LATTICES AND BLOCK CODES

Yulong Fu[1], Xiao Long Xin[2]

Block codes have been wildly used in error-correcting area of information communication for many years. Recently, some researchers found that the using of lattices may reduce the bottleneck of block codes, the lattices codes may be considered for the future 5G. However, the researches on this topic are still in its infancy. In this article, we considered a different encode/decode method by using lattices theory. We first introduced and studied a lattice-valued function on a set, by which we can generate binary block codes. Moreover we discuss how to get the lattices arising from binary block codes. We introduce the notion of semigroup codes and prove that any binary semigroup code V is a lattice in the order " $\leq_c$ ". From such lattice we can construct a lattice function f which determines a binary block code V_1 and $(V_1, \leq_c)$ is isomorphic to $(V, \leq_c)$. For the special semigroup code V, we can get a lattice function f such that f determines a binary block code V_1 and $V_1 = V$. Through the above arguments, we set up a connection between lattices and block codes, by which we can find new block codes by using properties and structures of lattices.

Keywords: lattice; block code; lattice function; q-cut; semigroup code

1. Introduction

In coding theory, *block codes* belong to the most popular and efficient type [5]. Many wildly used real communication coding methods, such as the low-density parity-check (LDPC) and Hamming codes are both in block codes family. Especially some schemes of block codes are selected as the forward error correcting (FEC) systems in multiple communication standards, such as IEEE 802.3an, 802.11n, 802.15, 802.16, ETSI 2nd Gen. DVB, 3GPP LTE (4G) and ITU-T G.9960 and G.709 to name a few [1], the researches on *block codes* are very important.

In a regularly scheme of FEC, the sender can add redundancy to a message, and the receiver can decode it with minimal errors, provided that the information rate would not exceed the channel capacity. While, in a scheme of block codes, the codewords will have a fixed length (unlike the source coding schemes such as Huffman coding, and unlike channel coding methods like convolutional encoding), regardless the length of the original message is. Typically, a block code takes a k-digit information word, and transforms it into an n-digit codeword. Block coding is the primary type of channel coding used in earlier mobile communication systems. A block code is a code which encodes strings formed an alphabet set S into code words by encoding each letter of S separately.

[1]School of Cyber Engineering, Xidian University, Xian, 710071, China, e-mail: ylfu@xidian.edu.cn

[2]School of Mathematics, Northwest University, Xian, 710127, China, e-mail: xlxin@nwu.edu.cn

Recently, in paper [6], Y.B. Jun and S.Z. Song give a method to construct a finite binary block-codes by using of a finite BCK-algebra. At the end of the paper [6], they pose an open question on whether the converse of this statement is also true, that is, we can get the lattices arising from binary block codes. In the paper [3], the authors claim that, in some circumstances, the above question can be solved. But we think that their proof may be incorrect.

In this paper we shall introduce the notion of lattice-valued functions, by which we can get the binary block codes based by use of lattices and also the lattices arising from binary block codes.

The rest of the paper is organized as follows. In Section II, we introduced the preliminary definitions of our method. In Section III, the detail considerations about the lattice-valued functions are given. In Section IV, the processes of block codes induced by lattice functions will be discussed herein, Then in Section V, the methods for lattices arising from block codes are presented. And finally, we conclude the article and propose our future work in the last section.

2. **Preliminaries**

Definition 2.1. [2, 4] *Assume that L is a nonempty set and "$\wedge$" and "$\vee$" are binary operations on L. We call L a lattice, if it satisfies the following: for all $x, y, z \in L$*
(1) $x \wedge x = x$, $x \vee x = x$;
(2) $x \wedge y = y \wedge x$, $x \vee y = y \vee x$;
(3) $(x \wedge y) \wedge z = x \wedge (y \wedge z)$, $(x \vee y) \vee z = x \vee (y \vee z)$;
(4) $(x \wedge y) \vee x = x$, $(x \vee y) \wedge x = x$.

