Future Needs on Computational Techniques in Algebraic Coding Theory and Related Topics

CESGA HPCN 2010

Edgar Martínez Moro

edgar@maf.uva.es

SINGACOM Computing Team

University of Valladolid, Spain

http://www.singacom.uva.es/

Nov. 25, 2010

Abstract

The following position paper has been prepared by a group of six mathematicians whose core areas are in Computer Algebra and Algebraic Geometry and all of them concerned with research topics in the area of Algebraic Coding Theory and Related Areas (ACT&RA) including Cryptography.

The idea of writing this work was originated at the *Soria Summer School on Computational Mathematics: Algebraic Geometric Modeling in Information Theory* hosted by SINGACOM group at Campus of Soria of the Universidad de Valladolid (Spain) and it collects the ideas of the authors and some other participants that where discussed in the school about the future needs of computational techniques and resources in Coding Theory. This report is a compilation of such discussions, thoughts and needs just through a case study on one of the main problems in coding theory.

The team

- **Stanislav Bulygin** Center for Advanced Security Research Darmstadt.
- **Fernando Hernando** School of Mathematical Sciences, University College Cork and SINGACOM computing Team.
- **David Joyner**, Mathematics Department, U.S. Naval Academy.
- **Irene Marquez-Corbella, Edgar Martínez-Moro, Carlos Munuera** SINGACOM computing Team, Universidad de Valladolid.
- **Diego Ruano** Department of Mathematical Sciences, Aalborg University and SINGACOM computing Team .

Overview

Abstract

The team

Computing and ACT&RA research

- Computing and ACT&RA research

A case study on the first ACT&RA problem

- A case study on the first ACT&RA problem

Computational needs

- Computational needs

Acknowledgements

- Acknowledgements

Computing and ACT&RA research

Computing and ACT&RA research

Coding Theory and the object of its study (codes, decoding and encoding procedures-algorithms) are responsible of the protocols that ensure reliable communications. Moreover Coding Theory is also used for data compression, cryptography, network coding etc. In Coding Theory various scientific disciplines are involved, such as information theory, electrical engineering, mathematics, statistics, computer science etc.

Computing and ACT&RA research

Coding Theory and the object of its study (codes, decoding and encoding procedures-algorithms) are responsible of the protocols that ensure reliable communications. Moreover Coding Theory is also used for data compression, cryptography, network coding etc. In Coding Theory various scientific disciplines are involved, such as information theory, electrical engineering, mathematics, statistics, computer science etc.

We will focus on those problems in ACT&RA related to mathematics. It is well known that Mathematics and especially Algebra in a broad sense, Combinatorics and Geometry play an important role in many facets of Coding Theory, even some of the topics in Algebraic Coding Theory have become mathematical disciplines on their own besides the original purpose of designing protocols that ensure reliable communications.

Computing and ACT&RA research

Also recently Coding Theory has gained a new role in cryptography due to the Public Key Cryptosystem (PKC) of McEliece and Niederreiter based on the hardness of decoding arbitrary codes. This PKC has become an alternative, together with other cryptosystems based on lattices, hash functions, and multivariate polynomials, to classical Number Theory systems (like RSA, ECC). It is known that Shor's algorithm for a quantum computer is able to break these classical cryptosystems. This means that if a large enough quantum computer is built, all widely used classical cryptosystems will be broken. Therefore, having such alternatives in a "post-quantum age" is vital.

Computing and ACT&RA research

Researches in ACT&RA use computers and computational techniques in at least the following situations and in several levels of intensity (programming, users, database search etc.)

1.- **As Computational Tools**. There are many tools used by the ACT&RA community going from research-written programs in C++, Phyton etc. to libraries included in software packages and systems for Computer Algebra such as SINGULAR, PolyBoRi, Magma, CoCoA, Macaulay2, Maple, Asir/Risa, GAP etc.

Computing and ACT&RA research

2.-

For Searching and discovering good structures. Note that for the construction of (linear) algebraic codes as well as encoding and decoding procedures several algebraic structures are used coming from different mathematical topics: group theory, graph theory, linear algebra, combinatorial designs etc. Moreover this algebraic structure provides a rich number of properties to the constructed objects usually reflected in the automorphism group of the code. These different types of constructions usually make the task of classification, comparison and search through different type of codes difficult. In fact, finding a canonical form, standard generation and isomorphism testing are hard task in coding theory.

Computing and ACT&RA research

3.-

In Database Search. In connection with the previous item, ones a code has been proposed it is important to check whether this code has been studied before (in its actual form or derived from other techniques) as well as know how good are its parameters compared to other codes of the same length and dimension. There are some nice efforts in this direction based on Brouwer's tables but as pointed before there is a lack of a canonical form for a code and isomorphism testing.

