} + \omega x^{13} + x^{12} + \omega x^{11} + \omega x^{10} + x^8 + \omega^2 x^7 + x^6 + \omega^2 x^3 + \omega x^2 + x + \omega$
	invol=(1, 28)(2, 17)(3, 6)(4, 25)(5, 14)(7, 22)(8, 11)(9, 30)(10, 19)(12, 27)(13, 16)(15, 24)(18, 21)(20, 29)(23, 26)
(30,14)	$g(x) = x^{22} + \omega x^{21} + \omega x^{19} + x^{18} + x^{16} + \omega^2 x^{15} + \omega x^{14} + \omega x^{13} + \omega x^{12} + \omega x^{11} + \omega x^{10} + \omega x^9 + \omega x^8 + \omega^2 x^7 + \omega x^5 + \omega^2 x^3 + \omega x^2 + \omega^2 x + 1$
	invol=(1, 22)(2, 26)(3, 30)(4, 19)(5, 23)(6, 12)(7, 16)(8, 20)(9, 24)(10, 28)(11, 17)(13, 25)(14, 29)(15, 18)(21, 27)
(30,15)	$g(x) = x^{24} + x^{22} + x^{21} + \omega x^{20} + x^{19} + \omega^2 x^{18} + x^{17} + x^{16} + \omega x^{14} + \omega x^{13} + \omega x^{12} + \omega x^{11} + \omega x^{10} + \omega^2 x^9 + \omega x^8 + \omega^2 x^7 + \omega^2 x^6 + \omega^2 x^4 +$ $x^3 + \omega^2 x^2 + \omega^2 x + \omega$
	invol=(1, 16)(2, 17)(3, 18)(4, 19)(5, 20)(6, 21)(7, 22)(8, 23)(9, 24)(10, 25)(11, 26)(12, 27)(13, 28)(14, 29)(15, 30)
(30,16)	$g(x) = x^{24} + \omega x^{23} + x^{22} + x^{20} + \omega x^{19} + \omega^2 x^{17} + \omega^2 x^{16} + x^{15} + x^9 + \omega x^8 + x^7 + x^5 + \omega x^4 + \omega^2 x^2 + \omega^2 x + 1$
	invol=(1, 22)(2, 26)(3, 30)(4, 19)(5, 23)(6, 27)(7, 16)(8, 20)(9, 24)(10, 28)(11, 17)(12, 21)(13, 25)(14, 29)(15, 18)
(30,18)	$g(x) = x^{26} + \omega x^{25} + \omega x^{24} + x^{23} + \omega^2 x^{22} + x^{21} + \omega x^{19} + x^{18} + \omega^2 x^{17} + \omega x^{15} + x^{11} + \omega x^{10} + \omega x^9 + x^8 + \omega^2 x^7 + x^6 + \omega x^4 + x^3 + \omega^2 x^2 + \omega$
	invol=(1, 22)(2, 26)(3, 30)(4, 19)(5, 23)(6, 27)(7, 16)(8, 20)(9, 24)(10, 28)(11, 17)(12, 21)(13, 25)(14, 29)(15, 18)
(30,20)	$g(x) = x^{26} + \omega x^{25} + \omega^2 x^{23} + \omega^2 x^{22} + x^{21} + x^{20} + \omega^2 x^{19} + \omega^2 x^{18} + \omega x^{16} + x^{15} + x^{11} + \omega x^{10} + \omega^2 x^8 + \omega^2 x^7 + x^6 + x^5 + \omega^2 x^4 + \omega^2 x^3 + \omega x + 1$
	invol=(1, 22)(2, 21)(3, 20)(4, 19)(5, 18)(6, 17)(7, 16)(8, 30)(9, 29)(10, 28)(11, 27)(12, 26)(13, 25)(14, 24)(15, 23)

Table 10.53: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $n = 30$ **.**

(n,d)	generator polynomials, involutions and coset leaders
$(30, 22)$	$g(x) = x^{27} + x^{25} + \omega x^{24} + x^{23} + \omega x^{21} + \omega x^{20} + \omega x^{19} + x^{18} + x^{17} + \omega x^{16} + \omega^2 x^{15} + x^{12} + x^{10} + \omega x^9 + x^8 + \omega x^6 + \omega x^5 + \omega x^4 + x^3 + x^2 + \omega x + \omega^2$
	invol=(1, 22)(2, 26)(3, 30)(4, 19)(5, 23)(6, 27)(7, 16)(8, 20)(9, 24)(10, 28)(11, 17)(12, 21)(13, 25)(14, 29)(15, 18)
$(30, 24)$	$g(x) = x^{28} + \omega^2 x^{27} + x^{26} + x^{25} + \omega^2 x^{23} + \omega x^{22} + \omega^2 x^{21} + \omega^2 x^{20} + \omega x^{18} + x^{17} + \omega x^{16} + \omega x^{15} + x^{13} + \omega^2 x^{12} + x^{11} + x^{10} + \omega^2 x^8 + \omega x^7 + \omega^2 x^6 + \omega^2 x^5 + \omega x^3 + x^2 + \omega x + \omega$
	invol=(1, 10)(2, 29)(3, 18)(4, 7)(5, 11)(6, 30)(8, 23)(9, 27)(12, 24)(13, 28)(14, 17)(15, 21)(16, 25)(19, 22)(20, 26)
$(29, 4)$	$g(x) = x^6 + x^5 + x^3 + 1$
	invol=(1, 2)(3, 10)(4, 18)(5, 21)(6, 27)(7, 19)(8, 23)(9, 22)(11, 25)(12, 28)(13, 20)(14, 26)(15, 16)(17, 24)
$(29, 6)$	$g(x) = x^{15} + x^{14} + x^{12} + x^7 + x^6 + x^4 + x^3 + x^2 + 1$
	invol=(1, 22)(2, 9)(3, 20)(4, 11)(5, 14)(6, 17)(7, 12)(8, 15)(10, 13)(16, 23)(18, 25)(19, 28)(21, 26)(24, 27)
$(29, 8)$	$g(x) = x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^9 + x^6 + x^5 + x^4 + x + 1$
	invol=(1, 2)(3, 10)(4, 18)(5, 21)(6, 27)(7, 19)(8, 23)(9, 22)(11, 25)(12, 28)(13, 20)(14, 26)(15, 16)(17, 24)
$(29, 11)$	$g(x) = x^{14} + \omega^2 x^{13} + \omega^2 x^{11} + \omega x^{10} + x^9 + \omega x^8 + \omega^2 x^7 + \omega x^6 + x^5 + \omega x^4 + \omega^2 x^3 + \omega^2 x + 1$
	invol=(1, 2)(3, 29)(4, 28)(5, 27)(6, 26)(7, 25)(8, 24)(9, 23)(10, 22)(11, 21)(12, 20)(13, 19)(14, 18)(15, 17)
$(29, 12)$	$g(x) = x^{15} + \omega^2 x^{14} + \omega x^{13} + \omega x^{12} + x^{11} + \omega x^{10} + \omega x^9 + x^8 + x^7 + \omega x^6 + \omega x^5 + x^4 + \omega x^3 + \omega x^2 + \omega^2 x + 1$
	invol=(1, 2)(3, 29)(4, 28)(5, 27)(6, 26)(7, 25)(8, 24)(9, 23)(10, 22)(11, 21)(12, 20)(13, 19)(14, 18)(15, 17)
$(29, 14)$	$g(x) = x^{25} + x^{24} + x^{21} + x^{20} + x^{17} + x^{16} + x^{13} + x^{12} + x^9 + x^8 + x^5 + x^4 + x + 1$
	invol=(1, 7)(2, 8)(3, 5)(4, 6)(9, 11)(10, 12)(13, 15)(14, 16)(17, 19)(18, 20)(21, 23)(22, 24)(25, 27)(26, 28)
$(29, 16)$	$g(x) = x^{25} + x^{24} + x^{23} + x^{21} + x^{18} + x^{17} + x^{16} + x^{14} + x^{11} + x^{10} + x^9 + x^7 + x^4 + x^3 + x^2 + 1$
	$\mathbf{L} = (\omega, 0, \omega^2, \omega, \omega^2, 1, \omega, 1, \omega^2, \omega^2, 1, 0, 0, 1, \omega^2, 1, 0, 0, 1, \omega^2, \omega^2, 1, \omega, 1, \omega^2, \omega, \omega^2, 0, \omega)$
	invol=(1, 15)(2, 23)(3, 6)(4, 11)(5, 7)(8, 22)(9, 16)(10, 13)(12, 14)(17, 20)(18, 25)(19, 21)(24, 27)(26, 28)
$(27, 6)$	$g(x) = x^{12} + \omega^2 x^{10} + \omega x^6 + \omega^2 x^2 + 1$
	invol= (1, 26)(2, 9)(3, 20)(4, 17)(5, 14)(6, 25)(7, 8)(10, 15)(11, 22)(12, 23)(13, 16)(18, 21)(19, 24)
$(27, 14)$	$g(x) = x^{24} + x^{22} + x^{20} + x^{18} + x^{16} + x^{14} + x^{12} + x^{10} + x^8 + x^6 + x^4 + x^2 + 1$
	$\mathbf{L} = (\omega, \omega^2, 1, \omega^2, 1, 1, 0, \omega^2, 1, \omega, 0, \omega, 1, 0, 1, \omega, 0, \omega, 1, \omega^2, 0, 1, 1, \omega^2, 1, \omega^2, \omega)$
	invol= (1, 4)(2, 3)(5, 6)(7, 8)(9, 10)(11, 12)(13, 14)(15, 16)(17, 18)(19, 20)(21, 22)(23, 24)(25, 26)
$(27, 18)$	$g(x) = x^{25} + x^{24} + x^{22} + x^{21} + x^{19} + x^{18} + x^{16} + x^{15} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^3 + x + 1$
	invol= (1, 2)(3, 6)(4, 5)(7, 8)(9, 12)(10, 11)(13, 14)(15, 18)(16, 17)(19, 20)(21, 24)(22, 23)(25, 26)
$(26, 13)$	$g(x) = x^{24} + x^{22} + x^{20} + x^{18} + x^{16} + x^{14} + x^{12} + x^{10} + x^8 + x^6 + x^4 + x^2 + 1$
	invol= (1, 14)(2, 15)(3, 16)(4, 17)(5, 18)(6, 19)(7, 20)(8, 21)(9, 22)(10, 23)(11, 24)(12, 25)(13, 26)
$(25, 3)$	$g(x) = x^5 + \omega x^4 + \omega x + \omega^2$
	invol= (1, 7)(2, 14)(3, 21)(4, 16)(5, 11)(6, 18)(8, 20)(9, 15)(10, 22)(12, 24)(13, 19)(17, 23)