Let L and M be lattices. A map $\theta : L \to M$ is called a lattice-homomorphism (or homomorphism) if it satisfies $\theta(x \wedge y) = \theta(x) \wedge \theta(y)$ and $\theta(x \vee y) = \theta(x) \vee \theta(y)$ for all $x, y \in L$. Moreover a homomorphism is called an isomorphism if it is a bijection. Note that a function θ is a lattice isomorphism if and only if θ is a bijection and bilateral order preserving.

3. **Lattice-valued functions**

From now on, we mean that A is a nonempty set and L is a bounded lattice, unless otherwise specified.

Definition 3.1. *We call a map $\tilde{A} : A \to L$ a lattice-valued function (briefly, lattice function) on A.*

Definition 3.2. *A map $\tilde{A}_q : A \to \{0, 1\}$ is said to be a cut function if it satisfies the following: $\tilde{A}_q(x) = 1$ iff $\tilde{A}(x) \leq q$ for all $x \in A$.*

Define $A_q := \{x \in A | \tilde{A}(x) \leq q\}$, called a cut subset or a q-cut of $\bar{A}$. Note that $A_1 = A$.

Proposition 3.1. *If L is a complete lattice then $\tilde{A}(x) = \inf\{q \in L | \tilde{A}_q(x) = 1\}$, for all $x \in A$.*

Proof. The proof is easy and we omit it. □

Proposition 3.2. *Assume that $\tilde{A} : A \to L$ is a lattice function on A. If $q \leq p$, then $A_q \subseteq A_p$, for all $p, q \in L$.*

Proof. The proof is easy and we omit it. □

Proposition 3.3. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. Then*
(1) $\tilde{A}(x) \neq \tilde{A}(y)$ iff $A_{\tilde{A}(x)} \neq A_{\tilde{A}(y)}$, for all $x, y \in A$,
(2) $\tilde{A}(x) \leq q$ iff $A_{\tilde{A}(x)} \subseteq A_q$, for all $q \in L$ and $x \in A$.

Proof. (1) ($\Leftarrow$) It is clear.
($\Rightarrow$) Let $\tilde{A}(x) \neq \tilde{A}(y)$ for $x, y \in A$. Then $\tilde{A}(x) \not\leq \tilde{A}(y)$ or $\tilde{A}(y) \not\leq \tilde{A}(x)$. Hence $A_{\tilde{A}(x)} = \{z \in A | \tilde{A}(z) \leq \tilde{A}(x)\} \neq \{z \in A | \tilde{A}(z) \leq \tilde{A}(y)\} = A_{\tilde{A}(y)}$.
(2) ($\Rightarrow$) It follows from Proposition 3.2.
($\Leftarrow$) Let $A_{\tilde{A}(x)} \subseteq A_q$ for $q \in L$ and $x \in A$. If $\tilde{A}(x) \not\leq q$, then $x \notin A_q$. Since $\tilde{A}(x) \leq \tilde{A}(x)$, we have $x \in A_{\tilde{A}(x)}$. Hence $A_{\tilde{A}(x)} \nsubseteq A_q$, we have a contradiction. □

Corollary 3.1. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. We have $\tilde{A}(x) \leq \tilde{A}(y)$ iff $A_{\tilde{A}(x)} \subseteq A_{\tilde{A}(y)})$, for all $x, y \in A$.*

Let $\tilde{A}$ be a lattice function. Define $A_L := \{A_q | q \in L\}$, $\tilde{A}_L := \{\tilde{A}_q | q \in L\}$.

