# Computing the Minimum Hamming Distance for $\mathbb{Z}_2\mathbb{Z}_4$-Linear Codes

Marta Pujol and Mercè Villanueva

Combinatorics, Coding and Security Group (CCSG)
Universitat Autònoma de Barcelona (UAB)
VIII JMDA, Almería

11-13 July 2012

# Outline

# Introduction

Error-correcting code

Examples:

Linear code

$$\begin{array}{ccc} 0 & \longrightarrow & 000 \\ 1 & \longrightarrow & 111 \end{array}$$

Nonlinear code

$$\begin{array}{ccc} 0 & \longrightarrow & 101 \\ 1 & \longrightarrow & 011 \end{array}$$

Linear code

$$\begin{array}{ccc} 00 & \longrightarrow & 00000 \\ 01 & \longrightarrow & 01101 \\ 10 & \longrightarrow & 10110 \\ 11 & \longrightarrow & 11011 \end{array}$$

# Introduction

- A **binary code** $C$ is a subset of binary vectors of length $n$, $C \subset \mathbb{Z}_2^n$.
- The elements of a code are called codewords.
- A subgroup of $\mathbb{Z}_2^n$ is called a **binary linear code**.
- Let $M$ be the number of codewords.
  If $C$ is a binary linear code of dimension $k$, $M = 2^k$.
- Hamming distance / weight.
- Minimum Hamming distance $d_H(C)$ / weight $\omega_H(C)$.
- Error correcting capability: $e = \lfloor (d_H(C) - 1)/2 \rfloor$.
- Transmission rate: $\log_2 M/n$.

**Objective:**

The aim of this research is to study the algorithms to compute the minimum distance for binary linear codes and quaternary codes and develop new algorithms for $\mathbb{Z}_2\mathbb{Z}_4$-additive codes using them as a reference.

# Introduction

Let $C$ be the binary linear code of length $5$:

$$\begin{array}{cc} 00000 & \mathbf{11011} \\ \mathbf{10110} & 01101 \end{array} \qquad \mathcal{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Generator matrix of a binary linear code in standard form:

$$\mathcal{G}_s = \left( \begin{array}{c|c} I_k & A \end{array} \right).$$

$$\mathcal{G}_s = \begin{pmatrix} \mathbf{1} & \mathbf{0} & 1 & 1 & 0 \\ \mathbf{0} & \mathbf{1} & 1 & 0 & 1 \end{pmatrix}.$$

- The number of codewords is $M = 2^2 = 4$.
- The dimension is $k = 2$.
- The minimum Hamming distance is $d_H(C) = 3$.
- The error correcting capability is $e = \lfloor (3-1)/2 \rfloor = 1$.

# Quaternary Linear Codes

- A **quaternary code** $\mathcal{C}$ is a subset of quaternary words of length $\beta$, $\mathcal{C} \subset \mathbb{Z}_4^\beta$.
- A codeword of a quaternary code contains 0, 1, 2 and 3.
- A subgroup of $\mathbb{Z}_4^\beta$ is called a **quaternary linear code**.
- The type of a quaternary linear code is $2^\gamma 4^\delta$.
- Lee weight of a coordinate of a quaternary word:
$$\omega_L(0) = 0,\ \omega_L(1) = \omega_L(3) = 1,\ \omega_L(2) = 2.$$
- Lee distance: $d_L(v, w) = \omega_L(v - w)$.
- Minimum Lee distance $d_L(\mathcal{C})$ / weight $\omega_L(\mathcal{C})$.

# Quaternary Linear Codes. Example

Let $\mathcal{C}$ be the quaternary linear code of length $4$:

| | | |
|---|---|---|
| **2110** | 2330 | 0220 |
| **1101** | 3303 | 2202 |
| 3211 | 1233 | 2022 |
| 1321 | 3123 | 0000 |
| 0312 | 0132 | |
| 1013 | 3031 | |

$$\mathcal{G} = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

The generator matrix in standard form is:

$$\mathcal{G}_s = \left(\begin{array}{ccc} 2T & 2I_\gamma & \mathbf{0} \\ \hline S & R & I_\delta \end{array}\right) \qquad \mathcal{G}_s = \begin{pmatrix} 2 & 1 & \mathbf{1} & \mathbf{0} \\ 1 & 1 & \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

- The number of codewords is $M = 2^0 4^2 = 16$.
- The type is $2^0 4^2$.
- The minimum Lee distance is $d_L(\mathcal{C}) = 3$.
- The error correcting capability is $e = \lfloor (3-1)/2 \rfloor = 1$.

