A polynomial time reduction from the multi-graph isomorphism problem to additive code equivalence

Simeon Ball[*] and James Dixon

Abstract

We present a polynomial time reduction from the multi-graph isomorphism problem to the problem of code equivalence of additive codes over finite extensions of $\mathbb{F}_2$.

1 Introduction

Let F be a finite field of characteristic p, where p is a prime. An *additive code* of length n over F is a subset of F^n with the property that for all $u, v \in C$, we have that $u + v \in C$. It is easy to prove that an additive code over F is linear over $\mathbb{F}_p$, the finite field with p elements. Thus, an additive code over F can be defined as the row space over $\mathbb{F}_p$ of a matrix whose elements are from F. The code equivalence problem for additive codes is the following. Given two $k \times n$ matrices over F, when is there a permutation of the columns of one of the matrices, together with a permutation σ_i, for $i \in \{1, \dots, n\}$, which is a permutation of the elements of F in the i-th coordinate, so that the row-spaces over $\mathbb{F}_p$ are the same. If $F = \mathbb{F}_p$ then an additive code is linear and if, furthermore, $p = 2$ then the code is a binary linear code. If the permitations σ_i are all additive then the codes are *additively equivalent*. It is not known if equivalent additive codes are necessarily additively equivalent, see [1].

A multi-graph is a graph whose edges are assigned a weight from $\{1, \dots, h\}$, for some natural number h. An edge joining vertices u and v of weight w is interpreted as w edges joining u and v. The multi-graph isomorphism problem is the following. Given two multi-graphs determine if there a bijection between the set of vertices which induces a bijection on the edges. i.e. edges of weight w are mapped to edges of weight w. If $h = 1$ then a multi-graph is simply a graph.

In [7], Petrank and Roth provide a polynomial time reduction from the graph isomorphism problem to the binary linear code equivalence problem. The graph isomorphism problem is in

[*] 30 August 2021. The first author acknowledges the support of the Spanish Ministry of Science and Innovation grant MTM2017-82166-P.

NP but is not known to be NP-complete. The problem is not known to be solvable in polynomial time either and is therefore a good candidate to belong to the computational complexity class NP-intermediate. Known algorithms for graph isomorphism include McKay's Nauty algorithm [5], Ullmann's algorithm [10], the VF2 algorithm [3] and the parameterised matching algorithm [6]. All these algorithms have exponential worst case performance. Solving isomorphism generally takes much longer time if there is no match, since all possible mappings are eventually searched until it is shown that there is no isomorphism. The latter extends to multi-graph isomorphism and is based on a parameterised sequence which is a walk that covers every vertex in the graph.

In this note we extend the polynomial time reduction of Petrank and Roth [7] to a polynomial time reduction of multi-graph isomorphism to additive code equivalence, where the code is over an extension of $\mathbb{F}_2$. Since a graph on n vertices has at most $\frac{1}{2}n(n-1)$ edges, we assume that the multi-graph has $h \leqslant \frac{1}{2}n(n-1)$ weights and that these are from the set $\{1,\ldots,h\}$.

2 The reduction of multi-graph isomorphism to additive code equivalence

Let Γ be a multi-graph with vertex set V, whose edges have weights belonging to the set $\{1,\ldots,h\}$. Let A be the incidence matrix of Γ, whose rows are indexed by the edges, whose columns are indexed by the vertices and where the edge-vertex entry is equal to the weight of the edge if the edge is incident with the vertex and zero otherwise. Let E denote the set of edges of Γ and let E_i be the subset of E of edges of weight at least i (so $E_1 = E$).

Let e be a primitive element of $\mathbb{F}_{2^r}$, where $r > \log_2 h$. Let D_i be the $|E_i| \times |E_i|$ diagonal matrix indexed by the edges of E_i whose diagonal entry is e^{j-1}, where j is the weight of the edge indexing the row of D_i.

Let

$$N = (h+2)|E_1| + |E_2| + \cdots + |E_h| + |V|.$$

We map Γ to the additive code which is the $\mathbb{F}_2$-row span of the $|E| \times N$ matrix

$$\mathrm{G}(\Gamma) = \left(\underbrace{D_1 \mid D_1 \mid \cdots \mid D_1}_{h+2} \mid \begin{array}{c} O_2 \\ D_2 \end{array} \mid \begin{array}{c} O_3 \\ D_3 \end{array} \mid \cdots \mid \begin{array}{c} O_h \\ D_h \end{array} \mid A \right)$$

where O_i is a matrix of zeros, whose dimensions are determined by the fact that the matrix $\mathrm{G}(\Gamma)$ is a $|E| \times N$ matrix and the matrix D_i is a $|E_i| \times |E_i|$ matrix.

The following two observations will be vital.

(O1) The row of $G(\Gamma)$ indexed by an edge of weight i is a codeword of weight $h+3+i$ whose coordinates are either 0 or e^{i-1}.