A case study on the first ACT&RA problem

A case study on the first ACT&RA problem

There are many computational challenging problems in coding theory, but we will focus on the first problem addressed in coding theory, namely *complete decoding problem* (CDP).

Given an $[n, k]_q$ code $\mathcal{C}$, that is a linear subspace of $\mathbb{F}_q^n$ of dimension k, and let $\mathbf{r}$ be a received word in $\mathbb{F}_q^n$ the CDP addresses to determine a codeword $\mathbf{c} \in \mathcal{C}$ that is closest to $\mathbf{r}$ using the Hamming distance (i.e. the number of positions ...). The t*-bounded distance decoding problem* (t-BDP) is to determine a codeword $\mathbf{c} \in \mathcal{C}$ such that $d(\mathbf{r}, \mathbf{c}) \leq t$ if such codeword exists.

A case study on the first ACT&RA problem

Note that if $t = \lfloor \frac{(d-1)}{2} \rfloor$ then the solution of the t-BDP is unique and if $t = \rho$ the covering radius then t-BDP is the same as CDP. Both problems are quite related to the *coset weights problem* (t-CWP) that can be stated as follows,

Given a binary $r \times n$ matrix and an r-dimensional vector $\mathbf{s}$ and $t \in \mathbb{Z}_{\geq 0}$, does a binary vector $\mathbf{e} \in \mathbb{F}_q^n$ exist such that $w(\mathbf{e}) \leq t$ and $H\mathbf{e} = \mathbf{s}$?

All these problems (CDP, t-BDP, t-CWP) have been shown to be NP-complete even if preprocessing is allowed!!!

A case study on the first ACT&RA problem

The complete decoding problem plays a central role in ACT&RA, let’s point some related problems in algebraic coding theory:

A case study on the first ACT&RA problem

Gradient descent complete decoding

(GDCD) The gradient descent complete decoding can be seen as syndrome decoding in which the syndrome look up is broken into smaller steps. Suppose we receive the word $\mathbf{m}$, the gradient function allows the decoder to first find a Hamming neighbor $\mathbf{m}'$ of $\mathbf{m}$ (Hamming distance $d(\mathbf{m}', \mathbf{m}) = 1$) such that

$$wt(\upharpoonleft \mathbf{m}') < wt(\upharpoonleft \mathbf{m}),$$

where $wt(\upharpoonleft \mathbf{m})$ denotes the least Hamming weight in the coset $\mathbf{m} + \mathcal{C}$. Then one replaces $\mathbf{m}$ with $\mathbf{m}'$ and iterates until $wt(\mathbf{m}) = 0$. Thus there is no need to identify the actual error $\mathbf{e}$.

A case study on the first ACT&RA problem

For understanding the next GDD algorithm we will need some knowledge of minimal (support) codewords. The *support of a codeword* $\mathbf{c} \in \mathcal{C}$ will be the set of its non-zero positions, i.e. $\mathrm{supp}(\mathbf{c}) = \{i \mid \mathbf{c}_i \neq 0\}$. A codeword $\mathbf{m}$ in the code $\mathcal{C}$ is said to be **minimal** if there is no other codeword $\mathbf{c} \in \mathcal{C}$ such that

$$\mathrm{supp}(\mathbf{c}) \subseteq \mathrm{supp}(\mathbf{m}).$$

There is another gradient-like decoding algorithm or *test set GDDA* that instead of reducing the codeword moving from one codes to another, in each step we subtract a codeword (ie. we stay in the same coset) such that it is a minimal codeword untill we arrive to a coset leader (a word in the coset with minimal Hamming weight). A minimal set of minimal codewords that accomplishes such a reduction is called a **minimal test set**.

A case study on the first ACT&RA problem

Recently it has been proven that both procedures are equivalent and also equivalent to some modular integer programming problem. Unfortunately, even if it is given an algorithm for computing a test set and also one for computing the minimal codewords the first one relies on a “clever” Groebner basis computation (that is both time and space consuming), and the second one is based on a Graver basis computation from the Groebner basis, thus again with great time and space consumption!!!

A case study on the first ACT&RA problem

Isomorphism testing of linear codes

There are some procedures for testing if two linear codes are isomorphic or not, the most well known and implemented procedure correspond to Leon's algorithm. The isomorphism of the code translated to the isomorphism of the Groebner presentation associated to the code (as a monoid) which is, in some sense, an invariant associated to the gradient descent complete decoding of it. Thus again both problems seem to be linked and their answer should need at least the same computation consumption. Note that both, from Leon's approach and the second approach there is only a test on linear isometry of two codes, but not a canonical form, this is proposed by T. Feulner.

McEliece cryptosystems

McEliece public key cryptosystem is based on hardness of decoding linear codes. It is defined as follows. For the key pair generation one first chooses a code C that has an efficient decoding algorithm that can correct up to $t \leq (d(C)-1)/2$ errors. Let G be a generator matrix of this code. Now one chooses random invertible matrix S and a random permutation matrix P and computes $G' = SGP$, where G' has the same dimensions as G and is a generator matrix of a code C' equivalent to C.