Proposition 3.4. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. We have*

$$(\forall Y \subseteq L)(\exists \inf Y \in L \Rightarrow A_{\inf\{q|q \in Y\}} = \cap\{A_q | q \in Y\}).$$

Proof. Let $Y \subseteq L$ and $\inf Y$ exists in L. Then we have
$x \in A_{\inf\{q|q\in Y\}} \Leftrightarrow \tilde{A}(x) \leq \inf\{q | q \in Y\} \Leftrightarrow (\forall r \in Y)(\tilde{A}(x) \leq r) \Leftrightarrow (\forall r \in Y)(x \in A_r) \Leftrightarrow x \in \cap\{A_q | q \in Y\}$.
This completes the proof. □

Corollary 3.2. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. We have*

$$(\forall p, q \in L)(A_p \cap A_q \in A_L).$$

Proposition 3.5. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. We have*

$$\cup\{A_q | q \in L\} = A$$

Proof. The proof is easy and we omit it. □

Proposition 3.6. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. We have*

$$(\forall x \in A)(\cap\{A_q | x \in A_q\} \in A_L).$$

Proof. The proof is easy and we omit it. □

Let $\tilde{A}: A \to L$ be a lattice function on A. Define a binary relation $\sim$ by
$(\forall p, q \in L)(p \sim q \Leftrightarrow A_p = A_q)$.
We can get that $\sim$ is clearly an equivalence relation on L. Define
$\tilde{A}(A) := \{q \in L | \tilde{A}(x) = q, for\ some\ x \in A\}$.
For $q \in L$, Define
$(q] := \{x \in L | x \leq q\}$.

Proposition 3.7. *Assume that $\tilde{A}: A \to L$ is a lattice function on A. Then we have*
$(\forall p, q \in L)(p \sim q \Leftrightarrow (p] \cap \tilde{A}(A) = (q] \cap \tilde{A}(A))$.

Proof. The proof is easy and we omit it. □

4. **Block codes induced by lattice functions**

Let $p \in L$. Define $p/\sim := \{q \in L | p \sim q\}$, which is an equivalence class containing p.

Lemma 4.1. *Assume that* $\tilde{A} : A \to L$ *is a lattice function on* A. *Then for each* $x \in A$, *we have* $\tilde{A}(x) = \sup\{\tilde{A}(x)/\sim \cap \tilde{A}(A)\}$.

Proof. Obviously $\tilde{A}(x) \in \{\tilde{A}(x)/\sim \cap \tilde{A}(A)\}$. Let $p \in \{\tilde{A}(x)/\sim \cap \tilde{A}(A)\}$. Then $p = \tilde{A}(y)$ for some $y \in A$ and $p \sim \tilde{A}(x)$. Therefore $A_{\tilde{A}(x)} = A_p$. Since $y \in A_p$, then $y \in A_{\tilde{A}(x)}$. Thus $\tilde{A}(y) \leq \tilde{A}(x)$, and so $p \leq \tilde{A}(x)$. We complete the proof. □

Assume that L is a finite lattice and $A = \{1, 2, \cdots, n\}$. We can induce a binary block-code V with length n by a lattice function $\tilde{A} : A \to L$ on A, as the following: Foe each $x/\sim$, where $x \in L$, construct a codeword $v_x = x_1x_2\cdots x_n$ such that $x_i = j \Leftrightarrow \tilde{A}_x(i) = j$, for $i \in A$ and $j \in \{0, 1\}$. Define an order relation $\leq_c$ on V by

$$v_x \leq_c v_y \Leftrightarrow x_i \leq y_i \text{ for } i = 1, 2, \cdots, n \qquad (4.1),$$

where $v_x = x_1x_2\cdots x_n$ and $v_y = y_1y_2\cdots y_n$ and $v_x, v_y \in V$.

