Iranian Journal of Mathematical Sciences and Informatics
Vol. 14, No. 1 (2019), pp 81-93
DOI: 10.7508/ijmsi.2019.01.008

Computation of Minimum Hamming Weight for Linear Codes

Esmaeil Rostami*, Reza Nekooei

Department of Pure Mathematics, Faculty of Mathematics and Computer,
Shahid Bahonar University of Kerman, Kerman, Iran.

E-mail: e_rostami@uk.ac.ir
E-mail: rnekooei@uk.ac.ir

Abstract. In this paper, we consider the minimum Hamming weight for linear codes over special finite quasi-Frobenius rings. Furthermore, we obtain minimal free R-submodules of a finite quasi-Frobenius ring R which contain a linear code and derive the relation between their minimum Hamming weights. Finally, we suggest an algorithm that computes this weight using the Gröbner basis and we show that under certain conditions a linear code takes the maximum of minimum Hamming weight.

Keywords: Algebraic coding theory, Linear code, quasi-Frobenius ring, Gröbner basis, SPAP-ring.

2000 Mathematics subject classification: 11T71, 94B05, 94B65, 16L60, 13P10.

1. Introduction

Linear codes have been extensively studied since, because of their algebraic structure, they are easier to describe, encode and decode than nonlinear codes. Let F_q be a field with q elements. Suppose that F_q^n denotes the vector space of all n-tuples over F_q. A code C of length n over F_q is a subset of F_q^n. We usually denote the vectors in F_q^n by $(a_1, a_2, ..., a_n)$ and call the vectors in C " codewords". If we do not impose further structure on a code, its usefulness is

*Corresponding Author

Received 13 August 2016; Accepted 30 December 2017
©2019 Academic Center for Education, Culture and Research TMU

[Downloaded from ijmsi.com on 2026-11-16]

limited. The most useful additional structure is that of linearity. The two most common representations of a linear code are a generator matrix and a parity check matrixs(see [15]). The foundations of classical algebraic coding theory over finite fields involve notions and results like dual code, MacWilliams identity and extension theorem. Kuzmin, Kurakin, Nechaev, Norton, and Sălăgean ([13, 17, 18, 20, 22]) developed a theory of linear codes over finite commutative rings and showed that the basic results may be suitably generalized to codes over quasi-Frobenius rings. Actually, coding theorists can extend many results in coding theory over fields to the quasi Frobenius rings. In the other words, quasi Frobenius rings lie on the intersection of coding theory and ring theory. Codes over finite rings, especially over $\mathbb{Z}_4$, have been of great interest, starting with Nechaev [19] and Hammons, et. al. [6]. Hammons, et. al showed that Kerdock and Preparata codes are linear over $\mathbb{Z}_4$ via the Gray map from $\mathbb{Z}_4^n$ to $\mathbb{Z}_2^{2n}$ and that viewed as $\mathbb{Z}_4$-codes they are duals [6]. However, the study of codes over finite rings other than finite fields goes back to the early 1970's. Many researchers have considered codes over finite chain rings (e.g., [7, 22]) and codes over finite Frobenius rings (e.g., [5, 26]) with respect to homogeneous weights [3]. In particular, Wood [26] has proved the two MacWilliams theorems, the extension theorem and the MacWilliams identities for codes over finite Frobenius rings. Horimoto, Storme, and Shiromoto [10, 11, 24] have introduced the singleton bound for codes over finite quasi-Frobenius (QF) rings. Codes over rings have been shown to have interesting connections to lattices, modular forms [16], and Hjelmslev geometries [8, 9] as well as many other branches of mathematics.

In this paper, we consider linear codes over an SPAP-ring (special kind of finite Frobenius rings). The best examples of SPAP-rings, are fields and $\mathbb{Z}_{p^2}$, where p is a prime number, which play an important role in coding theory. In particular $\mathbb{Z}_4$, is known in coding theory as a Gray code. So the SPAP-rings are more general than the rings considered in coding theory. Also, we characterize the structure of SPAP-rings in [21]. One of the advantages of this structure is that we can give an algorithm for computing almost all computable items, such as dimension, rank and basis, and ..., by using the theory of Gröbner basis. Hence SPAP-rings are applicable to the class of rings in coding theory. Then we obtain the relationship between the minimum Hamming weight and the minimum free module containing this code. Also, we propose an algorithm for computing the minimum Hamming weight for a linear code and state a condition for a code to be optimal. We refer to [15] for any undefined terms from coding theory.

2. Minimum Hamming Weight and Griesmer Bound

Let R be a finite QF-ring, that is, a ring which can be viewed as an injective module over itself (see [14]). Let R^n be the free R-module of rank n consisting

[Downloaded from ijmsi.com on 2026-11-16]

of all n-tuples of elements of R. With respect to component-wise addition and ring multiplication, R^n has the structure of an R-module. A linear code C of length n over R is an R-submodule of R^n. If C is a free R-submodule of R^n, we call C a free code. For a vector $x \in R^n$, the (Hamming) weight $wt(x)$ of x is defined to be the number of non-zero components in x and the minimum Hamming weight of a linear code C of length n over R, denoted be $d(C)$, is $min\{wt(x)|x \in C, x \neq 0\}$.