Table 10.54: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $25 \leq n \leq 30$ **.**

(n,d)			generator polynomials, involutions and coset leaders
$(21,4)$	$g(x)$	$=$	$x^5+\omega^2x^4+\omega x^3+\omega x^2+\omega^2x+1$
	invol	$=$	$(1,\,6)(2,\,5)(3,\,4)(7,\,20)(8,\,19)(9,\,18)(10,\,17)(11,\,16)(12,\,15)(13,\,14)$
$(21,5)$	$g(x)$	$=$	$x^7+\omega^2x^6+x^4+x^3+\omega^2x+1$
	invol	$=$	$(1,\,2)(3,\,21)(4,\,20)(5,\,19)(6,\,18)(7,\,17)(8,\,16)(9,\,15)(10,\,14)(11,\,13)$
$(21,6)$	$g(x)$	$=$	$x^9+\omega^2x^8+\omega^2x^7+\omega^2x^6+\omega^2x^3+\omega^2x^2+\omega^2x+1$
	$\mathbf{L}$	$=$	$(\omega,0,0,\omega,0,\omega,0,0,\omega^2,\omega,1,\omega,\omega^2,0,0,\omega,0,\omega,0,0,\omega)$
	invol	$=$	$(1,\,8)(2,\,7)(3,\,6)(4,\,5)(9,\,21)(10,\,20)(11,\,19)(12,\,18)(13,\,17)(14,\,16)$
$(21,14)$	$g(x)$	$=$	$x^{19}+x^{18}+x^{16}+x^{15}+x^{13}+x^{12}+x^{10}+x^9+x^7+x^6+x^4+x^3+x+1$
	invol	$=$	$(1,\,10)(2,\,3)(4,\,13)(5,\,6)(7,\,16)(8,\,9)(11,\,12)(14,\,15)(17,\,18)(20,\,21)$
$(21,16)$	$g(x)$	$=$	$x^{18}+\omega^2x^{17}+\omega^2x^{16}+x^{15}+x^{13}+\omega^2x^{12}+\omega^2x^{11}+x^{10}+x^8+\omega^2x^7+\omega^2x^6$ $+x^5+x^3+\omega^2x^2+\omega^2x+1$
	$\mathbf{L}$	$=$	$(\omega,0,0,\omega^2,1,0,\omega,1,\omega,\omega,0,\omega,\omega,1,\omega,0,1,\omega^2,0,0,\omega)$
	invol	$=$	$(1,\,11)(2,\,10)(3,\,9)(4,\,8)(5,\,7)(6,\,16)(12,\,15)(13,\,14)(17,\,20)(18,\,19)$
$(20,4)$	$g(x)$	$=$	$x^5+\omega^2x^4+\omega x^3+\omega x^2+\omega^2x+1$
	$\mathbf{L}$	$=$	$(\omega^2,0,\omega^2,0,0,\omega,\omega^2,0,\omega,1,1,\omega,0,\omega^2,\omega,0,0,\omega^2,0,\omega^2)$
	invol	$=$	$(1,\,7)(2,\,16)(3,\,5)(4,\,14)(6,\,12)(8,\,10)(9,\,19)(11,\,17)(13,\,15)(18,\,20)$
$(20,6)$	$\omega p(x)+q(x)$	$=$	$x^{12}+x^9+x^7+x^6+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{17}+x^{16}+x^{13}+x^{12}+x^9+x^8+x^5+x^4+x+1$
	invol	$=$	$(1,\,11)(2,\,12)(3,\,13)(4,\,14)(5,\,15)(6,\,16)(7,\,17)(8,\,18)(9,\,19)(10,\,20)$
$(20,7)$	$\omega p(x)+q(x)$	$=$	$x^{12}+x^{10}+x^7+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{18}+x^{16}+x^{14}+x^{12}+x^{10}+x^8+x^6+x^4+x^2+1$
	invol	$=$	$(1,\,11)(2,\,12)(3,\,13)(4,\,14)(5,\,15)(6,\,16)(7,\,17)(8,\,18)(9,\,19)(10,\,20)$
$(20,8)$	$g(x)$	$=$	$x^9+\omega x^8+\omega x^6+\omega x^5+\omega^2x^4+\omega^2x^3+\omega^2x+1$
	invol	$=$	$(1,\,18)(2,\,14)(3,\,20)(4,\,11)(5,\,7)(6,\,12)(8,\,16)(9,\,17)(10,\,15)(13,\,19)$
$(20,9)$	$\omega p(x)+q(x)$	$=$	$x^{13}+x^{11}+x^9+x^7+x^6+\omega^2x^4+\omega^2x^3+\omega^2x^2+\omega x+\omega$
	$r(x)$	$=$	$x^{20}+1$
	invol	$=$	$(1,\,11)(2,\,12)(3,\,13)(4,\,14)(5,\,15)(6,\,16)(7,\,17)(8,\,18)(9,\,19)(10,\,20)$
$(19,6)$	$\omega p(x)+q(x)$	$=$	$x^{10}+x^8+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{13}+x^{12}+x^7+x^6+x+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(19,7)$	$\omega p(x)+q(x)$	$=$	$x^{13}+x^{11}+x^{10}+x^6+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{16}+x^{14}+x^{12}+x^{10}+x^8+x^6+x^4+x^2+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(19,8)$	$\omega p(x)+q(x)$	$=$	$x^{17}+x^{13}+x^{12}+x^{10}+x^8+x^6+x^5+x^4+x^3+\omega^2x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{18}+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(19,9)$	$\omega p(x)+q(x)$	$=$	$x^{17}+x^{16}+x^{11}+x^7+\omega x^4+\omega^2x^2+x+\omega^2$
	$r(x)$	$=$	$x^{18}+1$
	$\mathbf{L}$	$=$	$(0,1,\omega^2,1,0,1,\omega,0,1,1,1,0,\omega,1,0,1,\omega^2,1,0)$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(18,5)$	$\omega p(x)+q(x)$	$=$	$x^{12}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{13}+x^{12}+x^7+x^6+x+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(18,6)$	$\omega p(x)+q(x)$	$=$	$x^{10}+x^5+x^4+x^3+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{13}+x^{12}+x^7+x^6+x+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(18,9)$	$\omega p(x)+q(x)$	$=$	$x^{17}+x^{16}+x^{15}+x^{12}+x^{11}+x^{10}+x^9+x^7+x^6+x^5+\omega x^4+\omega^2x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{18}+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(18,10)$	$\omega p(x)+q(x)$	$=$	$x^{17}+x^{12}+x^9+\omega x^8+x^7+\omega x^6+\omega^2x^5+x^4+\omega^2x^3+\omega^2x^2+x+\omega^2$
	$r(x)$	$=$	$x^{18}+1$
	invol	$=$	$(1,\,10)(2,\,11)(3,\,12)(4,\,13)(5,\,14)(6,\,15)(7,\,16)(8,\,17)(9,\,18)$
$(18,12)$	$\omega p(x)+q(x)$	$=$	$x^{15}+\omega x^{14}+\omega^2x^{13}+\omega^2x^{12}+x^9+\omega x^8+\omega^2x^7+\omega^2x^6+x^3+\omega x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{16}+x^{15}+x^{13}+x^{12}+x^{10}+x^9+x^7+x^6+x^4+x^3+x+1$
	invol	$=$	$(1,\,4)(2,\,5)(3,\,6)(7,\,10)(8,\,11)(9,\,12)(13,\,16)(14,\,17)(15,\,18)$