(O2) The only codewords of weight at most $2h+3$ are rows of $G(\Gamma)$.

The following is the main theorem of this note.

Theorem 1. *The multi-graphs* Γ *and* Γ' *are isomorphic if and only if the additive codes generated by* $\mathrm{G}(\Gamma)$ *and* $\mathrm{G}(\Gamma')$ *are equivalent.*

Proof. Suppose the multi-graphs Γ and Γ' are isomorphic. Then there is a permutation of the columns of A and a permutation of the rows of A, which gives A', the incidence matrix of the multi-graph Γ'. Apply the column permutation to the last $|V|$ columns of Γ and the permutation of the rows of A to the rows of $\mathrm{G}(\Gamma)$. Since the permutation of the rows of A, takes edges of weight i to edges of weight i, it takes codewords of weight $h+3+i$, which are rows of $G(\Gamma)$, to codewords of weight $h+3+i$, which are rows of $G(\Gamma')$ by (O1) and (O2). Thus, $G(\Gamma)$ and $G(\Gamma')$ generate equivalent codes.

Now suppose the additive codes generated by $\mathrm{G}(\Gamma)$ and $\mathrm{G}(\Gamma')$ are equivalent. Then, by definition, there is an $N\times N$ permutation matrix P, a $|E|\times|E|$ change of basis matrix S and permutations σ_i, $i\in\{1,\dots,N\}$, such that we can apply P to the columns of $\mathrm{G}(\Gamma)$, S to the rows of $\mathrm{G}(\Gamma)$ and the permutation σ_i to the i-th coordinate and obtain a matrix whose generates, over $\mathbb{F}_2$, the same code as $\mathrm{G}(\Gamma')$.

Properties (O1) and (O2) imply that the change of basis matrix S is in fact a permutation matrix, since rows of $\mathrm{G}(\Gamma)$ must be mapped to rows of $\mathrm{G}(\Gamma')$. Furthermore, S maps codewords of weight $h+3+i$ to codewords of weight $h+3+i$, so the initial $N-|V|$ columns of SG contain each vector of weight one, with a non-zero coordinate indexed by an edge of weight i, precisely $h+1+i$ times. Hence, the first $N-|V|$ columns of G' can be obtained by permuting the first $N-|V|$ columns of SG. This permutation of the columns then maps the columns of A to columns of A'.

The permutation σ_i must fix all elements of F appearing in the i-th column of a column of A, since by property (O1) the rows of $\mathrm{G}(\Gamma')$ of weight $h+3+i$ have coordinates 0 or e^{i-1}.

Hence, we obtain $\mathrm{G}(\Gamma')$ from $\mathrm{G}(\Gamma)$ by applying a permutation of the columns of A and a permutation of the rows of A. Thus, Γ and Γ' are isomorphic. □

The polynomial time reduction follows from Theorem 1, by noting that $|N|<chn^2<n^4$, for some constant c. Note also that we can replace equivalence with additive equivalence, since the permutations σ_i are always trivial.

It is interesting to ask if N could be decreased, possibly by increasing the size of the field extension. For $h=2$ it is possible to reduce N to $3|E_1|+|E_2|+|V|$ by simply deleting one of the initial D_1 matrices. One can check that the proof still works, although a little more subtlety is required. Observe that the sum of two codewords of weight 5 (corresponding to edges of weight one) cannot produce a codeword of weight 6 since this would require a repeated single edge, which is an edge of weight two.

3 Code equivalence algorithms

We discussed known algorithms for solving the graph isomorphism problem in the introduction. Algorithms for solving binary linear code equivalence include Bouyukliev's algorithm [2], which is similar to McKay's graph isomorphism algorithm [5], Leon's algorithm [4] and the support splitting algorithm of Sendrier [8], [9]. The latter does not apply to all binary linear codes but it is interesting because it is fast for binary linear codes in which the dimension of the hull $H(C) = C \cap C^{\perp}$ is small. It seems that this is the case for codes obtained from the polynomial time reduction of the graph isomorphism problem due to Petrank and Roth [7]. This warrants further investigation.

The support splitting algorithm allows one to determine signatures from the weight distribution of the hull of truncated binary linear codes, truncating one coordinate at a time. This works because one of the hulls of either the truncated code or the truncations of its dual is non-trivial. This can be seen as follows. Let C_i be the code obtained from C be setting the i-th coordinate to zero in all codewords. Suppose that $H(C) = \{0\}$. Thus, $\mathbb{F}_2^n = C \oplus C^{\perp}$.

Without loss of generality, we consider a truncation on the first coordinate. We have that

$$(1, 0, \ldots, 0) = u + v,$$

for some $u \in C$ and $v \in C^{\perp}$.