A case study on the first ACT&RA problem

The public key is (G', t), the private key is (S, G, P). In order to encrypt a message m for Bob, Alice takes Bob's public key G', encodes the message into C' and adds some random errors: $c = mG' + e, wt(e) \leq t$. The c is a ciphertext communicated to Bob. After receiving c Bob computes $cP^{-1} = mG'P^{-1} + eP^{-1} = mSG + eP^{-1}$. Since $wt(eP^{-1}) = wt(e)$, Bob may use the efficient decoder supplied with the code C to correct up to t errors and recover mS. After that he trivially recovers m.

A case study on the first ACT&RA problem

Now having just G', an attacker is supposed to do nothing better than to try to correct t errors with G' in order to recover m. If G' is “random enough” this is a hard problem. The original McEliece cryptosystem uses Goppa codes as suppliers for the code C. It is still unbroken up to date. Results on breaking certain parameter choices rely on the advances in general decoding algorithms, such as Information Set Decoding. There are other methods for general decoding of codes, e.g. the one based on solving quadratic systems with Gröbner bases.

A case study on the first ACT&RA problem

It is to be noted that the key sizes of McEliece are quite large compared to classical cryptosystems such as RSA. There were quite a few attempts to reduce the key sizes, both public and private. Different cryptanalytic techniques were used to break many of those proposals. A remarkable result is the one of Faugere, where polynomial systems solving was used to break certain versions on McEliece with the reduced public key sizes. The method there relies on efficient Gröbner basis computations. It would be an interesting challenge to investigate, how the methods along the lines of Faugere, could improve on cryptanalysing more schemes, thus providing better understanding of security of the McEliece.

Secret Sharing

A secret sharing scheme is a method of sharing a secret s among a set $\{P_1, P_2, \ldots, P_n\}$ of participants. The secret is assumed to be an element of a finite field. A dealer gives to each participant P_i some partial information on s, the share. Shares are secretly computed and distributed. In a later time some coalitions of participants can compute the secret as a function of the shares they collectively hold. Secret sharing schemes are used in key management, visual cryptography, electronic voting, steganography, etc.

A case study on the first ACT&RA problem

We can distinguish two main elements: the access structure, that is the set of coalitions able to determine the secret; and the sharing scheme, which is the way of computing the shares realizing the access structure. One of the main open problems in secret sharing is the classification of all possible access structures (up to isomorphism). Given its combinatorial nature, solving this problem requires a great computational capacity. To the date, only structures up to 5 participants have been classified.

A case study on the first ACT&RA problem

Linear error-correcting codes can be used to share a secret. By using a linear code $\mathcal{C}$, the corresponding access structure is determined by the minimal the codewords in the dual of $\mathcal{C}$, and the obtained scheme is very efficient, including the ability to detect cheaters. This leads to the problem of determining the set of minimal codewords of a linear code, which has already been cited before in relation to the gradient descent decoding. Computing the set of minimal codewords of a linear code is a hard problem in general, and it requires a great amount of computation.

Computational needs

Computational needs

Based on the problems stated before we can state the following challenge problems on Future on Computational Techniques in Algebraic Coding Theory.

High parallelization computing

Even for relatively modest size problems, determining the minimal codewords of a code, or equivalently a test set of the code or a gradient decoding function becomes very difficult by Groebner basis techniques even by traditional branch and cut techniques. One hope to tackle this obstacle is trying to parallelize the procedure using the matroid decomposition since it is well known that minimal codewords have a direct translation in $\mathbb{F}_q$-representable matroids. This will involve also to develop algorithms for backtracking the computation tree of the matroid decomposition.

Backtracking, randomized & isomorphism testing algorithms

As mentioned above, backtracking algorithms are needed for keeping track of some of the principal algorithms in coding theory. Some randomized and iterated information set decoding have been proposed that can be extended to our main problem. Moreover, the isomorphism testing algorithm poposed by Feulner involves a huge amount of massive computing since the search tree to huge to get results for larger parameters of the code, thus any simplification will be welcome.

Canonical database construction

One of the aims is to devise an algorithm that give us a canonical representative for each equivalence class of code. Such a representative is computed by Feulner's algorithm but also it will be nice to known, for a given code, the reduction path to the canonical representative (i.e. the semilinear isometry transform used) as well as some well known parameters and invariants of the code.

Acknowledgements

Acknowledgements

We would like to thank all the institutions that supported us during the *Soria Summer School on Computational Mathematics 2010: Algebraic Geometric Modeling in Information Theory*, namely Junta de Castilla y León, Proyecto i-Math and Universidad de Valladolid.

Thanks for your attention!!