Table 10.55: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $18 \le n \le 21$ **.**

(n,d)			generator polynomials, involutions and coset leaders
$(17,4)$	$g(x)$	$=$	$x^4+x^3+\omega x^2+x+1$
	invol	$=$	$(1,\,2)(3,\,17)(4,\,16)(5,\,15)(6,\,14)(7,\,13)(8,\,12)(9,\,11)$
$(17,7)$	$g(x)$	$=$	$x^8+\omega^2x^7+x^6+x^5+\omega x^4+x^3+x^2+\omega^2x+1$
	invol	$=$	$(1,\,2)(3,\,17)(4,\,16)(5,\,15)(6,\,14)(7,\,13)(8,\,12)(9,\,11)$
$(17,8)$	$g(x)$	$=$	$x^9+\omega x^8+\omega x^7+\omega^2x^5+\omega^2x^4+\omega x^2+\omega x+1$
	invol	$=$	$(1,\,2)(3,\,17)(4,\,16)(5,\,15)(6,\,14)(7,\,13)(8,\,12)(9,\,11)$
$(17,12)$	$g(x)$	$=$	$x^{13}+\omega^2x^{12}+\omega^2x^{11}+\omega x^{10}+x^9+\omega^2x^7+\omega^2x^6+x^4+\omega x^3+\omega^2x^2+\omega^2x+1$
	$\mathbf{L}$	$=$	$(0,\omega^2,\omega^2,0,1,1,0,\omega^2,\omega,\omega^2,0,1,1,0,\omega^2,\omega^2,0)$
	invol	$=$	$(1,\,9)(2,\,8)(3,\,7)(4,\,6)(10,\,17)(11,\,16)(12,\,15)(13,\,14)$
$(16,3)$	$g(x)$	$=$	$x^2+\omega x+\omega$
	invol	$=$	$(1,\,3)(2,\,8)(4,\,12)(5,\,15)(6,\,9)(7,\,11)(10,\,16)(13,\,14)$
$(16,4)$	$g(x)$	$=$	$x^3+\omega^2x^2+\omega^2$
	invol	$=$	$(1,\,3)(2,\,11)(4,\,14)(5,\,6)(7,\,15)(8,\,12)(9,\,16)(10,\,13)$
$(16,6)$	$\omega p(x)+q(x)$	$=$	$x^6+\omega^2x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^8+x^7+x^6+x^4+1$
	$\mathbf{L}$	$=$	$(1,0,\omega^2,1,1,\omega^2,1,0,0,1,\omega^2,1,1,\omega^2,0,1)$
	invol	$=$	$(1,\,9)(2,\,6)(3,\,16)(4,\,7)(5,\,11)(8,\,13)(10,\,12)(14,\,15)$
$(16,7)$	$g(x)$	$=$	$x^7+\omega x^5+\omega^2x^4+x^3+\omega x^2+\omega$
	$\mathbf{L}$	$=$	$(0,0,1,0,\omega^2,\omega^2,\omega^2,\omega,\omega,\omega^2,\omega^2,\omega^2,0,1,0,0)$
	invol	$=$	$(1,\,3)(2,\,11)(4,\,14)(5,\,6)(7,\,15)(8,\,12)(9,\,16)(10,\,13)$
$(16,8)$	$g(x)$	$=$	$x^8+\omega^2x^7+\omega x^6+\omega x^5+\omega^2x^4+x^3+x^2+\omega x+1$
	$\mathbf{L}$	$=$	$(0,\omega,\omega,\omega,1,0,\omega,0,0,\omega,0,1,\omega,\omega,\omega,0)$
	invol	$=$	$(1,\,3)(2,\,11)(4,\,14)(5,\,6)(7,\,15)(8,\,12)(9,\,16)(10,\,13)$
$(16,10)$	$\omega p(x)+q(x)$	$=$	$x^{13}+\omega x^{11}+\omega^2x^{10}+\omega^2x^9+\omega^2x^8+x^5+\omega x^3+\omega^2x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{15}+x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	$\mathbf{L}$	$=$	$(1,\omega^2,\omega,\omega,\omega^2,1,\omega,\omega,\omega,\omega,1,\omega^2,\omega,\omega,\omega^2,1)$
	invol	$=$	$(1,\,13)(2,\,14)(3,\,15)(4,\,16)(5,\,9)(6,\,10)(7,\,11)(8,\,12)$
$(16,11)$	$g(x)$	$=$	$x^{11}+x^{10}+\omega x^8+\omega^2x^7+\omega^2x^6+x^5+\omega^2x^4+x^3+x+\omega^2$
	$\mathbf{L}$	$=$	$(\omega^2,0,1,0,\omega,1,\omega,\omega,\omega,\omega,1,\omega,0,1,0,\omega^2)$
	invol	$=$	$(1,\,3)(2,\,11)(4,\,14)(5,\,6)(7,\,15)(8,\,12)(9,\,16)(10,\,13)$
$(16,12)$	$g(x)$	$=$	$x^{12}+\omega x^{10}+\omega^2x^9+\omega^2x^8+\omega^2x^6+\omega x^5+x^4+x^3+\omega^2x^2+x+\omega$
	invol	$=$	$(1,\,3)(2,\,8)(4,\,12)(5,\,15)(6,\,9)(7,\,11)(10,\,16)(13,\,14)$
$(15,3)$	$\omega p(x)+q(x)$	$=$	$x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	x^6+x^2+1
	invol	$=$	$(1,\,4)(2,\,13)(3,\,6)(5,\,8)(7,\,10)(9,\,12)(11,\,14)$
$(15,4)$	$\omega p(x)+q(x)$	$=$	$x^6+x^4+x^3+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^7+x^6+x^3+x^2+x+1$
	invol	$=$	$(1,\,6)(2,\,3)(4,\,11)(5,\,12)(7,\,14)(8,\,13)(9,\,10)$
$(15,5)$	$g(x)$	$=$	$x^6+x^5+\omega x^4+x^3+\omega x^2+x+1$
	invol	$=$	$(1,\,2)(3,\,15)(4,\,14)(5,\,13)(6,\,12)(7,\,11)(8,\,10)$
$(15,6)$	$\omega p(x)+q(x)$	$=$	$x^9+x^8+x^7+x^5+x^4+\omega x^3+\omega^2x^2+x+\omega^2$
	$r(x)$	$=$	$x^{10}+x^8+x^7+x^3+x+1$
	$\mathbf{L}$	$=$	$(1,1,1,0,\omega^2,1,\omega,\omega,\omega,1,\omega^2,0,1,1,1)$
	invol	$=$	$(1,\,8)(2,\,9)(3,\,10)(4,\,11)(5,\,12)(6,\,13)(7,\,14)$
$(15,10)$	$g(x)$	$=$	$x^{11}+\omega x^{10}+\omega^2x^8+\omega^2x^7+x^6+x^5+\omega^2x^4+\omega^2x^3+\omega x+1$
	$\mathbf{L}$	$=$	$(\omega,\omega,0,\omega,\omega,\omega^2,\omega,\omega,\omega,\omega^2,\omega,\omega,0,\omega,\omega)$
	invol	$=$	$(1,\,2)(3,\,15)(4,\,14)(5,\,13)(6,\,12)(7,\,11)(8,\,10)$
$(14,4)$	$\omega p(x)+q(x)$	$=$	$x^6+x^4+x^3+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^7+x^6+x^3+x^2+x+1$
	invol	$=$	$(1,\,8)(2,\,5)(3,\,10)(4,\,13)(6,\,11)(7,\,14)(9,\,12)$
$(14,5)$	$\omega p(x)+q(x)$	$=$	$x^8+x^7+x^6+x^5+x^4+\omega^2x^3+\omega^2x^2+x+\omega^2$
	$r(x)$	$=$	$x^9+x^8+x^6+x^5+x^4+x^3+1$
	invol	$=$	$(1,\,8)(2,\,9)(3,\,10)(4,\,11)(5,\,12)(6,\,13)(7,\,14)$

