Low Complexity Algorithm Approaching the ML Decoding of Binary LDPC Codes

Irina E. Bocharova[1,2], **Boris D. Kudryashov**[1], **Vitaly Skachek**[2], **and Yauhen Yakimenka**[2]

[1] Department of Information Systems
St. Petersburg University of Information Technologies,
Mechanics and Optics
St. Petersburg 197101, Russia
Email: { irina, boris }@eit.lth.se

[2] Institute of Computer Science
University of Tartu
Tartu 50409, Estonia
Email: { vitaly, yauhen } @ut.ee

***Abstract*—A novel method for decoding of low-density parity-check codes on the AWGN channel is presented. In the proposed method, first, a standard belief-propagation decoder is applied, then a certain number of positions is erased using a combination of a reliability criterion and a set of masks. A list erasure decoder is then applied to the resulting word. The performance of the proposed method is analyzed mathematically and demonstrated by simulations.**

I. Introduction

Performance of low-density parity-check (LDPC) codes under belief-propagation (BP) decoding is inferior to their performance under the maximum-likelihood (ML) decoding.

There is a variety of approaches for improving both LDPC code constructions and the corresponding decoding techniques. One method to improve the decoding success probability by lowering the error floors is based on detection of trapping sets followed by a post-processing of the corresponding subgraphs in the Tanner graph of the code [1]. In that work, the above approach is applied to regular LDPC codes with high decoding error floors. In [2], trapping sets are taken into account and eliminated when constructing LDPC codes. Both regular and irregular LDPC codes are studied therein. However, neither [1] nor [2] demonstrate any significant performance improvement in the waterfall region.

Near-optimum practical soft decoding techniques for linear block coding are considered in [3]–[5]. The main idea behind these techniques is multiple attempts of finding a "good" (error-free) information set and then reconstructing a codeword by re-encoding.

Reconstructing a codeword in a situation when only a part of the symbol set is known is equivalent to decoding on a binary erasure channel (BEC). Simple decoding algorithms for LDPC codes used for communicating over BEC appeared in [6]. Later, suboptimal erasure decoding algorithms for LDPC codes used over BEC were presented, for example, in [7], [8].

We note that ML decoding of an $[n,k]$ LDPC code (where n is length of the block and k is the number of information symbols) with ν erasures is equivalent to solving a system of linear equations of order ν, that is, can be performed via the Gaussian elimination with time complexity at most $O(\nu^3)$. By taking into account the sparsity of the parity-check matrix of the codes, the complexity can be lowered to approximately $O(\nu^2)$ (see overview and analysis in [9] and the references therein). Practically feasible algorithms with thorough complexity analysis can be found in [10].

The idea to introduce erasures in order to improve performances of BP decoding appears in [11], [12]. In [11], symbol log-likelihood ratios (LLR) after BP decoding are used as a criterion for introducing erasures. In [12], after BP decoding, the values of specially selected symbols are guessed, and the BP decoding is repeated for each guessing attempt.

In this paper, we propose an algorithm, which uses some ideas similar to [11] and [12], but differs significantly both in strategy for introducing erasures and in erasure decoding algorithm. A distinguishing feature of our algorithm is the use of a list decoding for the BEC.

II. List decoding in BEC

Consider a BEC with erasure probability ϵ. Let H be an $r \times n$ parity-check matrix of a binary linear $[n,k,d_{\min}]$ block code, $r = n-k$. We denote by $\boldsymbol{h}_i$ the i-th column of H, $i = 1, 2, \cdots, n$. We use notation H_I for the submatrix of H, whose columns are indexed by the set $I \subseteq \{1,2,...,n\}$. We use $\mathbf{0}$ to denote a zero vector.

ML decoder corrects any pattern of ν erasures if $\nu \le d_{\min} - 1$. If $d_{\min} \le \nu \le n-k$ then a unique correct decision can be obtained for some erasure patterns. The number of such correctable patterns depends on the code structure.

Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ be a received vector, where $y_i \in \{0,1,\phi\}$, and the symbol ϕ represents erasures. We denote by $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)$ a binary vector, such that for all $i = 1, 2, \cdots, n$, $e_i = 1$ if $y_i = \phi$, and $e_i = 0$ if $y_i \in \{0,1\}$. Let $I(\boldsymbol{e})$ be a set of nonzero coordinates of $\boldsymbol{e}$, $|I(\boldsymbol{e})| = \nu(\boldsymbol{e})$, and $\boldsymbol{z} = (z_1, z_2, ..., z_\nu)$ be a vector of unknowns located on positions $I(\boldsymbol{e})$.