Lemma 4.2. *Assume that* $\tilde{A} : A \to L$ *is a lattice function on* A.
(1) For $x, y \in A$, *if* $\tilde{A}(x), \tilde{A}(y)$ *are incomparable, then* $\tilde{A}_{\tilde{A}(x)}, \tilde{A}_{\tilde{A}(y)}$ *are incomparable in the order* $\leq_c$;
(2) $(\forall x, y \in A)(\tilde{A}(x) \leq \tilde{A}(y) \Rightarrow \tilde{A}_{\tilde{A}(x)} \leq_c \tilde{A}_{\tilde{A}(y)})$;
(3) $(\forall x, y \in A)(\tilde{A}_{\tilde{A}(x)} \leq_c \tilde{A}_{\tilde{A}(y)} \Rightarrow \tilde{A}(x) \leq \tilde{A}(y))$.

Proof. (1) Let $x, y \in A$ and $\tilde{A}(x), \tilde{A}(y)$ are incomparable. Denote $\tilde{A}(x) = p, \tilde{A}(y) = q$. Then we have that $\tilde{A}_p(x) = 1, \tilde{A}_p(y) = 0$ and $\tilde{A}_q(x) = 0, \tilde{A}_q(y) = 1$. This means that $\tilde{A}_p, \tilde{A}_q$ are incomparable. Therefore $\tilde{A}_{\tilde{A}(x)}, \tilde{A}_{\tilde{A}(y)}$ are incomparable in the order $\leq_c$.
(2) Let $x, y \in A$. Denote $\tilde{A}(x) = p, \tilde{A}(y) = q$. If $\tilde{A}(x) \leq \tilde{A}(y)$, then $A_p \subseteq A_q$ by Proposition 3.1. If $\tilde{A}_p(z) = 1$ for $z \in A$, then $z \in A_p \subseteq A_q$ and so $z \in A_q$. This means that $\tilde{A}_q(z) = 1$ and hence $\tilde{A}_p \leq_c \tilde{A}_q$.
(3) Let $\tilde{A}_p \leq_c \tilde{A}_q$. By (1), $\tilde{A}(x), \tilde{A}(y)$ are comparable. If $\tilde{A}(x) > \tilde{A}(y)$, by (2) we get $\tilde{A}_p > \tilde{A}_q$, which contradicts the hypothesis. Therefore we get $\tilde{A}(x) \leq \tilde{A}(y)$. □

Theorem 4.1. *Let* L *be a finite lattice. Then it determines a block-code* V *such that* $(L, \leq)$ *and* $(V, \leq_c)$ *are lattice isomorphic, where* $\leq_c$ *is given by Eq. (4.1).*

Proof. Denote $L = \{a_1, a_2, \cdots, a_n\}$, in which a_1 and a_n are the least element and the greatest element, respectively. Let $A = \{1, 2, \cdots, n\}$ and $\tilde{A} : A \to L$ be the lattice function on A defined by $\tilde{A}(i) = a_i$ for $i = 1, 2, \cdots, n$. The decomposition of $\tilde{A}$ gives a set $V = \{\tilde{A}_q | q \in L\}$. Then $(V, \leq_c)$ is the desired code, where the order $\leq_c$ is given by Eq. (4.1). Define $f : L \to \{\tilde{A}_q | q \in L\}$ by $f(q) = \tilde{A}_q$ for all $q \in L$. By Lemma 4.1, each $\sim$-class contains only one element. Hence f is one-to-one. By Lemma 4.2, f is bilateral order preserving. Therefore f is a lattice isomorphism. □

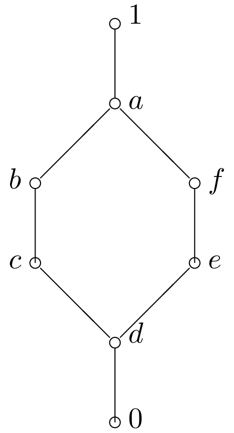
Example 4.1. *Consider the lattice* L *with the universe* $\{0, a, b, c, d, e, f, 1\}$. *Lattice ordering is such that* $0 < d < c < b < a < 1$ *and* $0 < d < e < f < a < 1$, *and* *elements from $\{b, c\}$ and $\{e, f\}$ are pairwise incomparable. Let $\tilde{A} : L \to L$ be a lattice function on L given by*