If an error event affects entire symbols, then the minimum Hamming weight comes naturally. In this case a code with minimum Hamming weight d can cope with all patterns of t erroneous symbols and e erased symbols, if $d \geq 2t+e$. So, most of the well-known algebraic decoding algorithms use minimum Hamming weight. For example Berlekamp-Massey algorithm was devised for $\mathbb{Z}_m$ using minimum Hamming weight for decoding. Also, for codes over rings the minimum Hamming weight is a lower bound for the Lee distance. So by computing minimum Hamming weight, we can find a lower bound for Lee distance. Furthermore, for some rings, for example $\mathbb{Z}_4$, the minimum Hamming weight and Lee distance coincide. A local ring (R, m) is called a special product of almost prime ideals ring (abbreviated, SPAP-ring) if for each $x \in m - m^2, (x^2) = m^2$ and $m^3 = 0$. This class of rings is a subclass of local rings introduced in 2008 by D. D. Anderson and M. Bataineh in [2]. The aim of this paper is to study the linear codes over SPAP-rings. For more information about SPAP-rings see [21]. Also, SPAP-rings have a graphical characterization which helps for understanding these rings, see [23, 25] for more informations. In [21], we show that finite SPAP-rings are quasi Frobenius, which are the most important rings considered in coding theory. Essential extension and injective hull are two concepts that help us to establish our results(see [14]). Now let C be a linear code. Hence C is a subspace of R^n for some n .Thus there exists a minimal free R-module in R^n that contains C(see [14]). We have the following proposition.

Proposition 2.1. *Let (R, m) be an SPAP-ring with finitely generated m and let C be a linear code of R^n. If $E(C)$ is an injective hull of C contained in R^n, then $E(C)$ is a minimum free R-module containing C and the minimum Hamming weight of C is equal to the minimum Hamming weight of $E(C)$. Furthermore, each minimum free R-module of R^n containing C is R-isomorphic with $E(C)$.*

Proof. Let C be a linear code of R^n. Hence C is a submodule of R^n. Since R^n is a free R-module, by [21, Lemma 2.3] , it is an injective R-module containing C. Therefore, by [14, Lemma 3.29], $E(C)$ is in R^n and by [21, Lemma 2.3], $E(C)$ is free. However by [21, Lemma 2.3], we know that over SPAP-rings a free R-module is equivalent to an injective R-module and $E(C)$ is a minimum injective R-module of R^n containing C. It follows that $E(C)$ is a minimum free R-module of R^n containing C. Now if L is a minimum free R-module of R^n

[Downloaded from ijmsi.com on 2026-11-16]

containing C, by [21, Lemma 2.3], it is a minimum injective R-module of R^n containing C. So it is an injective hull of C and hence by [14, Corollary 3.32], it is R-isomorphic with $E(C)$. Now we prove that the minimum Hamming weight of C is equal to the minimum Hamming weight of $E(C)$. We note that $C \subseteq E(C)$. So $d(E(C)) = min\{wt(x)|x \in E(C), x \neq 0\} \leq min\{wt(x)|x \in C, x \neq 0\} = d(C)$. Now let $d(E(C)) = n$, hence there exists $0 \neq y \in E(C)$ such that $d(E(C)) = wt(y)$. Since $E(C)$ is the injective hull of C, it is essential over C. Hence by [14, Remark 3.27], there exists $0 \neq r \in R$ such that $0 \neq ry \in C$. Now if we consider y and ry as elements of R^n, we see that $wt(ry) \leq wt(y)$. So $d(C) = min\{wt(x)|x \in C, x \neq 0\} \leq wt(ry) \leq wt(y) = n = d(E(C))$. Therefore, $d(E(C)) = d(C)$. □

In the theory of linear codes, minimum Hamming weight has an economic and error-correcting interpretation and we want to obtain a linear code such that its minimum Hamming weight is maximum. By Proposition 2.1, it is sufficient to maximize the minimum Hamming weight of the injective hull. For a linear code C of length n over R, we denote the rank of minimal free R-submodule of R^n which contain C by $k(C)$. So $k(C) = rank(E(C))$. Now we have the following theorem.

Theorem 2.2. [24] *Let (R, m) be a finite local Frobenius ring with $|\frac{R}{m}| = q$, where q is a prime power. For a linear code C of length n over R and rank $k(C)$, we have*

$$n \geq \sum_{i=0}^{k(C)-1} \lceil \frac{d(C)}{q^i} \rceil,$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

For a linear code C over a finite local Frobenius ring R with minimum Hamming distance d, the bound in the inequality in above theorem is known as the Griesmer bound for C. Consider a linear code C with $\text{rank}(E(C)) = k$. For computing minimum Hamming weight of C, we must first be able to compute the minimum Hamming weight of free $E(C)$. In the next section we propose an algorithm for computing the minimum Hamming weight of a free R-modules using the Gröbner basis and give a characterization of a free module which has the maximum minimum Hamming weight over SPAP-rings.