Consider a system of linear equations $\boldsymbol{y}H^{\mathrm{T}} = \mathbf{0}$ which can be reduced to

$$\boldsymbol{z}H^{\mathrm{T}}_{I(\boldsymbol{e})} = \boldsymbol{s}(\boldsymbol{e}) \ . \tag{1}$$

where $\boldsymbol{s}(\boldsymbol{e}) = \boldsymbol{y}_{I^{c}(\boldsymbol{e})} H^{\mathrm{T}}_{I^{c}(\boldsymbol{e})}$ is a syndrome vector computed using non-erased positions of $\boldsymbol{y}$ and $I^{c}(\boldsymbol{e}) = \{1,2,\ldots,n\} \setminus I(\boldsymbol{e})$.

The solution of (1) is unique if $\rho(\boldsymbol{e}) \triangleq \mathrm{rank} H_{I(\boldsymbol{e})} \ge \nu$, otherwise the full list $\mathcal{L}$ of candidate solutions contains $T = |\mathcal{L}| = 2^L$ elements, where $L = \nu - \rho(\boldsymbol{e})$.

This work is supported by the Norwegian-Estonian Research Cooperation Programme under the grant EMP133.

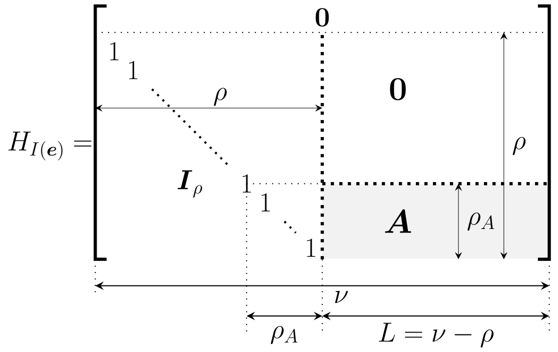

Fig. 1. Structure of the parity-check matrix after diagonalization and reordering of columns

If the code rate R approaches the BEC capacity $C = 1 - \epsilon$, the typical number of erasures $\nu \approx n\epsilon \to n(1-R) = n-k$. In that case, with high probability, the dimension L of the linear space of solutions of (1) is positive.

In general case, by column and row permutations, the submatrix $H_{I(e)}$ can be represented in the form shown in Fig. 1. The coordinates of L columns of the submatrix A determine a set I_{AA} of *arbitrarily assigned* (AA) positions.

Definition 1. *List erasure decoder (LED) is a decoder for the BEC, which for a given input $\boldsymbol{x}$ with ν erased positions, outputs a list $\mathcal{L}$ of codewords $\hat{\mathbf{c}}$ coinciding with the input word on all non-erased coordinates.*

One of possible implementations of the LED is presented as Algorithm 1. In this algorithm we call "pivot" a row chosen to eliminate nonzero elements in other rows on a position which we call "leader".

III. LED-BASED ALGORITHM FOR AWGN CHANNEL

In this section, we show how the LED can be used for decoding of LDPC codes on additive white Gaussian noise (AWGN) channel with input vector $\boldsymbol{v} = (v_1, v_2, ..., v_n)$, $v_i = 2c_i - 1$, and output vector $\boldsymbol{r} = (r_1, r_2, ..., r_n)$. To describe algorithm we introduce the following notations: L_1 is the number of erasures determined by the reliability values computed by the BP decoder, L_2 is the number of additional erasures, $\mathcal{M} = \{\mathcal{M}_i\}_{i=1,...,N}$ is a set of masks, where $\mathcal{M}_i \subset \{1, 2, ..., n\}$ and $|\mathcal{M}_i| = L_2$ for $i = 1, 2, \ldots, N$, $J_{\max} \leq 2^L$ denotes the maximal number of allowed candidate solutions, $\mu(\cdot)$ is a decoding metric.

The following three subroutines are used by Algorithm 2.

