Advances in Mathematics of Communications
Volume 11, No. 1, 2017, 123–137
doi:10.3934/amc.2017007

ON THE MULTIPLE THRESHOLD DECODING OF LDPC CODES OVER GF(q)

Alexey Frolov

Skolkovo Institute of Science and Technology (Skoltech)
and
Institute for Information Transmission Problems
Russian Academy of Sciences, Moscow, Russia

Victor Zyablov

Institute for Information Transmission Problems
Russian Academy of Sciences, Moscow, Russia

(Communicated by Vitaly Skachek)

Abstract. We consider decoding of LDPC codes over GF(q) with a hard-decision low-complexity majority algorithm, which is a generalization of the bit-flipping algorithm for binary LDPC codes. A modification of this algorithm with multiple thresholds is suggested. A lower estimate on the decoding radius realized by the new algorithm is derived. The estimate is shown to be better than the estimate for a single threshold majority decoder. At the same time, introducing multiple thresholds does not affect the order of decoding complexity.

1. Introduction

Low-density parity-check (LDPC) codes [8, 20] over GF(q) have advantages over binary LDPC codes. Davey and MacKay [5] were first who used belief propagation (BP) to decode such codes. They showed that non-binary LDPC codes significantly outperform their binary counterparts. Moreover, non-binary LDPC codes are especially good for the channels with burst errors and high-order modulations [19]. Unfortunately, their decoding complexity is still large, that is why iterative hard and soft-reliability based decoding majority algorithms (the terminology is from [16]) are of considerable interest for high-throughput practical applications.

In this paper we investigate the error-correcting capabilities of non-binary LDPC codes decoded with a hard-decision low-complexity majority algorithm. We perform the worst case analysis and estimate the decoding radius realized by this algorithm. By the decoding radius we mean the number of errors which is guaranteed to be corrected. Zyablov and Pinsker [25] were the first who showed that random binary LDPC codes of growing length can correct a non-vanishing fraction of errors. The improvements are given in [2, 4, 18, 23]. The lower estimate on the decoding radius

2010 *Mathematics Subject Classification:* Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases: Coding theory, iterative decoding, LDPC codes, majority logic decoding, threshold decoding, decoding radius.
The research was carried out at the IITP RAS and supported by the Russian Science Foundation (project no. 14-50-00150).
A part of the results of this paper were presented at the 2015 IEEE International Symposium on Information Theory, June 2015, Hong Kong [7].

of irregular binary LDPC codes was given in [17] and for generalized LDPC codes with Hamming component code – in [24]. Note, that in the binary case the majority algorithm is usually called the bit-flipping algorithm [18, 25]. In the non-binary case the majority decoding algorithm and the lower estimate on the decoding radius were given in [6].

Here we consider the decoding of LDPC codes over GF(q) with the low-complexity majority algorithm from [6]. In [6, Theorem 1] a lower estimate on the relative decoding radius ρ realized by the low-complexity majority algorithm is derived. Let us describe the result in more detail. Let N denote the code length. In [6] it is proved that there exist LDPC codes over GF(q) of sufficiently large length N capable of correcting any error vector of weight $W \leq \rho N$ with the decoding complexity $O(N \log N)$. Here and in what follows by weight we mean the Hamming weight, i.e. a number of non-zero elements in a vector.

We first improve the estimate on ρ. Then we consider multiple threshold decoding of LDPC codes over GF(q). Multiple threshold majority decoding for binary LDPC codes was first introduced in [11]. It was shown that transition to multiple thresholds increases the decoding radius of the majority algorithm without affecting the order of complexity. In this paper we generalize the ideas of [11] to the case of non-binary LDPC codes. We note, that multiple threshold decoding is also possible in presence of soft information from the channel (see e.g. [9, 16, 21, 22]). But we only consider the hard-decision case, i.e. we are given no soft information from the channel and rank the symbols in accordance to the messages from check nodes. In this case we can perform theoretical analysis.

Our contribution is as follows. We first improve the estimate on the relative decoding radius ρ for the single threshold case. Then we suggest the hard-decision majority decoding algorithm with multiple thresholds for LDPC codes over GF(q). A lower estimate on the decoding radius realized by the new algorithm is derived. The estimate is shown (see Tables 1 and 2) to be at least 1.2 times better than the estimate for a single threshold majority decoder (for sure, in case the single threshold estimate is non-zero). At the same time analogous to the result from [11] the transition to multiple thresholds does not affect the order of complexity.

Note, that the channel model is of no importance for us. Within the paper we proved, that if the number of errors W in the received sequence is less than or equal to ρN, then the sequence is guaranteed to be corrected by the decoder. The nature of errors (or the channel model) is not needed for this task, but we would need it e.g. if we want to calculate the probability of decoder failure.