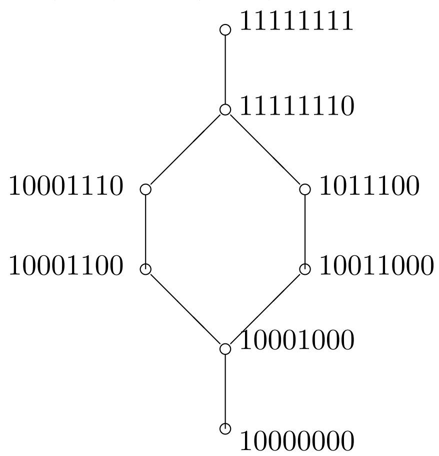
$$\tilde{A} = \begin{pmatrix} 0 & a & b & c & d & e & f & 1 \\ 0 & a & b & c & d & e & f & 1 \end{pmatrix}.$$

Then

$\tilde{A}_L$	0	a	b	c	d	e	f	1
$\tilde{A}_0$	1	0	0	0	0	0	0	0
$\tilde{A}_a$	1	1	1	1	1	1	1	0
$\tilde{A}_b$	1	0	1	1	1	0	0	0
$\tilde{A}_c$	1	0	0	1	1	0	0	0
$\tilde{A}_d$	1	0	0	0	1	0	0	0
$\tilde{A}_e$	1	0	0	0	1	1	0	0
$\tilde{A}_f$	1	0	0	0	1	1	1	0
$\tilde{A}_1$	1	1	1	1	1	1	1	1

Therefore the binary block-code V, determined by the lattice function $\tilde{A}$, is the following form: $V = \{10000000, 11111110, 10111000, 10011000, 10001000, 10001100, 10001110, 11111111\}$ and the diagrams of $(L, \leq)$ and $(V, \leq_c)$ are in the following:





5. **Lattices arising from block codes**

Assume that V is a binary block-code of length n. For $c_1 = (x_1 x_2 \cdots x_n)$, $c_2 = (y_1 y_2 \cdots y_n) \in V$, define

$c_1 * c_2 = ((x_1 \times y_1)(x_2 \times y_2) \cdots (x_n \times y_n))$,

$\theta = (10 \cdots 0), \mathbf{1} = (11 \cdots 1)$,

where $\times$ are ordinary multiplication operator.

Definition 5.1. *Assume that V is a binary block-code of length n. V is said a semigroup code if it satisfies the following conditions:*
(1) $\theta, \mathbf{1} \in V$,
(2) if $c_i = (x_1 x_2 \cdots x_n) \in V$, then $x_1 = 1$,
(3) for all $c_1, c_2 \in V$, $c_1 * c_2 \in V$.

Example 5.1. *Consider a binary block-code*

$$V = \{10000000, 11111110, 10111000, 11000110, 11111111\}.$$

We can check that V is a semigroup code.

Proposition 5.1. *Let V be a semigroup code of length n. Then $(V, *)$ is a semigroup with the identity $\mathbf{1}$.*

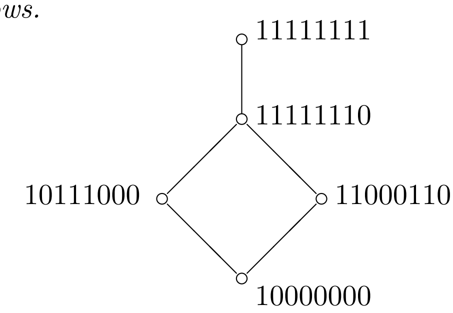
Proof. Straightforward. □

Indeed, semigroup codes have the stronger structures.