- $(\hat{\boldsymbol{v}}, \boldsymbol{x})$=BPDECOD($\boldsymbol{r}$), where $\hat{\boldsymbol{v}}$ and $\boldsymbol{x}$ are vectors of hard decisions and of symbol reliabilities, respectively, produced by the BP decoder. In order to avoid overestimates due to cycles in the Tanner graph, $\boldsymbol{x}$ is computed as the minimum of the absolute values of the symbol reliabilities in the first g_{T} iterations, where g_{T} is a girth of the corresponding Tanner graph.
- $(\boldsymbol{c}, I_{\mathrm{AA}})$=LED($\boldsymbol{\xi}$), where $\boldsymbol{\xi}$ is a vector $\boldsymbol{v}$ with zeros on ν erasured positions, and $\boldsymbol{c}$ is a vector of hard decisions with erasures ϕ on L AA positions. Function LED can be implemented as discussed in Section II.
- $\boldsymbol{c}_j$=CANDIDATE $(\boldsymbol{c}, I_{\mathrm{AA}}, j)$. This subroutine generates the j-th candidate codeword from the full list of solutions of (1). This is done by constructing a list $\mathcal{W}$ of $J_{\max}$ binary words of length $L = |I_{\mathrm{AA}}| = \log_2 |\mathcal{L}|$ ordered according to the ascending order of their weights. Then, the jth candidate is obtained by flipping AA bits in positions determined by the ones of the j-th element in $\mathcal{W}$.

Now, we discuss the strategy for selecting the N masks used in Step 3 of Algorithm 2. These masks are used to erase L_2 symbols in addition to already erased L_1 symbols. This step in the algorithm is similar to the bit flipping step in other information sets based decoding algorithms, such as [4], [5]. The choice of the parameters N and L_2 depends on the code length and the code rate. We found empirically that for the rate $R = 1/2$, for any code length, we can choose $5 \leq N \leq 10$ and $L_2 \approx 0.15n$.

We use pseudo-randomly pre-selected binary sequences of length $2L_2$ and weight L_2 as masks. Masks are applied to the next $2L_2$ less reliable entries, after L_1 positions have already been erased.

We use the Euclidean distance between the channel output $\boldsymbol{r}$ and $2\boldsymbol{c}-1$ as the decoding metric $\mu(\boldsymbol{c})$, where $\boldsymbol{c}$ is a candidate codeword. Alternatively, we can maximize the scalar product of $\boldsymbol{r}$ and $2\boldsymbol{c} - 1$.

As it is shown in Algorithm 2, first, $\nu = L_1 + L_2$ input symbols are erased (see below). Next, LED is used for correcting ρ erasures and producing a list I_{AA} of the AA positions.

IV. ALGORITHM ANALYSIS

There are two main reasons for the algorithm failure. First, an erroneous bit can be mistakenly identified as reliable, in that case it is not erased in Steps 2 and 3 of the algorithm. Second, the list size could be too large, so the correct decision is not found in Step 5 because of the decoding complexity restrictions.

A. Probability of non-erased hard decision errors after BP and masking

Unlike "classical" information set (IS) based decoding techniques [3] on the AWGN channel, symbol reliabilities can be used as a prompt for choosing the IS. Additionally, in ordered statistic decoding and Box-and-Match decoding [4], [5] the low-weight combinations of symbol inversions are employed to catch small error patterns in the IS. In the proposed algorithm, we follow a similar strategy, yet we apply a preliminary BP decoding step in order to improve the decoding efficiency.

In Fig. 2, we demonstrate the distribution of the erroneous hard decisions in the ordered (in descending order of reliabilities) code positions. Only the frames, for which the BP decoding leads to the decoding error, are taken into account. The simulation parameters are: SNR=2.5 dB, $n = 520$, maximum number of iterations 50. As it can be seen from the plot, the preliminary BP decoding step noticeably reduces the