Table 10.56: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $14 \le n \le 17$ **.**

(n,d)			generator polynomials, involutions and coset leaders
$(14,7)$	$\omega p(x)+q(x)$	$=$	$x^{10}+\omega^2x^5+\omega^2x^4+\omega x^3+x^2+x+\omega^2$
	$r(x)$	$=$	$x^{11}+x^{10}+x^9+x^7+x^4+x^3+x^2+1$
	$\mathbf{L}$	$=$	$(1,\omega^2,\omega^2,\omega^2,\omega^2,\omega,\omega,\omega,\omega,\omega^2,\omega^2,\omega^2,\omega^2,1)$
	invol	$=$	(1, 8)(2, 9)(3, 10)(4, 11)(5, 12)(6, 13)(7, 14)
$(14,10)$	$\omega p(x)+q(x)$	$=$	$x^{11}+x^{10}+\omega x^8+\omega x^6+x^5+\omega^2x^4+x^3+x^2+x+\omega^2$
	$r(x)$	$=$	$x^{14}+1$
	invol	$=$	(1, 8)(2, 9)(3, 10)(4, 11)(5, 12)(6, 13)(7, 14)
$(13,3)$	$\omega p(x)+q(x)$	$=$	$x^3+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^5+x^3+x^2+1$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(13,5)$	$\omega p(x)+q(x)$	$=$	$x^9+x^7+x^6+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{10}+x^8+x^6+x^4+x^2+1$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(13,6)$	$\omega p(x)+q(x)$	$=$	$x^8+x^6+\omega^2x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{10}+x^9+x^7+x^6+x^4+x^3+x+1$
	$\mathbf{L}$	$=$	$(\omega,0,\omega^2,1,1,0,1,0,1,1,\omega^2,0,\omega)$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(13,9)$	$\omega p(x)+q(x)$	$=$	$x^{10}+x^9+\omega^2x^8+x^6+x^5+\omega^2x^4+x^2+x+\omega^2$
	$r(x)$	$=$	$x^{12}+1$
	invol	$=$	(1, 11)(2, 12)(3, 9)(4, 10)(5, 7)(6, 8)
$(12,3)$	$\omega p(x)+q(x)$	$=$	$x^3+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^5+x^3+x^2+1$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(12,4)$	$\omega p(x)+q(x)$	$=$	$x^6+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	x^7+x^6+x+1
	invol	$=$	(1, 11)(2, 4)(3, 9)(5, 7)(6, 12)(8, 10)
$(12,5)$	$\omega p(x)+q(x)$	$=$	$x^9+x^7+x^6+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{10}+x^8+x^6+x^4+x^2+1$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(12,6)$	$\omega p(x)+q(x)$	$=$	$x^8+x^6+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(12,7)$	$\omega p(x)+q(x)$	$=$	$x^{10}+x^9+x^7+\omega x^5+\omega x^4+\omega^2x^3+\omega^2x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	invol	$=$	(1, 11)(2, 4)(3, 9)(5, 7)(6, 12)(8, 10)
$(12,8)$	$\omega p(x)+q(x)$	$=$	$x^{10}+x^8+\omega^2x^7+\omega x^5+\omega x^4+\omega x^3+\omega x^2+x+\omega^2$
	$r(x)$	$=$	$x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	invol	$=$	(1, 7)(2, 8)(3, 9)(4, 10)(5, 11)(6, 12)
$(12,9)$	$\omega p(x)+q(x)$	$=$	$x^{10}+\omega^2x^9+\omega^2x^8+x^6+\omega^2x^5+\omega^2x^4+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	invol	$=$	(1, 11)(2, 12)(3, 9)(4, 10)(5, 7)(6, 8)
$(11,5)$	$\omega p(x)+q(x)$	$=$	$x^7+x^5+x^4+x^3+x^2+\omega^2x+\omega^2$
	$r(x)$	$=$	$x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^2+x+1$
	invol	$=$	(1, 6)(2, 7)(3, 8)(4, 9)(5, 10)
$(11,8)$	$g(x)$	$=$	$x^8+\omega^2x^7+\omega^2x^6+x^5+x^3+\omega^2x^2+\omega^2x+1$
	invol	$=$	(1, 2)(3, 5)(4, 9)(6, 7)(8, 10)
$(10,4)$	$g(x)$	$=$	$x^4+\omega x^3+\omega x+1$
	invol	$=$	(1, 8)(2, 7)(3, 6)(4, 5)(9, 10)
$(10,5)$	$\omega p(x)+q(x)$	$=$	$x^7+x^6+x^5+x^4+\omega^2x^2+\omega^2$
	$r(x)$	$=$	$x^8+x^6+x^4+x^2+1$
	invol	$=$	(1, 6)(2, 7)(3, 8)(4, 9)(5, 10)
$(10,6)$	$\omega p(x)+q(x)$	$=$	$x^6+x^5+x^4+\omega^2x^2+x+\omega^2$
	$r(x)$	$=$	$x^{10}+1$
	invol	$=$	(1, 6)(2, 7)(3, 8)(4, 9)(5, 10)

Table 10.57: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $10 \leq n \leq 14$ **.**

(n,d)			generator polynomials, involutions and coset leaders
$(10, 8)$	$g(x)$	$=$	$x^8 + \omega^2 x^7 + \omega^2 x^6 + x^5 + x^3 + \omega^2 x^2 + \omega^2 x + 1$
	invol	$=$	$(1,\ 7)(2,\ 6)(3,\ 10)(4,\ 9)(5,\ 8)$
$(9, 4)$	$\omega p(x) + q(x)$	$=$	$x^6 + x^4 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^7 + x^6 + x^4 + x^3 + x + 1$
	invol	$=$	$(1,\ 6)(2,\ 5)(3,\ 4)(7,\ 9)$
$(9, 5)$	$\omega p(x) + q(x)$	$=$	$x^5 + \omega x^3 + \omega^2 x^2 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
	$\mathbf{L}$	$=$	$(0, \omega, 0, 0, \omega, 0, 0, \omega, 0)$
	invol	$=$	$(1,\ 5)(2,\ 6)(3,\ 7)(4,\ 8)$
$(9, 6)$	$\omega p(x) + q(x)$	$=$	$x^8 + x^4 + \omega^2 x^3 + x^2 + x + \omega^2$
	$r(x)$	$=$	$x^9 + 1$
	invol	$=$	$(1,\ 6)(2,\ 5)(3,\ 4)(7,\ 9)$
$(8, 3)$	$\omega p(x) + q(x)$	$=$	$x^2 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^5 + x^4 + x + 1$
	$\mathbf{L}$	$=$	$(\omega^2, 1, 1, 0, 0, 1, 1, \omega^2)$
	invol	$=$	$(1,\ 5)(2,\ 6)(3,\ 7)(4,\ 8)$
$(8, 4)$	$g(x)$	$=$	$x^3 + x + 1$
	invol	$=$	$(1,\ 4)(2,\ 3)(5,\ 8)(6,\ 7)$
$(8, 6)$	$\omega p(x) + q(x)$	$=$	$x^5 + \omega x^3 + x^2 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
	invol	$=$	$(1,\ 5)(2,\ 7)(3,\ 4)(6,\ 8)$
$(7, 3)$	$g(x)$	$=$	$x^2 + \omega^2$
	invol	$=$	$(1,\ 6)(2,\ 3)(4,\ 5)$
$(7, 4)$	$\omega p(x) + q(x)$	$=$	$x^3 + \omega x^2 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^4 + x^3 + x + 1$
	invol	$=$	$(1,\ 4)(2,\ 5)(3,\ 6)$
$(6, 3)$	$\omega p(x) + q(x)$	$=$	$x^3 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^4 + x^2 + 1$
	invol	$=$	$(1,\ 4)(2,\ 5)(3,\ 6)$
$(6, 4)$	$\omega p(x) + q(x)$	$=$	$x^3 + \omega x^2 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^4 + x^3 + x + 1$
	invol	$=$	$(1,\ 4)(2,\ 5)(3,\ 6)$
$(5, 3)$	$g(x)$	$=$	$x^2 + \omega x + 1$
	invol	$=$	$(1,\ 4)(2,\ 3)$
$(4, 3)$	$\omega p(x) + q(x)$	$=$	$x^2 + \omega^2 x + \omega^2$
	$r(x)$	$=$	$x^3 + x^2 + x + 1$
	invol	$=$	$(1,\ 3)(2,\ 4)$