3. An Algorithm for Computing Minimum Hamming Weight

In this section, we use the theory of Gröbner basis and the structure of SPAP-rings to compute the minimum Hamming weight of a free linear code. For this we suggest an algorithm that computes the minimum Hamming weight of a linear code. Moreover, this algorithm can distinguish, is a given subset of a code, linearly independent or not. In the other word this algorithm distinguishes, when a linear code is free and by omitting the additional elements in a generating set it can find a maximal linearly independent subset of a given

[Downloaded from ijmsi.com on 2026-11-16]

set. We also derive a condition for a linear code to have maximum minimum Hamming weight.

Definition 3.1. Let R be a ring and let $(f_1, ..., f_t)$ be an ordered t-tuple, where $f_i \in R$. The set of all $(a_1, ..., a_t) \in R^t$ such that $a_1f_1 + ... + a_tf_t = 0$ is an R-submodule of R^t, called the Syzygy module of $(f_1, ..., f_t)$ and denoted by $Syz(f_1, ..., f_t)$.

Now we state some results concerning Gröbner basis and COCOA programming that help us to compute the minimum Hamming weight of a free linear code over SPAP-rings. Note that in this section the ring $S = k[x_1, x_2, ..., x_n]$ is the ring of all polynomials in n variables with coefficients in the field k. Let $(f_1, ..., f_t)$ be an ordered t-tuple of elements with $f_i \in R$. Then $Syz(f_1, ..., f_t)$ is computable by an algorithm using COCOA programing and the command "$Syz(M : IDEALorMODULE, Index : INT) : MODULE$". Also if N and M be two submodules of R^n, then we can compute the intersection of N and M, by command "$Intersection(E_1 : MODULE, ..., E_n : MODULE) : MODULE : MODULE$" (see [1, 4, 12]). In [21], we characterize the structure of SPAP-rings, in this paper we focus in all rings of the form $\frac{k[x_1,x_2,...,x_n]}{(x_ix_j, x_i^2-x_j^2, x_i^3)_{i\neq j}}$ that are special kind of SPAP-rings, where k is a field. By an easy computation we see that two ideals $(x_ix_j, x_i^2 - x_j^2, x_i^3)_{i\neq j}$ and $(x_ix_j, x_1^2 - x_j^2, x_n^3)_{i\neq j}$ are equal. So we shall consider rings of the form $\frac{k[x_1,x_2,...,x_n]}{(x_ix_j, x_1^2-x_j^2, x_n^3)_{i\neq j}}$.

Let R be the SPAP-ring $\frac{k[x_1,x_2,...,x_n]}{(x_ix_j, x_1^2-x_j^2, x_n^3)_{i\neq j}}$, where k is a field. Suppose that C is a free linear code of R^n of rank k with basis $\Omega = \{(f_{i1} + I, f_{i2} + I, ..., f_{in} + I)\}$ for $i = 1, ..., k$ and $j = 1, 2, ..., n$, where $f_{ij} \in k[x_1, x_2, ..., x_n]$ and $I = (x_ix_j, x_1^2 - x_j^2, x_n^3)_{i\neq j}$. Let $M_j = Syz([f_{1j}, f_{2j}, ..., f_{kj}, x_1x_2, x_1x_3, ..., x_1x_n, x_2x_1, x_2x_3, ..., x_2x_n, ..., x_nx_1, x_nx_2, ...x_nx_{n-1}, x_1^2 - x_2^2, x_1^2 - x_3^2, ..., x_1^2 - x_n^2, x_n^3])$, for $j = 1, ..., n$.

We have the following lemma.

Lemma 3.2. *Let $[f_{1j} + I, f_{2j} + I, ..., f_{kj} + I]$ be an ordered k-tuple in R^k, then $[g_1 + I, g_2 + I, ..., g_k + I]$ is an element of $Syz([f_{1j} + I, f_{2j} + I, ..., f_{kj} + I])$ in R if and only if the elements of ordered k-tuple $[g_1, g_2, ..., g_k]$ are respectively the first k components of an element in M_j.*

Proof. Let $[g_1+I, g_2+I, ..., g_k+I]$ be an element of $Syz([f_{1j}+I, f_{2j}+I, ..., f_{kj}+I])$. Hence

$$\sum_{i=1}^{k}(f_{ij} + I)(g_i + I) = 0.$$

This means that $\sum_{i=1}^{k} f_{ij}g_i \in I$. So $\sum_{i=1}^{k} f_{ij}g_i$ is a combination of the generators $\{x_ix_j, x_1^2-x_j^2, x_n^3\}$ of I using the ordering in M_j. Therefore the elements of ordered the k-tuple $[g_1, g_2, ..., g_k]$ are respectively the first k components of an element in M_j. Conversely, let $[g_1, g_2, ..., g_k]$ be respectively the first k elements of an element in M_j. Hence by the definition of M_j, we have