2. Preliminaries

2.1. LDPC codes over GF(q).

An LDPC code $\mathcal{C}$ of length N over GF(q) is a null-space of an $M \times N$ sparse parity-check matrix $\mathbf{H} = [h_{j,i}]$, $1 \leq j \leq M$, $1 \leq i \leq N$, over GF(q). In this paper we consider regular LDPC codes, i.e the parity-check matrix $\mathbf{H}$ has constant row and column weights of n_0 and ℓ, respectively. The following inequality follows for the rate of the code $\mathcal{C}$

$$R(\mathcal{C}) \geq 1 - \frac{\ell}{n_0}.$$

In what follows we need the following notations:

$$\Gamma(i) = \{j : h_{j,i} \neq 0, 1 \leq j \leq M\}$$

and

$$\Phi(j) = \{i : h_{j,i} \neq 0, 1 \leq i \leq N\}.$$

2.2. TANNER GRAPH. The constructed code $\mathcal{C}$ can be described with the use of a bipartite graph, which is called the Tanner graph [20] (see Fig. 1). The vertex set of the graph consists of the set of variable nodes $V = \{v_1, v_2, \ldots, v_N\}$ and the set of check nodes $C = \{c_1, c_2, \ldots, c_M\}$. The variable node v_i and the check node c_j are connected with an edge if and only if $h_{j,i} \neq 0$. The edge has a label $h_{j,i}$ (see Fig. 1). Thus, all the check nodes have the same degree n_0 and all the variable nodes have the same degree ℓ. Such Tanner graphs are called regular.

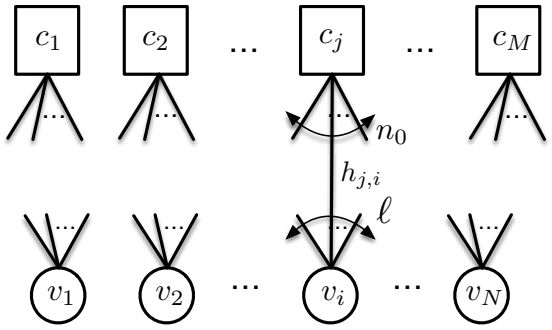

FIGURE 1. Tanner graph

To check if $\mathbf{r} = (r_1, r_2, \ldots, r_N) \in \mathrm{GF}(q)^N$ is a codeword of $\mathcal{C}$ we associate the symbols of $\mathbf{r}$ with the variable nodes $(v_i \leftarrow r_i, i = 1, \ldots, N)$. Each check node c_j, $1 \leq j \leq M$, imposes the following linear restriction

$$c_j : \sum_{t \in \Phi(j)} h_{j,t} r_t = 0$$

and we can say that linear single parity-check (SPC) codes of length n_0 over $\mathrm{GF}(q)$ are associated with the check nodes. In what follows we refer the codes as component codes. The word $\mathbf{r}$ is a codeword of $\mathcal{C}$ if and only if all the component codes are satisfied (the symbols which come to the codes via the edges of the Tanner graph form codewords of the component codes).

In what follows for the sake of simplicity we consider only the case when component codes are SPC codes over $\mathrm{GF}(q)$. The generalization to the case of a stronger component codes is simple. It will be briefly explained in Remark 6.

2.3. SYNDROME WEIGHT ESTIMATES. As usual [14], we calculate the syndrome of the sequence $\mathbf{r} = (r_1, r_2, \ldots, r_N) \in \mathrm{GF}(q)^N$ to be decoded as follows

$$\mathbf{S} = \mathbf{r}\mathbf{H}^T.$$

Let $wt(\mathbf{x})$ denote the (Hamming) weight of a vector $\mathbf{x}$, let W denote the number of errors in the received sequence. We note, that for any q-ary LDPC code $\mathcal{C}$ the following trivial *upper bound on the syndrome weight* holds

$$wt(\mathbf{S}) \leq U(W) \triangleq W\ell.$$

To formulate the lower bound on the syndrome weight we need to define the ensemble $\mathcal{E}(N, n_0, \ell)$ of LDPC codes over $\mathrm{GF}(q)$. We start with the ensemble of

binary LDPC codes suggested by Gallager [8]. All the elements of non-binary ensemble $\mathcal{E}(N, n_0, \ell)$ can be obtained as follows: we replace ones in parity-check matrices of codes from the Gallager's ensemble with arbitrarily elements of $\mathrm{GF}^*(q) = \mathrm{GF}(q)\backslash\{0\}$.

We need to introduce some notations:

For a real number $0 \le x \le 1$ let

$$h_q(x) = -x\log_q x - (1-x)\log_q(1-x) + x\log_q(q-1)$$

be q-ary entropy function.

Let $g_0(s, n_0)$ be the component code weight enumerator. In our case (case of q-ary SPC code)

$$g_0(s, n_0) = \frac{1}{q}\left(1 + (q-1)s\right)^{n_0} + \frac{q-1}{q}(1-s)^{n_0}.$$

To see this one need to apply the MacWilliams theorem [14] (the dual code is a q-ary repetition code).

Let $g_1(s, n_0)$ be the weight enumerator of all other words, i.e.

$$g_1(s, n_0) = \left(1 + (q-1)s\right)^{n_0} - g_0(s, n_0).$$

At last, we introduce the function

$$\begin{aligned} F(\alpha, \omega, n_0) &= h_q(\omega) - \frac{1}{n_0}h_q(\alpha\omega n_0) - \alpha\omega\log_q(q-1) \\ &+ \max_{s>0}\left\{\omega\log_q(s) - \frac{1}{n_0}\log_q\left(g_0(s, n_0)\right) - \right. \\ &- \left. \alpha\omega\log_q\left(\frac{g_1(s, n_0)}{g_0(s, n_0)}\right)\right\}, \end{aligned}$$

where the maximum is found over all positive s such that

$$\frac{\alpha\omega n_0}{1 - \alpha\omega n_0} \le \frac{g_1(s, n_0)}{g_0(s, n_0)}.$$

The next theorem gives the *lower bound on the syndrome weight.*

Theorem 1 ([6, Theorem 2])**.** *For any $4 < \ell < n_0$ there exists $\omega^*(n_0, \ell) > 0$ such that:*