Table 10.58: **Generator polynomials and coset leaders L for codes with more than 4 codewords,** $4 \leq n \leq 10$ **.**

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	**5918**	-
21	-	1444724736	90295296	5649280	-	-	-	22216	-
22	-	-	-	20698496	-	-	-	81256	20428
23	-	-	346131968	-	-	-	-	339416	85078
24	-	-	1384527872	-	-	-	-	1354360	339276
25	-	340746305536	5325107200	-	-	-	-	5203408	1301216
26	-	-	19778969600	-	-	-	302498	19332032	4832144
27	-	-	20539699200	5135431680	321036160	80241120	1254372	-	20064240
28	-	-	82158796800	20539699200	1283857920	320938240	5015136	-	80241120
29	-	81326254325760	317680680960	79420170240	4963760640	1240940160	19395928	-	-
30	-	-	-	-	-	-	-	-	-

Table 10.59: **Best lower bounds for $A_4^{GC}(n,d,w)$ for codes with an all 1s vector in the dual for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	21394	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	85680	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	326592	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	1211984	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	5015136	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	20060280	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	77558760	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	290851224	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.60: **Best lower bounds for $A_4^{GC}(n,d,w)$ for codes with an all 1s vector in the dual for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
9	-	-	-	-	6	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	-	54912	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-
17	49786880	-	-	-	-	-	-	196	-
18	199147520	-	3111680	-	-	-	-	760	-
19	*756760576*	47443968^{19}_{sb}	11824384	2970880^{20}_{sb}	-	-	-	-	-
20	*3027042304*	188604416	47297536	11844096^{21}_{sb}	-	-	-	-	-
21	11557797888	-	180590592	11844096	2821728	-	-	-	-
22	11557797888	-	180590592	4519424^{12}_{sb}	2842944^{19}_{sb}	-	-	-	-
23	44304891904	-	692263936	174316544	10883808	-	-	-	-
24	-	-	2769055744	696563712	43266496	-	-	-	-
25	-	-	10650214400	2665861120	166485760^{19}_{sb}	41602400	-	-	-
26	-	-	42600857600	10636984320	665638400	166409600	41602400	-	-
27	-	-	164317593600	41092628480	1283731200	-	160466400	40116600	-
28	-	-	-	41092628480	5134924800	1284618240	-	160466400	-
29	-	-	2541445447680	153384255488	19855042560	4965568512	-	-	155117520
30	-	-	10165781790720	635297857536	74456409600	18618212352	620656128	-	-

Table 10.61: **Lower bounds for $A_4^{GC}(n,d,w)$ from best linear codes for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	40^{20}_{sb}	14	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-
20	-	144^{19}_{sb}	32^{3}_{sb}	12	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
21	-	288	96	32^{21}_{sb}	24	-	-	-	-	2	-	-	-	-	-	-	-	-	-
22	-	351^{22}_{sb}	384	96	42^{17}_{sb}	6^{14}_{sb}	-	-	-	-	2	-	-	-	-	-	-	-	-
23	-	1399	198^{21}_{sb}	336	168	6^{24}_{sb}	-	2	-	-	-	-	-	-	-	-	-	-	-
24	-	5572	752	149^{12}_{sb}	672	18	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	2585^{16}_{sb}	1336^{19}_{sb}	716	340	60	20	-	-	-	-	-	-	-	-	-	-	-	-
26	-	9992	4686	684^{23}_{sb}	1313	20^{5}_{sb}	64	20	-	-	-	-	-	-	2	-	-	-	-
27	-	39042	19404	2597	654	291	52^{10}_{sb}	72	36	-	-	-	-	-	-	2	-	-	-
28	-	77976	19664^{20}_{sb}	9702	2542	1120	162	64^{12}_{sb}	126	-	-	-	-	-	-	-	2	-	-
29	-	301656	75480	18820^{20}_{sb}	9672	623	175^{25}_{sb}	158	42	20	-	-	-	-	-	-	-	2	-
30	-	1140255	284025	71350	18270	2500	632	-	156	68	-	4	2	2	2	2	2	2	-

Table 10.62: **Lower bounds for $A_4^{GC}(n,d,w)$ from best linear codes for** $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
9	-	-	-	-	2	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-
13	-	13688	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	26^{6}_{st}	-
17	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	380	-
19	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-

Table 10.63: **Lower bounds for $A_4^{GC,RC}(n,d,w)$ from best linear codes for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	5^{8}_{st}	4^{10}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	3^{10}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	47^{10}_{st}	16^{3}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	10^{4}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	160^{4}_{st}	192	3^{9}_{sl2}	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	24^{9}_{st}	10^{11}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	2744	48^{4}_{st}	10^{9}_{st}	320	8^{7}_{sl2}	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	188	86	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	605^{4}_{st}	157^{11}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	9790^{10}_{st}	2420^{10}_{st}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	141810	35480	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 10.64: **Lower bounds for $A_4^{GC,RC}(n,d,w)$ from best linear codes for $12 \leq d \leq 30$**