[Downloaded from ijmsi.com on 2026-11-16]

$$\sum_{i=1}^{k}(f_{ij})(g_i)+\sum(h_\alpha)(g_\alpha)=0$$

where g_α is the ordering of elements according to M_j. So we have

$$\sum_{i=1}^{k}(f_{ij}+I)(g_i+I)+\sum(h_\alpha+I)(g_\alpha+I)=0+I.$$

Therefore

$$\sum_{i=1}^{k}(f_{ij}+I)(g_i+I)=0.$$

Hence $[g_1+I,g_2+I,...,g_k+I]$ is an element of $Syz([f_{1j}+I,f_{2j}+I,...,f_{kj}+I])$. □

Now we consider the following matrix equation:

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1n}+I & f_{2n}+I & \dots & f_{kn}+I \end{pmatrix}_{n\times k} \begin{pmatrix} g_1+I \\ \vdots \\ g_k+I \end{pmatrix}_{k\times 1} = 0.$$

By Lemma 3.2, $(g_1+I,g_2+I,...,g_k+I)$ is the solution of this matrix equation if and only if for all $j=1,...,n$, the elements of ordered k-tuple $(g_1,g_2,...,g_k)$ are respectively the first k components of an element in M_j. Note that an element $h=(l_1+I,l_2+I,...,l_n+I)\in R^n$ is an element of C if it is a linear combination of its basis. This means there exists an element $(g_1+I,g_2+I,...,g_k+I)\in R^k$ such that

$$h=(g_1+I,g_2+I,...,g_k+I)\begin{pmatrix} f_{11}+I & f_{12}+I & \dots & f_{1n}+I \\ f_{21}+I & f_{22}+I & \dots & f_{2n}+I \\ \vdots & & & \vdots \\ f_{k1}+I & f_{k1}+I & \dots & f_{kn}+I \end{pmatrix}\in C$$

where $g_1+I,g_2+I,...,g_k+I$ are the coefficients of this combination. Hence

$$l_j+I=\begin{pmatrix} g_1+I, & g_2+I, & \dots & ,g_k+I \end{pmatrix}_{1\times k}\begin{pmatrix} f_{1j}+I \\ \vdots \\ f_{kj}+I \end{pmatrix}_{k\times 1}$$

$$=\begin{pmatrix} f_{1j}+I, & f_{2j}+I, & \dots & ,f_{kj}+I \end{pmatrix}_{1\times k}\begin{pmatrix} g_1+I \\ \vdots \\ g_k+I \end{pmatrix}_{k\times 1}.$$

So $l_j+I=0$ if and only if $(g_1+I,g_2+I,...,g_k+I)$ belongs to $Syz([f_{1j}+I,f_{2j}+I,...,f_{kj}+I])$ or by Lemma 3.2, equivalently the elements of the ordered k-tuple $[g_1,g_2,...,g_k]$ are respectively the first k components of an element in M_j. Now, by definition of minimum Hamming weight, it is enough to find the nonzero element of C with maximum zero component. The minimum Hamming

[Downloaded from ijmsi.com on 2026-11-16]

weight of C will be equal to $n-n'$ where n' = "the number of zero component". To obtain the minimum Hamming weight of C we first compute the M_j and then the following sets:

$$T_{\emptyset} = \bigcap_j M_j, T_{i_1} = \bigcap_{j\neq i_1} M_j \ ,T_{i_1 i_2} = \bigcap_{j\notin\{i_1,i_2\}} M_j,..., T_{i_1,...,i_l} = \bigcap_{j\notin\{i_1,...,i_l\}} M_j,$$

where $i_r \in \{1,2,...,n\}$ and $l \in \{1,2,...,n-1\}$.

Now if for all elements of the above sets we keep the first k components and erase the rest and take the remaining set modulo I, we obtain new sets, say,

$$\widehat{T_{\emptyset}}, \widehat{T_{i_1}}, \widehat{T_{i_1 i_2}}, ..., \widehat{T_{i_1,...,i_l}}$$

where $i_r \in \{1,2,...,n\}$ and $l \in \{1,2,...,n-1\}$.

Theorem 3.3. (*Algorithm*) *In the above notation for a free linear code C, if for some $m-1 \in \{1,2,...,n-1\}$ and $i_r \in \{1,2,...,n\}$ all $\widehat{T_{i_1,...,i_{m-1}}} = 0$ and there exists $i_1, i_2, ..., i_m \in \{1,2,...,n\}$ such that $\widehat{T_{i_1,...,i_m}} \neq 0$, then the minimum Hamming weight of C is m.*

Proof. Suppose that the hypothesis is true for $m = 1$. Then at last one $\widehat{T_{i_1}}$ is nonzero, suppose $\widehat{T_n}$ is nonzero. So by definition of $\widehat{T_n}$, there exists $(g_1+I, g_2+I, ..., g_k+I) \neq (0,0,...,0) \in R^k$ such that the elements of $(g_1+I, g_2+I, ..., g_k+I)$ are respectively the first k components of an element in $T_n = \bigcap_{j\notin\{n\}} M_j$. This means that $(g_1+I, g_2+I, ..., g_k+I)$ is a nontrivial solution of the following matrix equation.

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1(n-1)}+I & f_{2(n-1)}+I & \dots & f_{k(n-1)}+I \end{pmatrix}_{(n-1)\times k} \begin{pmatrix} X_1+I \\ \vdots \\ X_k+I \end{pmatrix}_{k\times 1} = 0$$

But since C is free, it is not a solution of the following equation because the columns of the matrix are linearly independent.