- *there are codes in the ensemble $\mathcal{E}(N, n_0, \ell)$ for which the following inequality holds*

$$wt(\mathbf{S}) > L(W) = \frac{W\ell}{2} \tag{1}$$

  for all error vectors of weight W, $0 < W \le W^*(N, n_0, \ell) = \omega^*(n_0, \ell)N$.
- *the number of such codes $(G(N, n_0, \ell))$ satisfy the following relation*

$$\lim_{N\to\infty}\frac{G(N, n_0, \ell)}{|\mathcal{E}(N, n_0, \ell)|} = 1.$$

The value $\omega^*(n_0, \ell)$ is the smallest positive root of equation

$$h_q(\omega) - \ell F(0.5, \omega, n_0) = 0.$$

In what follows for simplicity we omit the parameters N, n_0 and ℓ and use notations W^* and ω^*.

Remark 1 (Interconnection to expander graphs)**.** Note, that (1) means that the underlying Tanner graph is an unbalanced bipartite $(\ell, n_0, \omega^*, \ell/2)$ expander graph (we use the terminology from [18]). This means, that for any subset of variable nodes $V' \subset V$

$$|V'| \leq \omega^* N \Rightarrow |\Gamma(V')| > \frac{\ell}{2}|V'|,$$

where $\Gamma(V') \subset C$ is the set of check nodes connected to the set of variable nodes V'.

The usual way to check the expansion properties of a graph is to examine its second-largest eigenvalue (see e.g. [1]). The explicit constructions with the greatest possible separation between the first and second-largest eigenvalues were given by Margulis [15] and Lubotzky, Phillips, and Sarnak [13]. Unfortunately, the explicit constructions of expander graphs with expansion greater than $\ell/2$ are not known (Kahale [10] even shows that eigenvalue separation cannot certify greater expansion).

In what follows we need just an LDPC code over GF(q) which satisfies the property (1). We denote that code by $\mathcal{C}^*$.

2.4. SINGLE THRESHOLD MAJORITY DECODING. Let us describe a single-threshold majority decoding algorithm from [6]. The algorithm is an iterative hard-decision decoding algorithm. By *iteration* we mean the sequential processing of all N symbols of the sequence to be decoded ($\mathbf{r} = (r_1, r_2, \ldots, r_N)$). It is important to note, that the algorithm works with the symbols in the sequential manner. This means, that in case of replacement all the changes are introduced to the sequence to be decoded and to the syndrome and then the algorithm proceeds to the next symbol. The algorithm is a generalization of bit-flipping algorithm for the binary case. The difference is as follows. In the binary case check nodes can send only binary message to the variable nodes (flip or no flip). In non-binary case check nodes send the error value and the algorithm should take the value into consideration.

Please see Algorithm 1 for full description, here we give some comments and explanations. Assume the algorithm is considering the symbol r_i.

Calculation of messages. The corresponding variable node v_i is connected to ℓ check nodes c_j, $j \in \Gamma(i)$. Each of these nodes sends a message $m_{j\to i}$, $j \in \Gamma(i)$, to v_i. The messages are calculated as follows (see line 6)

$$m_{j\to i} = z_{j,i} - r_i, \;\; j \in \Gamma(i),$$

where $z_{j,i}$ is the value of v_i, such that the check node c_j is satisfied ($s_j = 0$).

It is easy to check, that

$$z_{j,i} = -h_{j,i}^{-1} \left(\sum_{t \in \Phi(j), t \neq i} h_{j,t} r_t \right)$$

and thus, for $j \in \Gamma(i)$

$$m_{j\to i} = -h_{j,i}^{-1} \left(\sum_{t \in \Phi(j), t \neq i} h_{j,t} r_t \right) - r_i = -h_{j,i}^{-1} s_j.$$

Remark 2. We would like to point out, that a message $m_{j\to i}$ is actually an error value, sent by the check mode c_j. Indeed, if we replace r_i with $r_i + m_{j\to i}$, then the syndrome of the j-th component code (s_j) will become zero. If a check node is satisfied, it sends a zero message.

Replacement criterion. Assume the algorithm is considering the symbol r_i and the messages $m_{j\to i}$, $j \in \Gamma(i)$, are calculated in accordance to the rules above. Let $A_{\max}$ denote a subset of equal non-zero messages of maximal cardinality, let $a = |A_{\max}|$ and m be a value of the messages from $A_{\max}$. Let a threshold θ be an integer such that $0 \le \theta < \ell$. At last let z be a number of zero messages. The replacement criterion is as follows. If $a - z > \theta$ (line 11) we replace the symbol r_i with $r_i + m$, the syndrome is updated, and the algorithm continues with the next symbol.

Remark 3. Note, that in accordance to the replacement criterion the syndrome weight is reduced by at least $\theta+1$ with each change. Indeed, $a = |A_{\max}|$ unsatisfied component codes become satisfied, z satisfied check nodes become unsatisfied and $a - z > \theta$.

Remark 4. Note, that within the section $\theta = 0$, we introduced the parameter here just for our convenience. We will use it in what follows.

Stopping criterion. The last thing we have not mentioned yet is a stopping criterion. We stop the algorithm if no changes in $\mathbf{r}$ were made during the iteration. The flag b (see line 14) indicates if there were changes during the iteration. Recall, that in accordance to the replacement criterion the syndrome weight is reduced by at least $\theta + 1$ with each change, that is why the oscillations are not possible and we do not restrict the maximal number of iterations.