n/d	3	4	5	6	7	8	9	10	11
4	$\mathit{12}_a$	$\mathit{4}_a$	-	-	-	-	-	-	-
5	20_a	$\mathit{10}_a$	2_a	-	-	-	-	-	-
6	80_a	$\mathit{40}_a$	4_a	$\mathit{4}_a$	-	-	-	-	-
7	$\mathbf{280}_a$	70_a	14_a	$\mathit{7}_a$	2_a	-	-	-	-
8	560_a	224_a	44_a	$\mathit{28}_a$	3_a	$\mathit{4}_a$	-	-	
9	2016_a	504_a	132_a	$\mathbf{42}_a$	6_b	4_a	$\mathit{3}_a$	-	-
10	6720_a	$\mathbf{2016}_a$	$\mathbf{504}_a$	$\mathbf{144}_a$	12_a	$\mathit{16}_a$	3_a	$\mathit{4}_a$	-
11	29568_a	$\mathit{7392}_a$	924_a	$\mathbf{472}_a$	42_a	16_a	3_a	$\mathit{4}_a$	2_a
12	$\mathit{118272}_a$	$\mathit{29568}_a$	$\mathbf{3696}_a$	$\mathbf{1888}_a$	168_a	64_a	12_a	4_a	2_a
13	439296_a	54912_b	7232_a	$\mathbf{3432}_a$	$\mathbf{464}_a$	128_a	28_a	11_a	4_a
14	$\mathit{1537536}_a$	$\mathbf{439296}_a$	$\mathbf{28032}_a$	$\mathbf{7424}_a$	$\mathbf{1856}_a$	$\mathbf{512}_a$	64_a	$\mathbf{49}_a$	10_a
15	$\mathit{6589440}_a$	$\mathit{1647360}_a$	$\mathbf{103680}_a$	$\mathbf{27840}_a$	$\mathbf{6960}_a$	$\mathbf{1920}_a$	$\mathbf{240}_a$	$\mathbf{124}_a$	31_a
16	$\mathit{26357760}_a$	$\mathit{6589440}_a$	$\mathbf{414720}_a$	$\mathit{111360}_a$	$\mathbf{27840}_a$	6480_a	423_a	124_a	104_a
17	49786880_b	12446720_a	777920_a	388960_a	$\mathit{48620}_a$	$\mathit{24310}_a$	$\mathbf{1530}_a$	196_b	100_a
18	199147520_b	24893440_a	3111680_b	1400256_a	$\mathbf{97520}_a$	$\mathit{87516}_a$	$\mathbf{5508}_a$	760_b	132_a
19	$\mathit{756760576}_b$	47443968_b	11824384_b	2970880_b	369512_a	$\mathit{92378}_a$	6138_a	1056_a	536_a
20	$\mathit{3027042304}_b$	$\mathbf{378896384}_a$	47297536_b	11844096_b	$\mathit{1478048}_a$	$\mathbf{369512}_a$	$\mathbf{24552}_a$	$\mathbf{5918}_\mathbf{c}$	$\mathbf{2112}_a$
21	11557797888_b	1444724736_c	180590592_b	11844096_b	2821728_b	516096_a	44764_a	22216_c	8064_a
22	11557797888_b	5301108736_a	180590592_b	45194240_b	2842944_b	707168_a	177352_a	81256_c	20428_c
23	44304891904_b	22160015360_a	692263936_b	174316544_b	10883808_b	2705248_a	676956_a	339416_c	85078_c
24	354439135232_a	88609783808_a	2769055744_b	696563712_b	43266496_b	10816624_a	2704156_a	1354360_c	339276_c
25	1362985222144_a	340746305536_a	10650214400_b	2665861120_b	166485760_b	41602400_b	10400600_a	5203408_c	1301216_c
26	5064454307840_a	1266113576960_a	42600857600_b	10636984320_b	665638400_b	166409600_b	41602400_b	19332032_c	4832144_c
27	21034728161280_a	5258682040320_a	164317593600_b	41092628480_b	1283731200_b	80241120_c	160466400_b	40116600_b	20064240_c
28	84130607923200_a	21032651980800_a	82158796800_c	41092628480_b	5134924800_b	1284618240_b	5015136_c	160466400_b	80241120_c
29	325290622451712_a	81326254325760_c	2541445447680_b	153384255488_b	19855042560_b	4965568512_b	19395928_c	4859904_a	155117520_b
30	1219947795578880_a	304986948894720_a	10165781790720_b	635297857536_b	74456409600_b	18618212352_b	620656128_b	18224640_a	1417920_a

Table 10.65: **Best lower bounds constructed for $A_4^{GC}(n,d,w)$ for linear codes for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	$\mathit{4}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	$\mathit{4}_a$	2_a	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	7_a	2_a	$\mathit{4}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	$\mathbf{15}_a$	2_a	$\mathit{4}_a$	$\mathit{3}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	$\mathit{60}_a$	2_a	2_a	3_a	$\mathit{4}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	100_a	8_a	2_a	3_a	4_a	2_a	-	-	-	-	-	-	-	-	-	-	-	-	-
18	$\mathbf{180}_a$	14_b	8_a	3_a	2_a	2_a	$\mathit{4}_a$	-	-	-	-	-	-	-	-	-	-	-	-
19	180_a	40_b	14_b	12_a	2_a	1_a	3_a	2_a	-	-	-	-	-	-	-	-	-	-	-
20	$\mathbf{520}_a$	144_b	44_a	12_b	10_a	1_a	2_a	2_a	$\mathit{4}_a$	-	-	-	-	-	-	-	-	-	-
21	2240_a	288_b	96_b	32_b	24_b	1_a	2_a	1_a	2_a	3_a	-	-	-	-	-	-	-	-	-
22	5856_a	351_b	384_b	96_b	42_b	6_a	2_a	1_a	2_a	3_a	3_a	-	-	-	-	-	-	-	-
23	21434_a	1399_b	198_b	336_b	168_b	6_a	6_a	2_b	2_a	2_a	3_a	2_a	-	-	-	-	-	-	-
24	85680_c	5572_b	752_b	149_b	672_b	18_b	6_a	6_a	2_a	2_a	3_a	2_a	3_a	-	-	-	-	-	-
25	326592_c	2585_b	1336_b	716_b	340_b	60_b	20_b	6_a	10_a	2_a	3_a	2_a	3_a	2_a	-	-	-	-	-
26	1211984_c	9992_b	4686_b	684_b	1313_b	20_b	64_b	20_b	10_a	2_a	3_a	2_a	3_a	-	2_a	-	-	-	-
27	5015136_c	39042_b	19404_b	2597_b	654_b	291_b	52_b	72_b	36_b	5_a	2_a	2_a	3_a	-	2_a	3_a	-	-	-
28	20060280_c	77976_b	19664_b	9702_b	2542_b	1120_b	162_b	64_b	126_b	5_a	8_a	2_a	3_a	-	-	3_a	2_a	-	-
29	77558760_a	301656_b	75480_b	18820_b	9672_b	623_b	175_b	158_b	42_b	20_b	8_a	8_a	-	-	-	-	-	2_a	-
30	290851224_c	1140255_b	284025_b	71350_b	18270_b	2500_b	632_b	-	156_b	68_b	30_a	4_b	2_b	2_b	2_b	2_b	2_b	2_b	3_a

Table 10.66: **Best lower bounds constructed for $A_4^{GC}(n,d,w)$ for linear codes for $12 \leq d \leq 30$**

n/d	3	4	5	6	7	8	9	10	11
4	$\mathit{6}_a$	1_a	-	-	-	-	-	-	-
5	4_a	$\mathit{3}_a$	-	-	-	-	-	-	-
6	27_a	$\mathit{16}_a$	-	1_a	-	-	-	-	-
7	35_a	22_a	-	1_a	-	-	-	-	-
8	280_a	108_a	22_a	$\mathit{12}_a$	-	1_a	-	-	-
9	252_a	126_a	22_a	8_a	2_b	1_a	-	-	-
10	3280_a	$\mathbf{860}_a$	$\mathbf{210}_a$	50_a	5_a	4_a	-	2_a	-
11	3656_a	920_a	226_a	56_a	5_a	4_a	-	$\mathit{2}_a$	-
12	$\mathbf{59136}_a$	$\mathbf{14784}_a$	$\mathbf{1848}_a$	$\mathit{924}_a$	84_a	24_a	6_a	-	$\mathit{2}_a$
13	54752_a	13688_b	1696_a	880_a	68_a	30_a	4_a	3_a	-
14	$\mathbf{768768}_{\mathbf{a}}$	$\mathit{192192}_a$	7424_a	$\mathbf{3712}_a$	$\mathbf{796}_a$	$\mathbf{208}_a$	12_a	$\mathbf{21}_a$	4_a
15	411560_a	213584_a	12835_a	$\mathbf{6648}_a$	802_a	$\mathbf{410}_a$	15_a	31_a	3_a
16	$\mathit{13174400}_a$	$\mathit{3293600}_a$	51608_a	$\mathbf{55424}_a$	$\mathbf{13856}_a$	$\mathbf{3776}_a$	194_a	26_b	$\mathbf{68}_a$
17	777640_a	3111120_a	$\mathbf{97520}_a$	48550_a	12120_a	$\mathit{6060}_a$	196_a	28_a	5_a
18	$\mathbf{44933184}_a$	$\mathbf{11266112}_a$	$\mathbf{699624}_a$	$\mathbf{699624}_a$	$\mathbf{43632}_a$	$\mathbf{10908}_a$	$\mathbf{2691}_a$	380_b	28_a
19	11824384_a	11266112_a	$\mathbf{1477796}_a$	$\mathbf{738772}_a$	$\mathbf{92252}_a$	$\mathbf{11542}_a$	2893_a	198_a	53_a
20	755728384_a	$\mathbf{189432064}_a$	$\mathbf{11822368}_a$	738520_a	$\mathbf{369008}_a$	$\mathit{184756}_a$	6012_a	1592_a	400_a
21	90291264_a	188416000_a	22573824_a	1412068_a	176772_a	45112_a	11148_a	2926_a	847_a
22	10602158336_a	2650495232_a	22607872_a	5643456_a	1410864_a	353496_a	88424_a	22088_a	5522_a
23	1384513088_a	670222080_a	43264648_a	10816624_a	2703694_a	676312_a	169182_a	42968_a	10701_a
24	177279886336_a	44319794176_a	346436544_a	43355616_a	21631400_a	5406464_a	1351616_a	338016_a	84964_a
25	21300369664_a	11204500480_a	10399676_a	1299844_a	41600552_a	10399676_a	2599688_a	649922_a	162986_a
26	2532157069312_a	633038608384_a	326893568_a	618544192_a	38656528_a	83204800_a	20801200_a	5199376_a	1299844_a
27	-	158680788992_a	-	20057442_a	-	300224_a	40114884_a	10029150_a	2506644_a
28	42061705248768_a	10515426312192_a	5154680832_a	10262347776_a	641396736_a	159987712_a	627776_a	80226336_a	20056584_a
29	-	2625500086272_a	-	6691200_a	-	3853632_a	-	-	38777664_a
30	609973884610560_a	152493461268480_a	80766566400_a	149011451520_a	9313176480_a	2332609440_a	9080016_a	9110544_a	708168_a