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1n}+I & f_{2n}+I & \dots & f_{kn}+I \end{pmatrix}_{n\times k} \begin{pmatrix} X_1+I \\ \vdots \\ X_k+I \end{pmatrix}_{k\times 1} = 0.$$

This means that C contains the element $X = (0,0,...,0,\sum_{i=1}^{k}(f_{in}+I)(g_i+I))$, where

$$\sum_{i=1}^{k}(f_{in}+I)(g_i+I) \neq 0.$$

.

So $wt(X) = 1$ and therefore the minimum Hamming weight of C is 1.

Now suppose all $\widehat{T_{i_1}}$ are zero modules and at last one $\widehat{T_{i_1 i_2}}$ is nonzero. Let $\widehat{T_{n-1,n}}$ be nonzero. By definition of $\widehat{T_{n-1,n}}$, there exists $(g_1+I, g_2+I, ..., g_k+$

[Downloaded from ijmsi.com on 2026-11-16]

$I) \neq (0,0,...,0) \in R^k$ such that the elements of $(g_1+I, g_2+I, ..., g_k+I)$ are respectively the first k components of an element in $T_{n-1,n} = \bigcap_{j \notin \{n-1,n\}} M_j$. This means that $(g_1+I, g_2+I, ..., g_k+I)$ is a nontrivial solution of the following matrix equation:

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1(n-2)}+I & f_{2(n-2)}+I & \dots & f_{k(n-2)}+I \end{pmatrix}_{(n-2)\times k} \begin{pmatrix} X_1+I \\ \vdots \\ X_k+I \end{pmatrix}_{k\times 1} = 0.$$

However since $\widehat{T_n} = 0, \widehat{T_{n-1}} = 0$, it is not the the solution of following tow matrix equations :

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1(n-1)}+I & f_{2(n-1)}+I & \dots & f_{k(n-1)}+I \end{pmatrix}_{(n-1)\times k} \begin{pmatrix} X_1+I \\ \vdots \\ X_k+I \end{pmatrix}_{k\times 1} = 0.$$

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1(n-2)}+I & f_{2(n-2)}+I & \dots & f_{k(n-2)}+I \\ f_{1(n)}+I & f_{2(n)}+I & \dots & f_{k(n)}+I \end{pmatrix}_{n-1\times k} \begin{pmatrix} X_1+I \\ \vdots \\ X_k+I \end{pmatrix}_{k\times 1} = 0.$$

This means that C has an element of the form $X = (0,0,...,0, \sum_{i=1}^{k}(f_{i(n-1)}+I)(g_i+I), \sum_{i=1}^{k}(f_{in}+I)(g_i+I))$, where $\sum_{i=1}^{k}(f_{in}+I)(g_i+I) \neq 0 \neq \sum_{i=1}^{k}(f_{i(n-1)}+I)(g_i+I)$. So $wt(X) = 2$ and therefore the minimum Hamming weight of C is less than or equal 2. Suppose C has an element of weight one. Then all components of this element except one member are zero. In particular, suppose it is of the form $(0,0,...,0,f+I)$, where $f+I \neq 0$. Since $(0,0,...,0,f+I)$ is an element in C, it is the linear combination of its basis. This means that there exists $(g_1+I, g_2+I, ..., g_k+I) \in R^k$ such that

$$(0,0,...,0,f+I) = (g_1+I, g_2+I, ..., g_k+I) \begin{pmatrix} f_{11}+I & f_{12}+I & \dots & f_{1n}+I \\ f_{21}+I & f_{22}+I & \dots & f_{2n}+I \\ \vdots & & & \vdots \\ f_{k1}+I & f_{k1}+I & \dots & f_{kn}+I \end{pmatrix}.$$

We can write the above as follows:

[Downloaded from ijmsi.com on 2026-11-16]

$$\begin{pmatrix} f_{11}+I & f_{21}+I & \dots & f_{k1}+I \\ f_{12}+I & f_{22}+I & \dots & f_{k2}+I \\ \vdots & & & \vdots \\ f_{1(n)}+I & f_{2(n)}+I & \dots & f_{k(n)}+I \end{pmatrix}_{n\times k} \begin{pmatrix} g_1+I \\ \vdots \\ g_k+I \end{pmatrix}_{k\times 1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ f+I \end{pmatrix}_{n\times 1}.$$

This means that $\widehat{T_n}$ is not zero, which is a contradiction. Hence C has no element of weight 1. Since C is nonzero, hence the minimum Hamming weight of C is 2. The rest of the proof follows by induction. □

It is easy to see that the maximum of minimum Hamming weight of a free linear code is least or equal to $n-1$.