Algorithm 1 Single threshold majority decoding algorithm

Input:
received sequence $\mathbf{r}$, threshold $\theta : 0 \le \theta < \ell$
Output:
decoded sequence $\mathbf{c}$, failure flag F

1: $\mathbf{S} \leftarrow \mathbf{r}\mathbf{H}^T$; $b \leftarrow 1$ ▷ Initialization
2: **while** $b = 1$ **do**
3: $b \leftarrow 0$
4: **for all** $1 \le i \le N$ **do**
5: **for all** $j \in \Gamma(i)$ **do**
6: $m_{j\to i} \leftarrow -h_{j,i}^{-1} s_j$ ▷ s_j is a syndrome of a j-th component code
7: **end for**
8: $A_{\max} \leftarrow$ maximal subset of equal non-zero messages
9: $a \leftarrow |A_{\max}|$; $m \leftarrow$ value from $A_{\max}$
10: $z \leftarrow$ number of zero messages
11: **if** $a - z > \theta$ **then**
12: $r_i \leftarrow r_i + m$
13: update $\mathbf{S}$
14: $b \leftarrow 1$ ▷ Replacement occurred, set flag to 1
15: **end if**
16: **end for**
17: **end while**
18: $F \leftarrow 1$
19: $\mathbf{c} \leftarrow \mathbf{r}$
20: **if** $wt(\mathbf{S}) = 0$ **then**
21: $F \leftarrow 0$
22: **end if**

Let us now recall the results of [6].

Lemma 1 ([6, Theorem 3])**.** *Let*

$$wt\,(\mathbf{S}) > \frac{W\ell}{2}$$

then there exists at least one symbol that will be changed by Algorithm 1 *(with* $\theta = 0$*) within one iteration.*

Theorem 2 ([6, Theorem 4])**.** *Let* $\mathcal{C}^*$ *be an LDPC code over GF(q), satisfying (1). If the number of errors in the received sequence*

$$W \le W^*/2,$$

Algorithm 1 *(with* $\theta = 0$*) will correct all the errors with the complexity* $O(N \log N)$*.*

3. Improvement in case of single threshold

We now improve the result of the previous theorem.

Theorem 3 (Single threshold)**.** *Let* $\mathcal{C}^*$ *be an LDPC code over GF(q), satisfying (1). If the number of errors in the received sequence*

$$W \le W^{(S)} = \frac{W^*}{2}\frac{\ell+2}{\ell+1},$$

Algorithm 1 *(with* $\theta = 0$*) will correct all the errors with the complexity* $O(N \log N)$*.*

Proof. To prove the theorem we need to prove that the number of errors at each step of the algorithm is less than or equal to W^* (see condition (1) and Lemma 1).

Any error vector can be mapped to a point of the following coordinate system: "syndrome weight – number of errors" (see Fig. 2). At the same time it is clear, that each point in the coordinate system corresponds to multiple error vectors. First, let us add the lines $L(W)$ and $U(W)$ to Fig. 2. Recall, that the syndrome weight of any error vector with $W \le W^*$ satisfies the inequality

$$L(W) < wt\,(\mathbf{S}) \le U(W).$$

Let us consider the decoding process. It corresponds to some trajectory in the coordinate system. We start from the initial error vector. With each replacement the syndrome weight decreases (we move down at least by 1). There are 3 possibilities for the number of errors:

- we can introduce an error (in this case we move right by 1);
- we can correct an error (in this case we move left by 1);
- as the alphabet is non-binary, we can replace an erroneous symbol with an erroneous one (in this case the number of errors stays the same).

The decoding is successful if we finish at the origin.

The area of correctable error vectors is filled by gray color in Fig. 2. Let us explain this fact. Assume we start from the point C (see Fig. 2) and only introduce errors. In this situation we move right and down by 1 with each step (move along the line CB). We can not come to the point B as it lies on the (strict) lower bound $L(W)$ so it is clear that the number of errors can not become greater than W^*. In this case the decoding (and the trajectory) finishes at origin. To finish the proof we just need to calculate the coordinate of intersection of two lines: $U(W)$ and CB (starts in W^* and has a slope equal to -1). The previous estimate ($W^*/2$, point A) is also shown in Fig. 2.

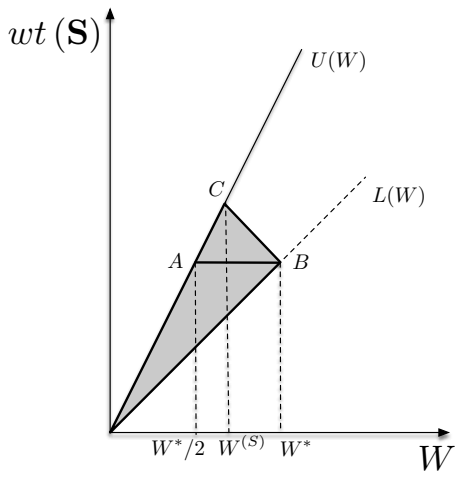

FIGURE 2. Single threshold decoding. The lower estimate $L(W)$ is valid only up to W^*, that is why the line is dashed after the point. Points A, C and B have coordinates $(W^*/2;\, W^*\ell/2)$, $(W^{(S)};\, W^{(S)}\ell)$ and $(W^*;\, W^*\ell/2)$ accordingly.