Table 10.67: **Best lower bounds constructed for $A_4^{GC,RC}(n,d,w)$ for linear codes for $3 \leq d \leq 11$**

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	1_a	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	3_a	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	$\mathit{24}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	26_a	-	-	-	$\mathit{2}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	16_a	5_b	4_b	-	$\mathit{2}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	28_a	-	-	3_b	2_a	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	$\mathit{179}_a$	47_b	16_b	5_a	$\mathbf{8}_a$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	357_a	-	3_a	10_b	6_a	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	2546_a	160_b	192_b	3_b	16_b	4_a	-	-	-	-	-	-	-	-	-	-	-	-	-
23	5312_a	-	-	24_b	10_b	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	42796_a	2744_b	48_b	10_b	320_b	8_b	4_a	-	-	-	-	-	-	-	-	-	-	-	-
25	81772_a	-	-	188_b	86_b	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	603734_a	4_a	605_b	157_b	80_a	4_a	8_a	-	6_a	-	-	-	-	-	-	-	-	-	-
27	1253105_a	-	3_a	-	8_a	-	3_a	-	-	-	-	-	-	-	-	-	-	-	-
28	10026576_a	-	9790_b	2420_b	-	-	28_a	-	8_a	-	4_a	-	-	-	-	-	-	-	-
29	19388832_a	-	20_a	-	24_a	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	281928_a	-	141810_b	35480_b	1104_a	-	48_a	-	58_a	-	12_a	-	6_a	-	-	-	-	-	-

Table 10.68: **Best lower bounds constructed for $A_4^{GC,RC}(n,d,w)$ for linear codes for $12 \leq d \leq 30$**

n/d	3	4	5	6	7	8	9	10	11
4	12	4	-	-	-	-	-	-	-
5	30_{gk}	10	3_{gk}	-	-	-	-	-	-
6	112_{gk}	40	8_{gk}	4	-	-	-	-	-
7	288_{m2}	78_{m2}	22_{gk}	7	3_{gk}	-	-	-	-
8	1056_{gk}	256_{gk}	63_{m2}	28	5_{gk}	4	-	-	
9	3012_{gk}	555_{gk}	134_{gk}	48_{m2}	18_{m2}	5_{gk}	3	-	-
10	10622_{gk}	2031_{m3}	504	144	34_{m1}	16	5_{gk}	4	-
11	32636_{gk}	7392	1848_{gk}	472	75_{m2}	32_{gk}	11_{m1}	4	3_{gk}
12	118272	29568	3696	1888	183_{m1}	118_{m2}	24_{m1}	9_{gk}	4_{gk}
13	473088_{gk}	109824_{gk}	8614_{gk}	3432	464	134_{gk}	46_{m1}	20_{gk}	8_{gk}
14	1537536	439296	28032	7424	1856	512	112_{gk}	49	17_{m2}
15	6589440	1647360	103680	27840	6960	1920	240	124	34_{m2}
16	26357760	6589440	414720	111360	27840	6680_{gk}	532_{gk}	177_{gk}	117_{gk}
17	105431040_{gk}	26357760_{gk}	1555840_{gk}	390080_{gk}	48620	24310	1530	380_{gk}	132_{gk}
18	210862080_{gk}	26357760_{gk}	5601024_{gk}	1400704_{gk}	97520	87516	5508	920_{gk}	282_{m2}
19	756760576_{gk}	94595072_{gk}	22404096_{gk}	5922048_{gk}	370128_{gk}	92378	7038_{gk}	2047_{m1}	615_{m1}
20	3027042304_{gk}	378896384	94595072_{gk}	23688192_{gk}	1478048	369512	24552	5918	2112
21	5778898944	1443692544	90295296	5641216	707168	516096	44764	11344	8064
22	21204434944	5301108736	90295296	20672512	2821728	707168	177352	45296	11324
23	88640061440	22160015360	174125056	21883904	5275648	2705248	676956	170044	42826
24	354439135232	88609783808	696500224	87005184	5437824	10816624	2704156	676956	170044
25	1362985222144	340746305536	2670166016	333684736	20855296	5244416	10400600	2600612	651028
26	5064454307840	1266113576960	10650386432	1237975040	77373440	19415040	168960	10400600	2599688
27	21034728161280	5258682040320	10268835840	5131714560	320732160	80056320	639232	319616	10028292
28	84130607923200	21032651980800	41049391104	20524695552	1282793472	319975424	2520576	1260288	40116600
29	325290622451712	81322655612928	158945771520	79360425984	4960026624	1237057536	9719808	4859904	330848
30	1219947795578880	304986948894720	161533132800	298023321600	18626457600	4665323520	18163200	18224640	1417920

Table 10.69: **Best known lower bounds for** $\max_w(A_4^{GC}(n,d,w))$ $3 \leq d \leq 11$

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	4	3_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	8_{gk}	4_{gk}	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	15	6_{gk}	4	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	60	12_{gk}	5_{gk}	4_{gk}	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	123_{gk}	24_{m2}	9_{gk}	5_{gk}	4	3_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-
18	180	46_{m2}	20_{m2}	9_{gk}	5_{gk}	4_{gk}	4	-	-	-	-	-	-	-	-	-	-	-	-
19	213_{m2}	83_{m2}	38_{m2}	17_{m2}	8_{gk}	5_{gk}	4_{gk}	3_{gk}	-	-	-	-	-	-	-	-	-	-	-
20	520	167_{m2}	69_{m2}	33_{m2}	16_{m2}	8_{gk}	5_{gk}	4_{gk}	4	-	-	-	-	-	-	-	-	-	-
21	2240	176_{m3}	86_{m2}	44_{m2}	22_{m2}	12_{m2}	7_{m2}	4_{m2}	4_{m2}	3	-	-	-	-	-	-	-	-	-
22	5856	348_{m3}	148_{m2}	72_{m2}	40_{m2}	20_{m2}	12_{m2}	6_{m2}	4_{m2}	4_{m2}	4_{m2}	-	-	-	-	-	-	-	-
23	21434	672_{m3}	252_{m2}	114_{m2}	62_{m2}	32_{m2}	17_{m2}	9_{m2}	5_{m2}	4_{m2}	4_{m2}	3_{m2}	-	-	-	-	-	-	-
24	85652	1380_{m3}	488_{m3}	204_{m2}	102_{m2}	54_{m2}	28_{m2}	15_{m2}	9_{m2}	5_{m2}	4_{m2}	4_{m2}	4_{m2}	-	-	-	-	-	-
25	326088	2804_{m3}	960_{m3}	380_{m3}	166_{m3}	130_{m3}	46_{m2}	24_{m2}	13_{m2}	8_{m2}	5_{m2}	4_{m2}	4_{m2}	3_{m2}	-	-	-	-	-
26	1208316	5948_{m3}	1954_{m3}	702_{m3}	296_{m3}	134_{m2}	76_{m2}	40_{m2}	22_{m2}	12_{m2}	8_{m2}	5_{m2}	4_{m2}	4_{m2}	4_{m2}	-	-	-	-
27	5013288	12616_{m3}	3854_{m3}	1310_{m3}	524_{m3}	228_{m2}	112_{m2}	64_{m2}	36_{m2}	19_{m2}	11_{m2}	8_{m2}	5_{m2}	4_{m2}	4_{m2}	3	-	-	-
28	20056584	27376_{m3}	9472	2620_{m3}	918_{m3}	388_{m2}	186_{m2}	100_{m2}	56_{m2}	30_{m2}	17_{m2}	10_{m2}	8_{m2}	5_{m2}	4_{m2}	4_{m2}	4_{m2}	-	-
29	77558760	54752_{m3}	16490_{m3}	5198_{m3}	1796_{m3}	706_{m3}	310_{m2}	144_{m2}	84_{m2}	48_{m2}	26_{m2}	15_{m2}	9_{m2}	6_{m2}	5_{m2}	4_{m2}	4_{m2}	3_{m2}	-
30	290845350	122540_{m3}	35354_{m3}	10852_{m3}	3534_{m3}	1092_{m2}	532_{m2}	254_{m2}	130_{m2}	72_{m2}	41_{m2}	23_{m2}	14_{m2}	9_{m2}	6_{m2}	5_{m2}	4_{m2}	4_{m2}	4_{m2}