Remark 3.4. This algorithm, is able to compute the minimum Hamming weight for a linear code of arbitrary length. It is enough to give the basis for a free linear code. This algorithm then easily computes the minimum Hamming weight of the code generated by this basis. Note that this is important when our code is infinite or is a code over a polynomial ring with many variables (The Example 4.1 that is considered in this paper is only for checking our algorithm and showing how this algorithm works, not for displaying its ability). Moreover, the first step of this algorithm can distinguish if a given subset of a code is linearly independent or not. In other words this algorithm recognizes a free linear code. Also, by omitting the additional elements in a generating set it can find a linearly independent subset of a given set. Furthermore, the algorithm mentioned in this paper connects coding theory and commutative algebra over polynomial rings using Gröbner basis and Syzygy modules. This algorithm works not only for SPAP-rings but also can be modified for any quotient of polynomial rings. One of this rings is $k[x]/(x^2)$, where k is a field. In particular this result are true for $Z_p[x]/(x^2)$, where p is a prime number.

Proposition 3.5. *Let C be a free linear code . Then the maximum of minimum Hamming weight of C is $n-1$ if and only if all $\widehat{T_{i_1}}$,$\widehat{T_{i_1 i_2}}$,...,$\widehat{T_{i_1,\dots,i_{n-2}}}$ are zero and at least one $\widehat{T_{i_1,\dots,i_{n-1}}}$ is nonzero.*

Proof. This is a direct consequence of Theorem 3.3. □

Theorem 3.6. (*Algorithm*) *Suppose that C is a linear code of R^n and $\Omega = \{(f_{i1}+I, f_{i2}+I, ..., f_{in}+I)\}$ for $i = 1, ..., k$ and $j = 1, 2, ..., n$, where $f_{ij} \in k[x_1, x_2, ..., x_n]$, is a subset of C. Then the elements of Ω are linearly independent if and only of $T_\emptyset = 0$.*

Proof. Clearly form definition and verification of Theorem 3.3. □

Theorem 3.7. (*Algorithm*) *Suppose that C is a linear code of R^n and $\Omega = \{(f_{i1}+I, f_{i2}+I, ..., f_{in}+I)\}$ for $i = 1, ..., k$ and $j = 1, 2, ..., n$, where $f_{ij} \in$*

[Downloaded from ijmsi.com on 2026-11-16]

$k[x_1, x_2, ..., x_n]$, is a subset of C. Then we can select the largest linearly independent subset of Ω.

Proof. If $T_\emptyset = 0$, then by above theorem Ω is this set. If $T_\emptyset \neq 0$, we define $\Omega^l = \{(f_{i1} + I, f_{i2} + I, ..., f_{in} + I)\}$ for $i = 1, ..., \hat{l}, ..., k$ and l omitted. Now if for some Ω^l, $T_\emptyset = 0$, then Ω^l is a wanted set. Otherwise, we continue this process until $T_\emptyset = 0$. □

4. Examples

Now we bring some examples. The first example that be considered is only for checking our algorithm and showing that how this algorithm works, not for the ability of it. This algorithm after making into a computer program, is able to compute the minimum Hamming weight for a linear code of arbitrary length by a click. After programming this algorithm, it is enough to give the basis for a free linear code, then this algorithm computes the minimum Hamming weight of the code that generated by this basis without cumbersome manual works on the elements.

Example 4.1. In this example all computation has been carried out using COCOA. We compute the minimum Hamming weight of a free linear code by our algorithm. Consider the field is $\mathbb{Z}_3$ and we work with two variables. So our SPAP-ring is $R = \frac{\mathbb{Z}_3[x,y]}{(xy, x^2-y^2, y^3)}$. Suppose that C is the submodule of R^3 generated by $(f_{11} + I, f_{12} + I, f_{13} + I) = (1 + I, 1 + I, 1 + I), (f_{21} + I, f_{22} + I, f_{23} + I) = (1 + I, 2 + I, 3 + I)$.

First we compute M_i, for $i = 1, 2, 3$.

Step 1: $UseR ::= ZZ/(3)[x, y]$;
$Syz([1, 1, xy, x^2 - y^2, y^3])$;
The output is:
$Module([[1, -1, 0, 0, 0], [0, xy, -1, 0, 0], [-y^2, x^2, 0, -1, 0], [0, y^3, 0, 0, -1]])$.
This means that
$M_1 = Module([[1, -1, 0, 0, 0], [0, xy, -1, 0, 0], [-y^2, x^2, 0, -1, 0], [0, y^3, 0, 0, -1]])$.

Step 2: $UseR ::= ZZ/(3)[x, y]$;
$Syz([1, 2, xy, x^2 - y^2, y^3])$;
The output is:
$Module([[1, 1, 0, 0, 0], [0, -xy, -1, 0, 0], [-y^2, -x^2, 0, -1, 0], [0, -y^3, 0, 0, -1]])$.
This means that
$M_2 = Module([[1, 1, 0, 0, 0], [0, -xy, -1, 0, 0], [-y^2, -x^2, 0, -1, 0], [0, -y^3, 0, 0, -1]])$

Step 3: $UseR ::= ZZ/(3)[x, y]$;
$Syz([1, 3, xy, x^2 - y^2, y^3])$;
The output is:
$Module([[0, 1, 0, 0, 0], [xy, 0, -1, 0, 0], [x^2 - y^2, 0, 0, -1, 0], [y^3, 0, 0, 0, -1]])$.