Algorithm 1 LED algorithm for decoding of LDPC code on the BEC channel

```
Input: $\boldsymbol{y} = (y_1, \cdots, y_n) \in \{0, 1, \phi\}^n$.
Initialization:
$I_{\mathrm{AA}} \leftarrow \varnothing$;      $I(\boldsymbol{e}) \leftarrow \{i : y_i = \phi\}$ ;
$\nu \leftarrow |I(\boldsymbol{e})|$;      $\boldsymbol{s} \leftarrow \boldsymbol{y}_{I^c(\boldsymbol{e})} H^{\mathrm{T}}_{I^c(\boldsymbol{e})}$.
Step 1:
while there is a check $j$ with one erased position $y_i$ do
    $y_i \leftarrow s_j$;
    $I(e) \leftarrow I(\boldsymbol{e}) \backslash \{i\}$;
    $\nu \leftarrow \nu - 1$;
    Update $\boldsymbol{s}$;
    if $\nu = 0$ then goto Step 3;
    end if
end while
Step 2:
if there is a check $j$ with erasures not used as pivot then
    Select a check $j$ as a pivot;
    Select $y_i = \phi$ as a leader;
    Gaussian elimination of row j:
    Modify all checks which include position $i$;
    Update $\boldsymbol{s}$;
    goto Step 1.
end if
Step 3: $I_{\mathrm{AA}} \leftarrow I(\boldsymbol{e})$ , $\boldsymbol{c} \leftarrow \boldsymbol{y}$.
Step 4: Return $\boldsymbol{c}$ and $I_{\mathrm{AA}}$ .
```

Algorithm 2 Algorithm for decoding of LDPC code on the AWGN channel

```
Input: the vector of LLRs $\boldsymbol{r} = (r_1, \ldots, r_n) \in \mathbb{R}^n$.
Let $\mu_{\mathrm{opt}} \leftarrow \infty$.
Step 1:
$(\hat{\boldsymbol{v}}, \boldsymbol{x}) = \textsc{BPDecod}(\boldsymbol{r})$; $\hat{\boldsymbol{c}} = (\hat{\boldsymbol{v}} + 1)/2$;
if $\hat{\boldsymbol{c}} H^{\mathrm{T}} = \mathbf{0}$ then goto Step 6;
end if
Step 2:
$\boldsymbol{\xi} \leftarrow \hat{\boldsymbol{v}}$ with zeros on $L_1$ least reliable positions in $\boldsymbol{x}$.
for $i = 1$ to $N$ do
    Step 3:
    Use mask $\mathcal{M}_i$ to erase $L_2$ non-erased positions in $\boldsymbol{\xi}$.
    Step 4:
    $(\boldsymbol{c}, I_{\mathrm{AA}}) \leftarrow \mathrm{LED}(\boldsymbol{\xi})$;
    Initialize AA positions in $\boldsymbol{c}_0$ by hard decisions from $\boldsymbol{r}$.
    Step 5:
    for $j = 1$ to $J_{\max}$ do
        Compute codeword $\boldsymbol{c}'$ from $\boldsymbol{c}$ and $\boldsymbol{c}_{j-1}$;
        if $\mu(\boldsymbol{c}') < \mu_{\mathrm{opt}}$ then let $\hat{\boldsymbol{c}} \leftarrow \boldsymbol{c}'$; $\mu_{\mathrm{opt}} \leftarrow \mu(\boldsymbol{c}')$;
        end if
      Generate next candidate $\boldsymbol{c}_j = \textsc{Candidate}(\boldsymbol{c}, I_{\mathrm{AA}}, j)$;
    end for
end for
Step 6: Return $\hat{\boldsymbol{c}}$.
```

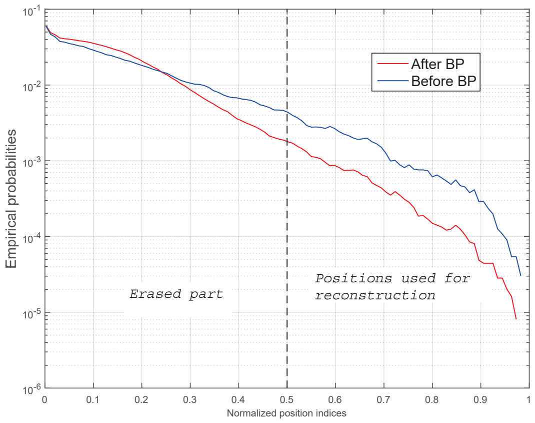

Fig. 2. Locations of hard decision errors in list of ascending reliability ordered positions

probability of the hard decision errors within the 50% of the most reliable positions (for code rate $R = 1/2$), thus leading to a lower overall decoding error probability.