# Quaternary Linear Codes. $\mathbb{Z}_4$-Linear Codes

Quaternary linear codes can be viewed as binary (nonlinear) codes, using in each coordinate the Gray map: $\varphi : \mathbb{Z}_4 \to \mathbb{Z}_2^2$ defined as

$$\varphi(0) = 00, \quad \varphi(1) = 01, \quad \varphi(2) = 11, \quad \varphi(3) = 10.$$

The corresponding binary code $C = \phi(\mathcal{C})$ is called $\mathbb{Z}_4$-**linear code**.

Example:

$$\begin{array}{ccc} \mathbb{Z}_4^n & \xrightarrow{\phi} & \mathbb{Z}_2^{2n} \\ 2110 & & 11\ 01\ 01\ 00 \\ 1101 & & 01\ 01\ 00\ 01 \\ 3211 & & 10\ 11\ 01\ 01 \\ \vdots & & \vdots \\ \text{Quaternary linear code} & & \mathbb{Z}_4\text{-linear code} \\ \text{Lee distance} & & \text{Hamming distance} \end{array}$$

The minimum Lee distance of a quaternary linear code $\mathcal{C}$ is equal to the minimum Hamming distance of the $\mathbb{Z}_4$-linear code $C = \phi(\mathcal{C})$.

# $\mathbb{Z}_2\mathbb{Z}_4$-Additive Codes

- A subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ is called a $\mathbb{Z}_2\mathbb{Z}_4$-**additive code**. Note that the first $\alpha$ coordinates are in $\mathbb{Z}_2$ and the last $\beta$ in $\mathbb{Z}_4$.
- The type is $(\alpha, \beta; \gamma, \delta; \kappa)$, where $\gamma$ and $\delta$ are the min. number of generators of order 2 and 4, resp.
- Lee distance/weight: Hamming distance/weight in the $\alpha$ coordinates plus Lee distance/weight in the $\beta$ coordinates.
- Minimum Lee distance $d_L(\mathcal{C})$ / weight $\omega_L(\mathcal{C})$.
- Using the Gray map in the $\mathbb{Z}_4$ coordinates, they can also be seen as binary (nonlinear) codes, called $\mathbb{Z}_2\mathbb{Z}_4$-**linear codes**.
- Some nonlinear codes ($\mathbb{Z}_4$-linear or $\mathbb{Z}_2\mathbb{Z}_4$-linear codes) are better than any linear code.

# $\mathbb{Z}_2\mathbb{Z}_4$-Additive Codes. Example

Let $\mathcal{C}$ be a $\mathbb{Z}_2\mathbb{Z}_4$-additive code generated by

$$\mathcal{G} = \left(\begin{array}{cc|cccc} 1 & 0 & 2 & 0 & 2 & 0 \\ \hline 1 & 1 & 2 & 2 & 1 & 1 \end{array}\right).$$

The generator matrix in standard form is:

$$\mathcal{G}_s = \left(\begin{array}{cc|ccc} I_\kappa & T_b & 2T_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2T_1 & 2I_{\gamma-\kappa} & \mathbf{0} \\ \hline \mathbf{0} & S_b & S_q & R & I_\delta \end{array}\right) \qquad \mathcal{G}_s = \left(\begin{array}{cc|cccc} \mathbf{1} & 0 & 2 & 0 & 2 & 0 \\ \hline 0 & 1 & 0 & 2 & 3 & \mathbf{1} \end{array}\right).$$

- The number of codewords is $M = 4^1 2^1$, so $\gamma = 1$ and $\delta = 1$.
- The binary coordinates are $X = \{1, 2\}$ and the quaternary ones are $Y = \{3, 4, 5, 6\}$, so $\alpha = 2$ and $\beta = 4$.
- The type of the code is $(2, 4; 1, 1; 1)$.
- The minimum Lee distance is $d_L(\mathcal{C}) = 4$.
- The error correcting capability is $e = \lfloor (4-1)/2 \rfloor = 1$.