[Downloaded from ijmsi.com on 2026-11-16]

This means that
$M_3 = Module([[0,1,0,0,0],[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]])$.
Now we have:
$T_\emptyset = M_1 \bigcap M_2 \bigcap M_3$
Step 4: $UseR ::= ZZ/(3)[x,y]$;
$Intersection(Module([[1,-1,0,0,0],[0,xy,-1,0,0],[-y^2,x^2,0,-1,0],[0,y^3,0,0,-1]]), Module([[1,1,0,0,0],[0,-xy,-1,0,0],[-y^2,-x^2,0,-1,0],[0,-y^3,0,0,-1]]), Module([[0,1,0,0,0],[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]]))$
The output is:
$Module([[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]])$.
This means that
$T_\emptyset = Module([[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]])$
So $\widehat{T_\emptyset} = Module([[xy+I,0+I],[x^2-y^2+I,0+I],[y^3I,0+I]]) = 0$
By Theorem 3.6, this shows that C is free.
For $T_1 = M_2 \bigcap M_3$, we have
Step 5: $UseR ::= ZZ/(3)[x,y]$;
$Intersection(Module([[1,-1,0,0,0],[0,xy,-1,0,0],[-y^2,x^2,0,-1,0],[0,y^3,0,0,-1]]), Module([[1,1,0,0,0],[0,-xy,-1,0,0],[-y^2,-x^2,0,-1,0],[0,-y^3,0,0,-1]]))$;
The output is:
$Module([[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1],[0,0,-x,y,1],[0,0,-y^2,0,x]])$.
This means that
$T_1 = Module([[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1],[0,0,-x,y,1],[0,0,-y^2,0,x]])$.
So $\widehat{T_1} = Module([[xy+I,0+I],[x^2-y^2+I,0+I],[y^3+I,0+I],[0+I,0+I],[0+I,0+I]]) = 0$.
For $T_2 = M_1 \bigcap M_3$, we have
Step 6: $UseR ::= ZZ/(3)[x,y]$;
$Intersection(Module([[1,-1,0,0,0],[0,xy,-1,0,0],[-y^2,x^2,0,-1,0],[0,y^3,0,0,-1]]), Module([[0,1,0,0,0],[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]]))$;
The output is:
$Module([[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]])$.
This means that
$T_2 = Module([[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]])$.
So $\widehat{T_2} = Module([[xy+I,0+I],[x^2-y^2+I,0+I],[y^3+I,0+I]]) = 0$.
For $T_3 = M_1 \bigcap M_2$, we have
Step 7: $UseR ::= ZZ/(3)[x,y]$;
$Intersection(Module([[1,1,0,0,0],[0,-xy,-1,0,0],[-y^2,-x^2,0,-1,0],[0,-y^3,0,0,-1]]), Module([[0,1,0,0,0],[xy,0,-1,0,0],[x^2-y^2,0,0,-1,0],[y^3,0,0,0,-1]]))$;

[Downloaded from ijmsi.com on 2026-11-16]

The output is:
$Module([[-xy, 0, 1, 0, 0], [-x^2 + y^2, 0, 0, 1, 0], [-y^3, 0, 0, 0, 1]])$.
This means that
$T_3 = Module([[-xy, 0, 1, 0, 0], [-x^2 + y^2, 0, 0, 1, 0], [-y^3, 0, 0, 0, 1]])$.
So $\widehat{T_3} = Module([[-xy + I, 0 + I], [-x^2 + y^2 + I, 0 + I], [-y^3 + I, 0 + I]) = 0$.

Since $\widehat{T_1} = \widehat{T_2} = \widehat{T_3} = 0$, hence the minimum Hamming weight of C is not 1. Now since $T_{12} = M_3 = Module([[0, 1, 0, 0, 0], [xy, 0, -1, 0, 0], [x^2-y^2, 0, 0, -1, 0], [y^3, 0, 0, 0, -1]]$, hence $\widehat{T_{12}} = Module([[0 + I, 1 + I], [xy + I, 0 + I], [x^2 - y^2 + I, 0 + I], [y^3 + I, 0 + I]] = Module([[0 + I, 1 + I]] \neq 0$.

This shows that $\widehat{T_{12}} \neq 0$ and so the minimum Hamming weight of C is 2. By Proposition 3.5, this code has maximum minimum Hamming weight.

In the following examples we omit the details.

Example 4.2. Consider the field $\mathbb{Q}$ (the field of rational number), and we work with tree variables. So our SPAP-ring is $R = \frac{\mathbb{Q}[x,y,z]}{(xy,xz,yz,x^2-y^2,x^2-z^2,y^2-z^2,z^3)}$. Suppose that C is the submodule of R^5 generated by:

$$\{(x^2 + I, y^2 + I, x + I, y + I, z + I), (y^2 + I, z^2 + I, y + I, z + I, x + I),$$
$$(z^2 + I, x^2 + I, z + I, x + I, y + I), (x + y + I, x + z + I, y + z + I, 1 + I, 1 + I)\}.$$

For this set of generators, we have $T_\emptyset = 0$. So this set is linearly independent.