The proof of the complexity estimate coincides with the proof from [6]. We omit it here. □

Corollary 1. *Let us introduce a notation*

$$\alpha^{(S)} = \frac{\ell + 2}{2(\ell + 1)}$$

and consider the asymptotic ($N \to \infty$) estimate of the relative decoding radius ($\rho^{(S)}$) realized by Algorithm 1*. We have*

$$\rho^{(S)} \geq \frac{W^{(S)}}{N} = \alpha^{(S)}\omega^*.$$

In the next section we will increase the estimate by means of transition to multiple decoding thresholds.

4. Decoding with multiple thresholds

Let us first introduce the sequence of integer thresholds (let $t \geq 1$)

$$0 = \theta_1 < \theta_2 < \ldots < \theta_t < \ell.$$

Now we are ready to describe the multiple threshold decoding algorithm. The idea of the new algorithm is in consequent applications of Algorithm 1 with different replacement thresholds to the sequence to be decoded (see line 3). We start from the largest threshold θ_t and end with $\theta_1 = 0$. Please see Algorithm 2 full description below for more details.

Remark 5. We note, that the implementation of Algorithm 2 is not optimal. It is much better to implement it in such a way. First calculate the syndrome, then sort all the symbols in a descending order of $a - z$ value (see previous section), then change the symbols consequently and update the sorted list. But nevertheless we

Algorithm 2 Multiple threshold majority decoding algorithm

Input:
Input: received sequence $\mathbf{r}$, t thresholds $0 = \theta_1 < \theta_2 < \ldots < \theta_t < \ell$
Output:
decoded sequence $\mathbf{c}$, failure flag F
$\mathbf{S} \leftarrow \mathbf{r}\mathbf{H}^T$ ▷ Initialization
for all $0 \leq i \leq t-1$ **do**
    Apply Algorithm 1 with $\theta = \theta_{t-i}$
    $\mathbf{r} \leftarrow$ output of Algorithm 1
end for
$F \leftarrow 1$
$\mathbf{c} \leftarrow \mathbf{r}$
if $wt(\mathbf{S}) = 0$ **then**
    $F \leftarrow 0$
end if

see here that the complexity of Algorithm 2 is no more than t times the complexity of Algorithm 1.

To estimate the decoding radius of Algorithm 2 we need the following Lemma.

Lemma 2. *Let θ be an integer, $0 \leq \theta < \ell$, let*

$$wt(\mathbf{S}) > P(\theta, W) = W\frac{\ell+\theta}{2}$$

then there exists at least one symbol that will be changed by Algorithm 2 within one iteration with threshold θ.

Proof. Consider a subgraph of the Tanner graph that contains only erroneous symbols (the number of errors is equal to W) and check nodes connected to these symbols. In the proof we work with this subgraph only.

Let us introduce the following notation:

- A is the set of check nodes that detect an error ($|A| = wt(\mathbf{S})$);
- A_i, $i = 1, \ldots, n_0$, is the subset of A containing only the check nodes with precisely i incoming edges ($a_i = |A_i|$);
- $A_{\geq 2} = A \backslash A_1$ is a subset of A containing only check nodes with at least 2 incoming edges ($a_{\geq 2} = |A_{\geq 2}|$);
- C is the set of check nodes that contain errors but do not detect them ($c = |C|$);
- $e_{A_1}^{(i)}$ is the number of edges outgoing from a symbol i and incoming to A_1;
- $e_C^{(i)}$ is the number of edges outgoing from a symbol i and incoming to C.

In Fig. 3 we present an example of a subgraph of the Tanner graph and illustrate the introduced notation.

First note, that if the condition

$$e_{A_1}^{(i)} > e_C^{(i)} + \theta$$

holds for the i-th symbol, then there exists at least one symbol that will be changed by Algorithm 2 within one iteration with threshold θ. To prove this it is sufficient to mention that the codes with a single error will give equal messages.

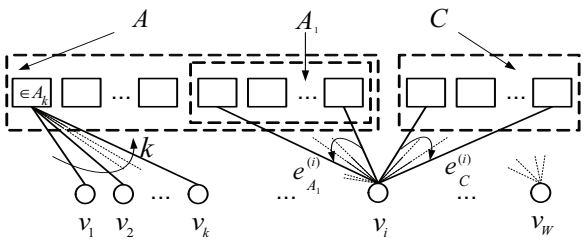

FIGURE 3. A subgraph of Tanner graph

Then we claim that if

$$a_1 > \sum_{i=1}^{W} e_C^{(i)} + W\theta,$$

then there exists a symbol i such that $e_{A_1}^{(i)} > e_C^{(i)} + \theta$.

In order to finish the proof we need to count the edges in the subgraph. The number of edges outgoing from W erroneous symbols is $W\ell$. These edges are adjacent with either check nodes that have detected an error ($A = A_1 \cup A_{\geq 2}$) or with check nodes that have not detected errors but contain them (C). Let us estimate the number of edges incoming to each of the three sets of check nodes:

- The number of edges connected to nodes of the set A_1 is $\sum\limits_{i=1}^{W} e_{A_1}^{(i)} = a_1$;
- The number of edges connected to nodes of the set $A_{\geq 2}$ is at least $2(wt\,(\mathbf{S}) - a_1)$ (here we use the fact every node has at least two incoming edges);
- The number of edges connected to nodes of the set C is $\sum\limits_{i=1}^{W} e_C^{(i)}$.