Theorem 5.1. *Let V be a semigroup code of length n. Then $(V, \leq_c)$ forms a lattice in which θ is the minimum element and $\mathbf{1}$ is the maximal element, where $\leq_c$ is the order given by Eq.(4.1).*

Proof. Let $a, b \in V$ and $a = (x_1, x_2, \cdots, x_n)$, $b = (y_1, y_2, \cdots, y_n)$. Obviously $a*b \leq a$ and $a * b \leq b$. Assume that $c = (z_1, z_2, \cdots, z_n) \in V$ and $c \leq a, b$. Then $z_i \leq x_i$ and $z_i \leq y_i$ and hence $c_i \leq x_i \times y_i$, that is $c \leq a * b$. This shows that $a * b$ is the infimum of a and b. Therefore $(V, \leq_c)$ forms a meet semi-lattice. Since V has the greatest element $\mathbf{1}$, $(V, \leq_c)$ becomes a lattice. □

Example 5.2. *The semigroup code $(V, \leq_c)$ given in Example 5.1 is a lattice with Hasse diagram as follows.*



Theorem 5.2. *Assume that V is a semigroup code, in which there are m codewords of length n. Then there exist a set A with m elements, a lattice L and a lattice function $f : A \to L$ such that f determines a binary block code V_1, and $(V_1, \leq_c)$ and $(V, \leq_c)$ are lattice isomorphic.*

Proof. Let $V = \{a_1, a_2, \cdots, a_m\}$ be a semigroup code. By Theorem 5.1, $(V, \leq_c)$ is a lattice. We consider $A = V$ and the identity map $f : A \to V$, such that $f(w) = w$ as a lattice function. By Theorem 4.1, f determines a binary block code V_1 and $(V_1, \leq_c)$ is lattice isomorphic to $(V, \leq_c)$. □

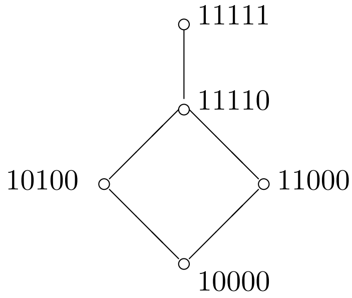
Example 5.3. *Consider the semigroup code $(V, \leq_c)$ given in Example 5.1. By Example 5.2, $(V, \leq_c)$ is a lattice.*

Let $A = \{x, y, z, u, v\}$ and let $\tilde{A} : A \to V$ be a lattice function on A defined by

$$\tilde{A} = \begin{pmatrix} x & y & z & u & v \\ 10000000(0) & 10111000(a) & 11000110(b) & 11111110(c) & 11111111(1) \end{pmatrix}$$

Then $\tilde{A}$ determines a binary block code V_1 by following table and $(V_1, \leq_c)$ has the following Hasse Diagram.

$\tilde{A}_V$	x	y	z	u	v
	0	a	b	c	1
$\tilde{A}_0$	1	0	0	0	0
$\tilde{A}_a$	1	1	0	0	0
$\tilde{A}_b$	1	0	1	0	0
$\tilde{A}_c$	1	1	1	1	0
$\tilde{A}_1$	1	1	1	1	1



Assume that V is a binary block-code which has n codewords of length n. Define the matrix $M_V = (m_{i,j})_{i,j\in\{1,2,\cdots,n\}} \in M_n(\{0,1\})$ where the rows consist of the codewords of V. We call M_V as the matrix associated to the code V.

Theorem 5.3. *Assume that V is a binary block-code which has n codewords of length n. If the matrix M_V is upper triangular with $m_{ij} = 1$, for all $i, j \in \{1, 2, \cdots, n\}$ and $i \leq j$, then there are a set A with n elements, a lattice L and a lattice function $f : A \to L$ such that f determines V.*