Table 10.70: **Best known lower bounds for** $\max_w(A_4^{GC}(n,d,w))$ $12 \leq d \leq 30$

n/d	3	4	5	6	7	8	9	10	11
4	6	2_{gk}	-	-	-	-	-	-	-
5	15_{gk}	3	1_{gk}	-	-	-	-	-	-
6	44_{gk}	16	4_{gk}	2_{gk}	-	-	-	-	-
7	135_{gk}	36_{gk}	11_{gk}	2_{gk}	1_{gk}	-	-	-	-
8	528_{gk}	128_{gk}	28_{gk}	12	2_{gk}	2_{gk}	-	-	-
9	1354_{gk}	275_{gk}	67_{gk}	21_{m3}	8_{gk}	2_{gk}	1_{gk}	-	-
10	4542_{gk}	860	210	54_{gk}	17_{m3}	8_{gk}	2_{gk}	2_{gk}	-
11	14405_{gk}	2457_{gk}	477_{gk}	117_{gk}	37_{m1}	14_{m1}	5_{gk}	2	1_{gk}
12	59136	14784	1848	924	87_{m1}	29_{m1}	12_{m3}	4_{gk}	2
13	167263_{gk}	27376_{gk}	3974_{m1}	924_{gk}	206_{m1}	62_{m1}	23_{m3}	10_{m3}	4_{gk}
14	768768	192192	11878_{gk}	3712	796	208	49_{m3}	21	8_{m1}
15	1646240_{gk}	411821_{gk}	25670_{gk}	6648	1600_{gk}	410	109_{m3}	37_{m1}	18_{gk}
16	13174400	3293600	55376_{gk}	55424	13856	3776	243_{m3}	83_{m3}	68
17	26355520_{gk}	6587200_{gk}	97520	97450_{gk}	12864_{gk}	6060	579_{m1}	175_{m3}	62_{m3}
18	44933184	11266112	699624	699624	43632	10908	2691	407_{m1}	133_{m3}
19	47102080_{gk}	23647760_{gk}	1477796	738772	92252	11542	3678_{m3}	960_{m1}	285_{m1}
20	756760576_{gk}	189432064	11822368	11806240_{gk}	369008	184756	11452_{gk}	2868_{gk}	766_{gk}
21	90291264	188416000	22573824	1412068	176772	45112	11148	2926	847
22	10602158336	2650495232	22607872	5643456	1410864	353496	88424	22088	5522
23	1384513088	670222080	43264648	10816624	2703694	676312	169182	42968	10701
24	177279886336	44319794176	346436544	43355616	21631400	5406464	1351616	338016	84964
25	21300369664	11204500480	10399676	1299844	41600552	10399676	2599688	649922	162986
26	2532157069312	633038608384	326893568	618544192	38656528	83204800	20801200	5199376	1299844
27	-	158680788992	-	20057442	-	300224	40114884	10029150	2506644
28	42061705248768	10515426312192	5154680832	10262347776	641396736	159987712	627776	80226336	20056584
29	-	2625500086272	-	6691200	-	3853632	-	-	38777664
30	609973884610560	152493461268480	80766566400	149011451520	9313176480	2332609440	9080016	9110544	708168

Table 10.71: **Best known lower bounds for** $\max_w(A_4^{GC,RC}(n,d,w))$ $3 \leq d \leq 11$

n/d	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
12	2_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	2_{gk}	1_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	4_{gk}	2_{gk}	2_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	6_{gk}	3_{gk}	2_{gk}	1_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	24	5_{gk}	2_{gk}	2_{gk}	2_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	30_{gk}	12_{m3}	4_{gk}	2_{gk}	2	1_{gk}	-	-	-	-	-	-	-	-	-	-	-	-	-
18	49_{m3}	21_{m3}	10_{m3}	4_{gk}	2	2_{gk}	2_{gk}	–	-	-	-	-	-	-	-	-	-	-	-
19	99_{m3}	39_{m3}	18_{m3}	8_{m3}	4_{gk}	2_{gk}	2_{gk}	1_{gk}	-	-	-	-	-	-	-	-	-	-	-
20	179	77_{m3}	33_{m3}	15_{m3}	8	4_{gk}	2_{gk}	2_{gk}	2_{gk}	-	-	-	-	-	-	-	-	-	-
21	357	88_{m3}	43_{m3}	22_{m3}	11_{m3}	6_{m3}	3_{m3}	2_{m3}	2_{m3}	1_{m3}	-	-	-	-	-	-	-	-	-
22	2546	174_{m3}	192_{L1}	36_{m3}	20_{m3}	10_{m3}	6_{m3}	2_{m3}	2_{m3}	2_{m3}	2_{m3}	-	-	-	-	-	-	-	-
23	5312	336_{m3}	126_{m3}	57_{m3}	31_{m3}	16_{m3}	8_{m3}	4_{m3}	2_{m3}	2_{m3}	2_{m3}	1_{m3}	-	-	-	-	-	-	-
24	42796	2744_{L1}	244_{m3}	102_{m3}	320_{L1}	27_{m3}	14_{m3}	7_{m3}	4_{m3}	2_{m3}	2_{m3}	2_{m3}	2_{m3}	-	-	-	-	-	-
25	81772	1402_{m3}	480_{m3}	190_{m3}	86_{L1}	65_{m3}	23_{m3}	12_{m3}	6_{m3}	4_{m3}	2_{m3}	2_{m3}	2_{m3}	1_{m3}	-	-	-	-	-
26	603734	2974_{m3}	977_{m3}	351_{m3}	148_{m3}	67_{m3}	38_{m3}	20_{m3}	11_{m3}	6_{m3}	4_{m3}	2_{m3}	2_{m3}	2_{m3}	2_{m3}	-	-	-	-
27	1253105	6308_{m3}	1927_{m3}	655_{m3}	262_{m3}	114_{m3}	56_{m3}	32_{m3}	18_{m3}	9_{m3}	5_{m3}	4_{m3}	2_{m3}	2_{m3}	2_{m3}	1_{m3}	-	-	-
28	10026576	13688_{m3}	9790_{L2}	2420_{L2}	459_{m3}	194_{m3}	93_{m3}	50_{m3}	28_{m3}	15_{m3}	8_{m3}	4_{m3}	4_{m3}	2_{m3}	2_{m3}	2_{m3}	2_{m3}	-	-
29	19388832	29292_{m3}	8245_{m3}	2599_{m3}	898_{m3}	353_{m3}	155_{m3}	77_{m3}	42_{m3}	24_{m3}	12_{m3}	7_{m3}	4_{m3}	3_{m3}	2_{m3}	2_{m3}	2_{m3}	1_{m3}	-
30	281928	61270_{m3}	141810_{L1}	35480_{L1}	1767_{m3}	546_{m3}	266_{m3}	127_{m3}	65_{m3}	36_{m3}	20_{m3}	11_{m3}	7_{m3}	4_{m3}	3_{m3}	2_{m3}	2_{m3}	2_{m3}	2_{m3}

Table 10.72: **Best known lower bounds for** $\max_w(A_4^{GC,RC}(n,d,w))$ $12 \leq d \leq 30$