Thus

$$W\ell \geq a_1 + 2(wt\,(\mathbf{S}) - a_1) + \sum_{i=1}^{W} e_C^{(i)}.$$

After some transformations, we have

$$a_1 - \sum_{i=1}^{W} e_C^{(i)} \geq 2\,wt\,(\mathbf{S}) - W\ell.$$

This immediately implies that if the condition of the Lemma holds then

$$a_1 > \sum_{i=1}^{W} e_C^{(i)} + W\theta. \qquad \square$$

The previous lemma provides us with a sufficient condition for Algorithm 2 to find at least one symbol which satisfies the replacement criterion. The next lemma will show that if there is a gap in between the syndrome weight and the line $P(\theta, W)$, then a linear fraction of symbols will be changed by Algorithm 2.

Lemma 3. *Let θ be an integer, $0 \leq \theta < \ell$, let*

$$wt\,(\mathbf{S}) = P(\theta + \varepsilon, W) = W\frac{\ell + \theta + \varepsilon}{2}$$

then at least δW symbols are changed by Algorithm 2 within one iteration with threshold θ, where

$$\delta = \frac{\varepsilon}{(\ell-\theta)(\ell(n_0-1)+1)}.$$

Proof. In accordance with Lemma 2

$$\sum_{i=1}^{W}\left(e_{A_1}^{(i)} - e_C^{(i)}\right) \geq (\theta+\varepsilon)W$$

Let us estimate the number of symbols (W') which satisfy the replacement criterion. Clearly, $\theta < \left(e_{A_1}^{(i)} - e_C^{(i)}\right) \leq \ell$ for all symbols that will be changed, and $-\ell \leq \left(e_{A_1}^{(i)} - e_C^{(i)}\right) \leq \theta$ for symbols, that will not be changed.

$$W'\ell + (W-W')\,\theta \geq \sum_{i=1}^{W}\left(e_{A_1}^{(i)} - e_C^{(i)}\right) \geq (\theta+\varepsilon)W$$

Thus, we have

$$W' \geq \frac{\varepsilon}{\ell-\theta}W.$$

Recall, that the syndrome is updated after every symbol change. Changing any symbol has an influence on decisions for the others. Assume that symbol i is changed; let us estimate the number of symbols that can be affected. Symbol i is connected to ℓ component codes, which can contain at most $\ell(n_0-1)$ symbols other than i. Thus, in the worst case δW symbols will be changed within one iteration with threshold θ. □

Theorem 4 (Multiple thresholds)**.** *Let $\mathcal{C}^*$ be an LDPC code over GF(q), satisfying (1). Let $0 = \theta_1 < \theta_2 < \ldots < \theta_t < \ell$ be a sequence of thresholds. If the number of errors in the received sequence fulfills*

$$W \leq W_{t+1},$$

where

$$W_i = W_{i-1}\frac{\ell+3\theta_{i-1}+2}{\ell+2\theta_{i-1}+\theta_i+2},\ W_1 = W^*, \theta_{t+1} = \ell,$$

Algorithm 2 corrects all the errors with complexity $O(N\log N)$.

Proof. The proof consists of two parts. First we estimate the decoding radius and then we consider the complexity of the decoding algorithm.

Let us start with the *decoding radius*. The area of correctable error vectors is shown in Fig. 4. For now the area is more difficult because the slope at threshold θ_i is equal to $\theta_i + 1$. To prove the Theorem we need to consequently calculate coordinates of intersection of the area bound and lines $P(\theta_i, W)$.

We now consider the *decoding complexity*. It is clear, that the complexity of an iteration is $O(N)$, so our aim is to estimate the number of iterations. Assume the syndrome weight is $P(\theta+\varepsilon, W)$ and consider an iteration with threshold θ. In accordance to Lemma 3 δW symbols will be changed by Algorithm 2, thus the syndrome weight will be reduced at least

$$\gamma = \frac{P(\theta+\varepsilon,W)}{P(\theta+\varepsilon,W)-\delta\theta W} = \frac{\ell+\theta+\epsilon}{\ell+\theta+\epsilon-2\delta\theta} > 1$$

times. Note, that $\gamma > 1$ for any θ. This means, that the required number of iterations is logarithmic and the complexity is $O(N\log N)$. □

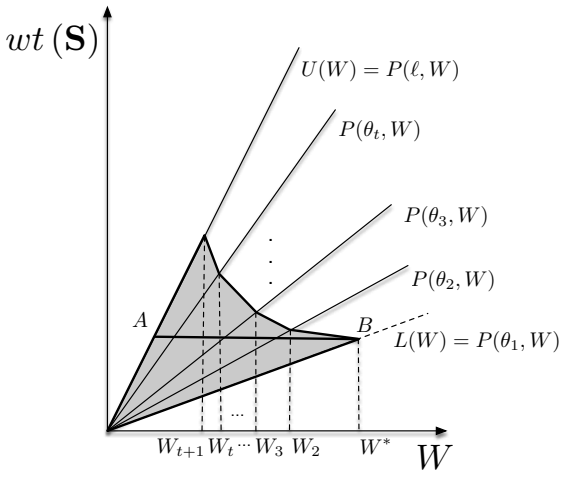

FIGURE 4. Multiple thresholds. The lower estimate $L(W)$ is valid only up to W^*, that is why the line is dashed after the point.