5. Conclusion

The main purpose of this paper is to consider a part of coding theory, error-correcting, with techniques of computational commutative algebra, specially Syzygy modules. So we have brought an algorithm for computing the minimum Hamming weight of a free linear code over rings that are SPAP-rings. We believe that the techniques that are considered can be extended to many of the quotient of the polynomial rings with several variables. Also, the mentioned techniques can be used for other kinds of codes that were defined by polynomial ring.

References

1. W. Adams, P. Loustaunau, *An Introduction to Gröbner basis*, Graduate Studies in Math. 3, Amer. Math. Soc, Providence, 1994.
2. D. D. Anderson, M. Bataineh, Generalization of Prime Ideals, *Comm. in Algebra*, **36**, (2008), 686–696.
3. I. Constantinescu, W. Heise, T. Honold, Monomial Extensions of Isometries between Codes over Z_m, *Proceedings of the 5th International Workshop on Algebraic and Combinatorial Coding Theory (ACCT '96), Unicorn Shumen*, (1998), 98–104.
4. D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms*, Springer, New York, 1992.
5. M. Greferath, S. Schmidt, Finite-ring Combinatorics and MacWilliams' Equivalence Theorem, *J. of Combin. Theory, Ser. A.*, **92**, (2000), 17–28.

[Downloaded from ijmsi.com on 2026-11-16]

6. J. A. Hammons, P. Kumar, A. Calderbank, N. Sloane,P. Sol´, The Z_4 linearity of Kerdock, Preparata, Goethals and Related Codes, *IEEE Trans. Inform. Theory*, **40**, (1994), 301–319.
7. T. Honold, I. Landjev, Linear Codes over Finite Chain Rings, *Electronic J. of Combinatorics*, (2000), 7–11.
8. T. Honold, I. Landjev, Projective Hjelmslev geometries, *Proceedings of the Second International Workshop on Optimal Codes*, (1998), 97–115.
9. T. Honold, I. Landjev, Arcs in Projective Hjelmslev planes, *Discrete Math. Appl.*, **11**, (2001), 53–70.
10. H. Horimoto, K. Shiromoto, A Singleton Bound for Linear Codes over quasi-Frobenius Rings, *Proceedings of the 13th AAECC Symposium On Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, Hawaii*, (1999), 51–52.
11. H. Horimoto, K. Shiromoto, MDS Codes over Finite quasi-Frobenius Rings, preprint.
12. M. Kreuzer, L. Robbiano, *Computational Commutative Algebra 1 and 2*, Springer-Verlag, 2000.
13. A. S. Kuzmin, V. L. Kurakin, V. T. Markov, A. V. Mikhalev, A. A. Nechaev, *Linear codes and Polylinear Recurrences over Finite Rings and Modules (Survey)*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. Proc. of the 13-th. Int Symp. AAECC-13. LNCS, Springer, 1999.
14. T. Y. Lam, *Lectures on Modules and Rings*, Graduate Texts in Mathematics, Vol. 189, Springer, New York, 1998.
15. F. MacWilliams, N. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam, The Netherlands: North-Holland, 1997.
16. G. Nebe, E. Rains, N. Sloane, *Self-Dual Codes and Invariant Theory*, Springer, Berlin, 2006.
17. A. A. Nechaev, Linear Codes and Polylinear Recurrences over Finite Rings and quasi-Frobenius modules, *Dokl. Akad. Nauk.*(in Russian), **345**(1), (1995), 229–254.
18. A. A. Nechaev, Linear Codes over Modules and over Spaces, MacWilliams Identity, *Proceedings of the 1996 IEEE Int. Symp. Inf. Theory and Appl., Victoria B. C., Canada*, (1996), 35–38.
19. A. A. Nechaev, The Kerdock Code in a Cyclic Form, *Discrete Math. Appl.*, **1**, (1991), 365–384.
20. A. A. Nechaev, A. S. Kuzmin, V. T. Markov, Linear Codes over Finite Rings and Modules, *Fundamentalnaja prikladnaja matematika*(in Russian), **2**(3), (1996), 195–254.
21. R. Nekooei, E. Rostami, On SPAP-rings, *Bull. Iranian Math. Soc.*, **41**, (2015), 907–921.
22. G. Norton , A.Sălăgean, On the Hamming Distance of Linear Codes over a Finite Chain Ring. *IEEE Trans. Inform. Theory*, **46**, (2000), 1060–1067.
23. E. Rostami, A Graphical Characterization for SPAP-Rings, *Iran. J. Math. Sci. Inform.*, **13**(1), (2018), 67–73.
24. K. Shiromoto, L. Storme, A Griesmer Bound for Linear Codes over Finite quasi-Frobenius Rings, *Discrete Applied Mathematics*, **128**, (2003), 263–274.
25. A. Tehranian, H. R. Maimani, A study of the Total graph, *Iran. J. Math. Sci. Inform.*, **6**(2), (2011), 75–80.
26. J. Wood, Duality for Modules over Finite Rings and Applications to Coding Theory, *Amer. J. Math.*, **121**, (1999), 555–575.

[Downloaded from ijmsi.com on 2026-11-16]