# $\mathbb{Z}_2\mathbb{Z}_4$-Additive Codes. $\mathbb{Z}_2\mathbb{Z}_4$-Linear Codes

- These codes have been studied by our research group (CCSG).
- There exists a package for this type of codes that will be integrated in MAGMA. With the implementation of these functions, the package will be completed.
- MAGMA:
  - Private license software (1993), developed by the Computational Algebra Group in Sydney University.
  - Software large package, computationally solve difficult problems in algebra, coding theory, and combinatorics.
  - Many MAGMA functions are implemented in C language.

# Minimum Weight and Distance

- Distance invariant: minimum distance = minimum weight.
- It is easier to compute the minimum weight.
- **Binary codes:**
  - Brute Force: small codes
  - Brouwer-Zimmerman
  - Probabilistic algorithms
- $\mathbb{Z}_4$**-linear codes:**
  - Brute Force: small codes
  - Adaptation of Brouwer-Zimmerman
- $\mathbb{Z}_2\mathbb{Z}_4$**-linear codes:** There was no implementation. The aim of this research is to study different algorithms and implement them in MAGMA using the existing ones as references.

# Algorithms Implemented for $\mathbb{Z}_2\mathbb{Z}_4$-Linear Codes

- **Brute Force Adding Bounds:** It generates all linear combinations between the rows of the generator matrix. Improvement of the algorithm adding a lower bound and an upper bound.
- **Kernel-Leaders (nonlinear):** Any $\mathbb{Z}_2\mathbb{Z}_4$-linear code can also be seen as a binary (nonlinear) code. It can be represented as the union of cosets of a binary linear code denoted by $K(C)$:

$$C = \bigcup_{i=0}^{t} \Big( K(C) + c_i \Big),$$

The same techniques used in general for binary nonlinear codes can be used to compute the minimum weight of $\mathbb{Z}_2\mathbb{Z}_4$-linear codes.

# Algorithms Implemented for $\mathbb{Z}_2\mathbb{Z}_4$-Linear Codes

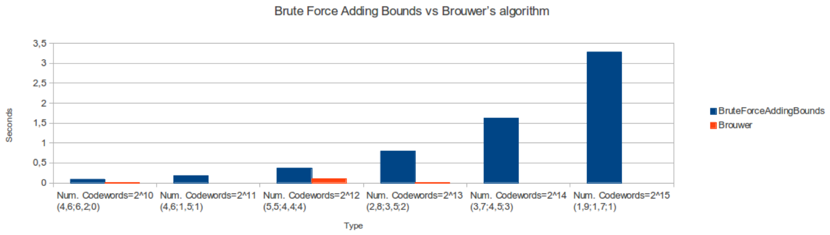
- **Brouwer:**
  - It is an adaptation of Brouwer algorithm for binary linear and quaternary linear codes.
  - It uses several generator matrices in standard form.
  - The columns used in the information set of one matrix cannot be used in another standard form generator matrix.
- **Zimmerman:**
  - The next algorithm that will be implemented.
  - Similar to Brouwer algorithm.
  - The columns used in the information set of one matrix can be used again in another standard form generator matrix.

# Types of Testing

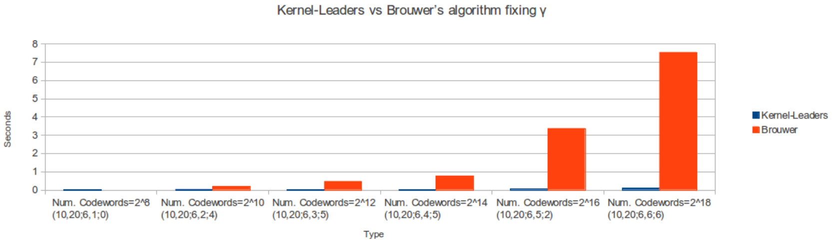
- **Black Box tests:** They take into account the expected result. There is an exhaustive analisys of the requirements and functionalities and, with this information, the tests are designed.

- **Performance test:** They are designed to see how much time needs the function to obtain the results. Doing this type of test, we optimized some parts of the implementation.