The most interesting case for us is the one where we have all the thresholds from 0 to $\ell - 1$. In this case

$$W^{(M)} = \left(\prod_{i=1}^{\ell} \frac{\ell + 3i - 1}{\ell + 3i} \right) W^*.$$

Let us introduce a notation

$$\alpha^{(M)} = \prod_{i=1}^{\ell} \frac{\ell + 3i - 1}{\ell + 3i}$$

and consider the asymptotic ($N \to \infty$) estimate of the relative decoding radius ($\rho^{(M)}$) realized by Algorithm 2 (when we have all the thresholds). We have

$$\rho^{(M)} \geq \frac{W^{(M)}}{N} = \alpha^{(M)} \omega^*.$$

In Fig. 5 the comparison of $\alpha^{(S)}$ and $\alpha^{(M)}$ is shown.

At last let us estimate $\alpha^{(M)}$.

Lemma 4. *The following inequalities hold*

$$0.630... = \sqrt[3]{\frac{1}{4}} \leq \alpha^{(M)} \leq \sqrt[3]{\frac{\ell + 2}{4\ell + 2}}.$$

Proof. For $j \in \{-2, -1, 0, 1, 2\}$ let us define the following functions

$$f_j = \prod_{i=1}^{\ell} \left(1 - \frac{1}{\ell + 3i + j} \right).$$

Note, that $\alpha^{(M)} = f_0$. It is easy to check, that

$$f_{-2} f_{-1} f_0 \leq f_0^3 \leq f_0 f_1 f_2,$$

as $f_{-2} \leq f_{-1} \leq f_0 \leq f_1 \leq f_2$.

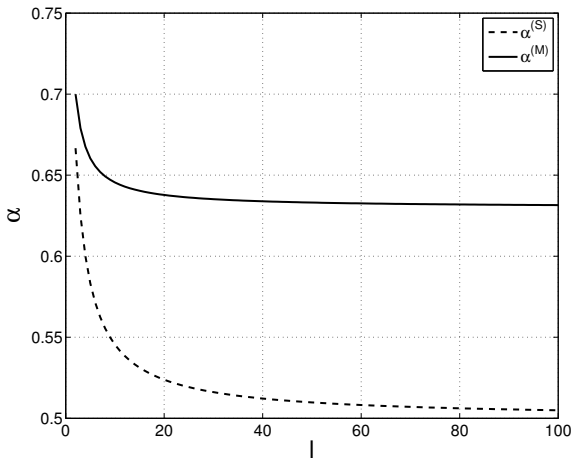

FIGURE 5. The dependency of $\alpha^{(S)}$ and $\alpha^{(M)}$ on ℓ

To complete the proof, we observe that

$$f_{-2}f_{-1}f_0 = \prod_{i=1}^{3\ell}\left(1-\frac{1}{\ell+i}\right) = \frac{1}{4},$$

and

$$f_0f_1f_2 = \prod_{i=1}^{3\ell}\left(1-\frac{1}{\ell+i+2}\right) = \frac{\ell+2}{4\ell+2}. \qquad \square$$

Remark 6 (Generalized LDPC codes [3, 12, 20])**.** Here we briefly consider the case of generalized LDPC codes, i.e. the case when the component codes are not SPC codes but some more powerful codes. All our theorems work in this case if we use the so-called generalized syndrome rather then an ordinary syndrome. Generalized syndrome consists of syndromes of component codes. The weight of generalized syndrome is just the number of unsatisfied component codes. We would like to point out, that under our analysis (analogous to [6]) the transition to generalized LDPC codes does not lead to a gain in the decoding radius.

5. NUMERICAL RESULTS

The numerical results are given in Table 1 for $q = 16$ and Table 2 for $q = 64$. In each Table the dependencies of δ (the relative minimum distance), ω^*, $\rho^{(S)}$ and $\rho^{(M)}$ on the code rate R are presented. Note, that ℓ (in each case) is chosen to maximize the functions. For our case the maximal values of ω^*, $\rho^{(S)}$ and $\rho^{(M)}$ were achieved for the same ℓ, the value of ℓ is also given in the Tables.

We note, that the value of $\rho^{(M)}/\rho^{(S)} \geq 1.2$ for all the rates we considered. So transition to multiple thresholds leads to the gain in the decoding radius without affecting the order of complexity. To the best knowledge of the authors the obtained estimates are currently the best estimates of the decoding radius for low-complexity majority decoder of LDPC codes over GF(q).

TABLE 1. Results for $q = 16$

(ℓ, n_0); R	δ	ω^*	$\rho^{(S)}$	$\rho^{(M)}$	$\rho^{(M)}/\rho^{(S)}$
(45, 52); 0.135	0.6130	0.0103	0.0053	0.0065	1.226
(43, 58); 0.26	0.4855	0.0095	0.0049	0.0060	1.224
(40, 64); 0.375	0.3797	0.0085	0.0044	0.0054	1.227
(31, 62); 0.5	0.2808	0.0072	0.0037	0.0046	1.243
(24, 64); 0.625	0.1935	0.0053	0.0028	0.0034	1.214
(24, 96); 0.75	0.1168	0.0033	0.0017	0.0021	1.235
(26, 208);0.875	0.0507	0.0015	0.0008	0.0010	1.250

TABLE 2. Results for $q = 64$

(ℓ, n_0); R	δ	ω^*	$\rho^{(S)}$	$\rho^{(M)}$	$\rho^{(M)}/\rho^{(S)}$
(21, 24); 0.125	0.7355	0.0156	0.0082	0.0099	1.207
(24, 32); 0.25	0.5863	0.0131	0.0068	0.0083	1.221
(20, 32); 0.375	0.4585	0.0104	0.0054	0.0066	1.222
(22, 44); 0.5	0.3445	0.0081	0.0042	0.0052	1.238
(27, 72); 0.625	0.2415	0.0059	0.0031	0.0038	1.226
(24, 96); 0.75	0.1485	0.0037	0.0019	0.0024	1.263
(26, 208); 0.875	0.0661	0.0017	0.0009	0.0011	1.222

6. CONCLUSION

We improved the estimate on the relative decoding radius ρ for the single threshold majority decoder of LDPC codes over GF(q). The majority decoding algorithm with multiple thresholds is suggested. A lower estimate on the decoding radius realized by the new algorithm is derived. The estimate is shown to be at least 1.2 times better than the estimate for a single threshold majority decoder. At the same time analogous to the result from [11] the transition to multiple thresholds does not affect the order of complexity.