Proof. Let $V = \{w_1, w_2, \cdots, w_n\}$, with $w_1 \leq_c w_2 \leq_c \cdots \leq_c w_n$ where the order $\leq_c$ is as in Eq. (4.1). From here, we obtain that $w_1 = \underbrace{10\cdots0}_{(n-1)\ time}$ and $w_n = \underbrace{11\cdots1}_{n\ time}1$. We remark that $w_1 = \theta$ is a minimum element and $w_n = 1$ is a maximal element in $(V, \leq_c)$. We can see that V is a semigroup code. By Theorem 5.1 $(V, \leq_c)$ is a lattice. We consider $A = V$ and the identity map $f : A \to V$, such that $f(w) = w$ as a lattice function. Using Theorem 5.2, f determines f a binary block code V_1. We prove $V = V_1$. For $w_i \in V$, we have $f_{w_i}(w_j) = 1$ iff $f(w_j) \leq w_i$ iff $w_j \leq w_i$. Therefore for $j = 1, 2, \cdots, i$ $w_j \leq w_i$ and so $f_{w_i}(w_j) = 1$ for $j = 1, 2, \cdots, i$. That is $f_{w_i}(w_j) = 1$ for $j = 1, 2, \cdots, i$. This shows that $f_{w_i} = \underbrace{11\cdots1}_{i\ times}0\cdots0 = w_i$, for $i = 1, 2, \cdots, n$. This shows $V = V_1$. □

Example 5.4. *Consider a binary block-code $V = \{10000, 11000, 11100, 11110, 11111\} \doteq \{0, a, b, c, 1\}$. Let $A = V$ and let $f : A \to V$ be the identity map. Then f determines a binary block code $V_1 = \{f_0, f_a, f_b, f_c, f_1\}$ as follows.*

f_V	0	a	b	c	1
f_0	1	0	0	0	0
f_a	1	1	0	0	0
f_b	1	1	1	0	0
f_c	1	1	1	1	0
f_1	1	1	1	1	1

We can see that $V = V_1$.

6. Conclusion

In this paper, by use of the notion of lattice-valued functions, we established block-codes. Conversely we proved that to each semigroup code V we can associate a lattice L such that the binary block-code V_L, generated by L, is isomorphic to V. In some particular case, we have $V_L = V$.

Future research will focus on finding new codes by using lattice-valued functions. It is important method that various classical error-correcting codes are constructed by ideals. As well known, all cyclic codes are principal ideals and other classes of codes also are ideals in group algebras. By using other algebraic structure, ones have developed faster encoding and decoding algorithms for these codes (see, for example, [7, 8, 9]). We also will construct certain codes by use of the ideal theory of lattice-algebras.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (11571281,61602359), China Postdoctoral Science Foundation (2015M582618), China 111 Project (B16037) and the Fundamental Research Funds for the Central Universities (80027215617901,80027215617903).

REFERENCES

[1] *J. Adnrade, G. Falcao, V. Silva and L. Sousa*, *A Survey on Programmable LDPC Decoders*, IEEE Access, 2016.

[2] *G. Birkhoof*, *Lattice Theory*, American Mathematical Society Colloquium, 1940.

[3] *C. Flaut*, BCK-algebras arising from block codes, Journal of Intelligent and Fuzzy Systems, vol. **28**(2015), 1829-1833.

[4] *G. Grätzer*, Lattice Theory. W H Freeman and Company, San Francisco, 1979.

[5] *L. Hu, C. Duau, D. Zhao and X. Liao*, *Study Status and Prospect of Lattice Codes in Wireless Communication*, 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control, 2016.

[6] *Y.B. Jun, S.Z. Song*, Codes based on BCK-algebras, Information Sciences, vol. **181**(2011), 5102-5109.

[7] *A.V. Kelarev, P. Solé*, Error-correcting codes as ideals in group ring, Contemp. Math., vol. **273**(2001), 11-18.

[8] *P. Landrock, O. Manz*, Classical codes as ideals in group algebras, Des. Codes Cryptogr., vol. **2**(1992), 273-285.

[9] *H.N. Ward*, Visible codes, Arch. Math., vol. **54**(1990), 307-312.