If $(u_1, v_1) = (1, 0)$ then, since $v_1 = 0$, we have that $v \in (C_1)^{\perp}$ and since $v = (1, 0, \ldots, 0) + u$, we have that $v \in C_1$. Hence, $v \in H(C_1)$.

Similarly, if $(u_1, v_1) = (0, 1)$ then, since $u_1 = 0$, we have that $u \in ((C^{\perp})_1)^{\perp}$ and since $u = (1, 0, \ldots, 0) + v$, we have that $u \in (C^{\perp})_1$. Hence, $u \in H((C^{\perp})_1)$.

This truncation trick does not carry over to additive codes. It is possible that both $H((C^{\perp})_1)$ and $H(C_1)$ are trivial. However, if we are interested in establishing additive equivalence then we can employ a slightly modified support splitting algorithm; instead of removing the i-th coordinate we take a subspace. In effect, truncating in the binary case would be equivalent to taking the subspace $\{0\}$. This is done in the following way. The i-th column of a generator matrix for C is a vector $v_i \in \mathbb{F}_{2^r}^k$, where $|C| = 2^k$. Writing v_i out over the basis $\{1, \alpha, \ldots, \alpha^{r-1}\}$ of $\mathbb{F}_{2^r}$ over $\mathbb{F}_2$,

$$v_i = \sum_{j=0}^{r-1} v_i^{(j)} \alpha^j,$$

where $v_i^{(j)} \in \mathbb{F}_2^k$. Let ρ_i be the subspace of $\mathbb{F}_2^k$, of dimension at most r, spanned by $v_i^{(0)}, \ldots, v_i^{(r-1)}$. The subspace ρ_i is unaffected by an additive permutation on the i-th coordinate, which only has the effect of changing the basis. Thus, we can select a subspace π_i of ρ_i and replace the i-th column of the generator matrix with the corresponding vector of $\mathbb{F}_{2^r}^k$ obtained from the subspace π_i by reversing the construction of ρ_i above. This code we denote by C_{i,π_i}, where i is the selected coordinate and π_i is the subspace. For equivalent codes, we have that selecting a coordinate and a subspace in C, there must be a corresponding coordinate and subspace in C' which is equivalent, i.e. C_{i,π_i} is equivalent to $C'_{i',\pi'_{i'}}$, for some i' and $\pi'_{i'}$. The weight distribution of the hull of C_{i,π_i} can then be used as a signature with an aim of establishing the permutation taking C to C', as in the support splitting algorithm.

References

[1] S. Ball and J. Dixon, The equivalence of linear codes implies semi-linear equivalence, preprint (2021). arXiv:2107.07912.

[2] I. Bouyukliev, About the code equivalence. in: *Advances in coding theory and cryptography*, *Ser. Coding Theory Cryptol.*, **3** (2007) 126–151.

[3] L. P. Cordella, P. Foggia, C. Sansone, M. Vento, A (sub) graph isomorphism algorithm for matching large graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26** (2004) 1367–1372.

[4] J. Leon, Computing automorphism groups of error-correcting codes, *IEEE Trans. Inform. Theory*, **28** (1982) 496-511.

[5] B. D. McKay, Practical graph isomorphism, *Congressus Numerantium*, **30** (1981) 45-87.

[6] J. Mendivelso, S. Kim, S. Elnikety, Y. He, S. Hwang, Y. Pinzón Y, Solving Graph Isomorphism Using Parameterized Matching. In: Kurland O., Lewenstein M., Porat E. (eds) *String Processing and Information Retrieval. SPIRE 2013*. Lecture Notes in Computer Science, vol **8214**. Springer, Cham, 2013. https://doi.org/10.1007/978-3-319-02432-5_26

[7] E. Petrank and R. M. Roth, Is code equivalence easy to decide?, *IEEE Trans. Inform. Theory*, **43** (1997) 1602–1604.

[8] N. Sendrier, The support splitting algorithm, Research report no 3637, 33 pages 1999.

[9] N. Sendrier, Finding the permutation between equivalent linear codes: the support splitting algorithm, *IEEE Trans. Inform. Theory*, **46** (2000), 1193–1203.

[10] J. R. Ullmann, An algorithm for subgraph isomorphism. *Journal of the ACM*, **23** (1976) 31–42.

[11] Lamand SR Bateman JF.Collagen VI disorders: Insights on form and function in the extracellular matrix and beyond. Matrix Biol. 2018 Oct;71-72:348-367. doi: 10.1016/j.matbio.2017.12.008. Epub 2017 Dec 22.

Simeon Ball
Departament de Matemàtiques,
Universitat Politècnica de Catalunya,
Carrer Jordi Girona 1-3,
08034 Barcelona, Spain
simeon.michael.ball@upc.edu

James Dixon
Facultat de Matemàtiques,
Universitat Politècnica de Catalunya,
Carrer de Pau Gargallo, 14,
08028 Barcelona, Spain
james.dixon@estudiantat.upc.edu