All the results are obtained for the case when the component codes are SPC codes over GF(q). The case of more powerful component codes is considered. It is shown that analogous to [6] the transition to generalized LDPC codes does not lead to a gain in the decoding radius.

To the best knowledge of the authors the obtained estimates are currently the best estimates of the decoding radius for low-complexity majority decoder of LDPC codes over GF(q).

ACKNOWLEDGMENTS

The authors would like to thank the Editor, Vitaly Skachek, and the anonymous reviewers for their valuable comments and suggestions, which were helpful in improving the paper.

REFERENCES

[1] N. Alon, Eigenvalues and expanders, *Combinarorica*, **6** (1986), 83–96.
[2] A. Barg and A. Mazumdar, On the number of errors correctable with codes on graphs, *IEEE Trans. Inf. Theory*, **57** (2011), 910–919.
[3] J. Boutros, O. Pothier and G. Zémor, Generalized low density (Tanner) codes, in *Proc. IEEE Int. Conf. Comm.*, Vancouver, 1999, 441–445.

[4] D. Burshtein, On the error correction of regular LDPC codes using the flipping algorithm, *IEEE Trans. Inf. Theory*, **54** (2008), 517–530.
[5] M. C. Davey and D. MacKay, Low-density parity check codes over GF(q), *IEEE Commun. Lett.*, **2** (1998), 165–167.
[6] A. Frolov and V. Zyablov, Asymptotic estimation of the fraction of errors correctable by q-ary LDPC codes, *Probl. Inf. Transm.*, **46** (2010), 142–159.
[7] A. Frolov and V. Zyablov, On the multiple threshold decoding of LDPC codes over GF(q), in *Proc. 2015 IEEE Int. Symp. Inf. Theory*, Hong Kong, 2015, 2673–2677.
[8] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, 1963.
[9] F. Garcia-Ferrero, D. Declercq and J. Valls, Non-binary LDPC decoder based on symbol flipping with multiple votes, *IEEE Commun. Lett.*, **18** (2014), 749–752.
[10] N. Kahale, On the second eigenvalue and linear expansion of regular graphs, in *Proc. IEEE Symp. Found. Comp. Sci.*, 1992, 296–303.
[11] S. Kovalev, Decoding of low-density codes, *Probl. Inf. Transm.*, **27** (1991), 51–56.
[12] M. Lentmaier and K. Zigangirov, On generalized low-density parity-check codes based on Hamming component codes, *IEEE Commun. Lett.*, **3** (1999), 248–250.
[13] A. Lubotzky, R. Phillips and P. Sarnak, Ramanujan graphs, *Combinarorica*, **8** (1988), 261–277.
[14] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1991.
[15] G. A. Margulis, Explicit constructions of concentrators, *Probl. Inform. Transm.*, **9** (1975), 325–332.
[16] J. Massey, *Threshold Decoding*, MIT Press, Cambridge, 1963.
[17] P. Rybin, On the error-correcting capabilities of low-complexity decoded irregular LDPC codes, in *Proc. 2014 IEEE Int. Symp. Inf. Theory*, Honolulu, 2014, 3165–3169.
[18] M. Sipser and D. A. Spielman, Expander codes, *IEEE Trans. Inf. Theory*, **42** (1996), 1710–1722.
[19] H. Song and J. R. Cruz, Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording, *IEEE Transact. Magnetics*, **39** (2003), 1081–1087.
[20] R. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inf. Theory*, **27** (1981), 533–547.
[21] J. Webber, T. Nishimura, T. Ohgane and Y. Ogawa, A study on adaptive thresholds for reduced complexity bit-flip decoding, in *Proc. Int. Conf. Adv. Commun. Techn.*, 2012, 497–501.
[22] J. Webber, T. Nishimura, T. Ohgane and Y. Ogawa, Performance investigation of reduced complexity bit-flipping using variable thresholds and noise perturbation, in *Proc. Int. Conf. Adv. Commun. Techn.*, 2014, 206–2013.
[23] K. Zigangirov, A. Pusane, D. Zigangirov and D. Costello, On the error-correcting capability of LDPC codes, *Probl. Inf. Transm.*, *44* (2008), 214–225.
[24] V. Zyablov, R. Johannesson and M. Lončar, Low-complexity error correction of Hamming-code-based LDPC codes, *Probl. Inf. Trans.*, **45** (2009), 95–109.
[25] V. Zyablov and M. Pinsker, Estimation of the error-correction complexity for Gallager low-density codes, *Probl. Inf. Transm.*, **11** (1975), 18–28.

Received July 2015; revised March 2016.

E-mail address: alexey.frolov@iitp.ru
E-mail address: zyablov@iitp.ru