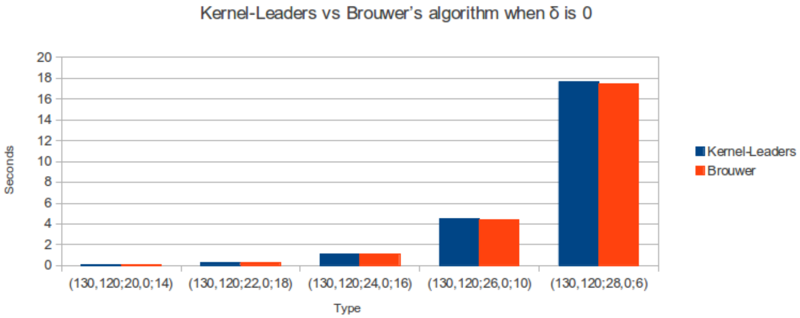
# Performance Test 1



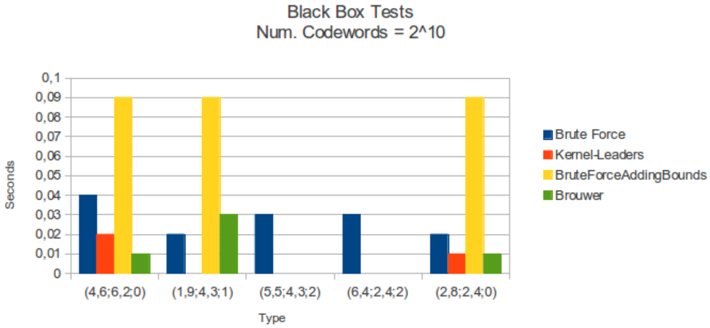
# Performance Test 2



# Performance Test 3



# Performance Test 4



# Conclusions

- Four algorithms for $\mathbb{Z}_2\mathbb{Z}_4$-linear codes: Brute Force, Kernel-Leader, Brute Force Adding Bounds, and Brouwer.
- A unifying function needs to be implemented.
- The final function that computes the minimum Hamming distance for $\mathbb{Z}_2\mathbb{Z}_4$-linear codes is basic for implementing other functions that complete the current package for $\mathbb{Z}_2\mathbb{Z}_4$-linear codes. Then, this package will have the same functionality as the existing package for binary linear codes that is in MAGMA.

# Future Work

- Improve the performance of the functions using Brouwer-Zimmerman algorithm.
- Study which is the best algorithm depending on the parameters of the given $\mathbb{Z}_2\mathbb{Z}_4$-linear code. The main function should select the best to apply in each situation.
- Develop the remaining functions related to the minimum distance, to complete the package on $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.
- Study new theoretical results to improve the performance of these functions. Improve the current functions in MAGMA to compute the minimum weight of a $\mathbb{Z}_4$-linear code.
- Apply these functions to find new $\mathbb{Z}_2\mathbb{Z}_4$-linear optimal codes.

- J. Borges, C. Fernández, J. Pujol, J. Rifà and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-linear codes: generator matrices and duality," *Designs, Codes and Cryptography*, vol. 54, pp. 167-179, 2010.
- J. Borges, C. Fernández, J. Pujol, J. Rifà and M. Villanueva, "$\mathbb{Z}_2\mathbb{Z}_4$-linear codes. A MAGMA package," Universitat Autònoma de Barcelona, 2007.
- J. J. Cannon and W. Bosma (Eds.), *Handbook of* MAGMA *Functions*, Edition 2.13, 2006, (`http://magma.maths.usyd.edu.au/magma/`)
- A. Foster, "A polynomial-time probabilistic algorithm for the min. distance of an arbitrary linear error-correcting code", Math. Honor Report, 2004.
- J. Pujol, M. Villanueva and F. Zeng, "Minimum Distance of Binary Nonlinear Codes," in *Proc. 18th International Conference on Applications of Computer Algebra*, Sofia, Bulgaria, June 2012.
- Z.-X. Wan, *Quaternary Codes*, World Scientific, 1997.
- G. White, "Enumeration-based Algorithms in Coding Theory," PhD Thesis, University of Sydney, 2006.

# Thank you