B. Probability of large list size after LED

For simplicity, hereafter we consider an ensemble of random (J, K)-regular LDPC codes equivalent to the Gallager ensemble [13]. In the considered ensemble a parity-check matrix H consists of the strips of width $M = r/J$ rows each. All strips are random column permutations of the strip where the ith row contains K ones in positions from $(i-1)K + 1$ up to iK, $i = 1, 2, \cdots, n/K$.

A set of solutions of $\boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})} = \mathbf{0}$ represents a coset of solutions of (1), that is, has the same number of solutions as (1). We denote this number by T. Note that the solution will be unique if $H_{I(\boldsymbol{e})}$ has rank $\rho = \nu$.

Introduce a random variable

$$\chi(\boldsymbol{z}) = \begin{cases} 1, & \boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})} = \mathbf{0} \\ 0, & \boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})} \neq \mathbf{0} \end{cases} .$$

Then $T = \sum_{\boldsymbol{z}} \chi(\boldsymbol{z})$ and

$$\mathbf{E}[T] = \sum_{\boldsymbol{z}} \mathbf{E}[\chi(\boldsymbol{z})] = \sum_{\boldsymbol{z}} \Pr(\boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})} = \mathbf{0}) . \tag{2}$$

The RHS of (2) can represented in the form

$$\begin{aligned} \mathbf{E}[T] &= 2^{\nu} \cdot \sum_{\boldsymbol{z}} \Pr(\boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})} = \mathbf{0}) \frac{1}{2^{\nu}} \\ &= 2^{\nu} \cdot \sum_{\boldsymbol{z}} \Pr(\boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})} = \mathbf{0}) p(\boldsymbol{z}) \end{aligned}$$

assuming the vector $\boldsymbol{z}$ is chosen uniformly at random from $\{0, 1\}^{\nu}$. Rank of $H_{I(\boldsymbol{e})}$ can be reduced: (i) due to all-zero rows in $H_{I(\boldsymbol{e})}$ and (ii) due to linearly-dependent columns in the sparse matrix $H_{I(\boldsymbol{e})}$. Consider a random vector $\boldsymbol{s} = \boldsymbol{z} H^{\mathrm{T}}_{I(\boldsymbol{e})}$. Denote by $\boldsymbol{s}_i^j$ a subvector $(s_i, ..., s_j)$. The probability of the all-zero subvector of $\boldsymbol{s}$ corresponding to one-strip part of the parity-check matrix is

$$p(\boldsymbol{s}_1^M = \mathbf{0}) = p(s_1 = 0) \prod_{i=2}^{M} p(s_i = 0 | \boldsymbol{s}_1^{i-1} = \mathbf{0}) ,$$

TABLE I
EXAMPLE OF CRITICAL VALUES OF $\alpha = \nu/r$

Rate	1/5	1/4	1/2	5/8	3/4	
(J, K)	(4,5)	(3,4)	(4,8)	(3,8)	(3,12)	(4,16)
α	0.9995	0.994	0.994	0.975	0.944	0.984

where

$$\begin{aligned} & p(s_i = 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) \\ = \quad & p(s_i = 0|w_i = 0, \boldsymbol{s}_1^{i-1} = \mathbf{0})p(w_i = 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) \\ + \quad & p(s_i = 0|w_i > 0, \boldsymbol{s}_1^{i-1} = \mathbf{0})p(w_i > 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) \\ \overset{(a)}{=} \quad & p(w_i = 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) + \frac{1}{2}p(w_i > 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) \\ = \quad & \frac{1}{2}\left(1 + p(w_i = 0|\boldsymbol{s}_1^{i-1} = \mathbf{0})\right), \end{aligned}$$

and $p(w_i = w|\boldsymbol{s}_1^{i-1} = \mathbf{0})$ denotes the conditional probability that the ith row of $H_{I(\boldsymbol{e})}$ has weight w given that the syndrome components $(s_1, s_2, \ldots, s_{i-1})$ are zeros. In order to obtain (a) first we substitute $p(s_i = 0|w_i = 0, \boldsymbol{s}_1^{i-1} = \mathbf{0}) = 1$ and additionally we take into account that that a projection of the random vector $\boldsymbol{z}$ on a nonzero parity check of $H_{I(\boldsymbol{e})}$ is equal to zero with probability 1/2. Notice that

$$p(w_i = 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) = p(w_i = 0)\frac{p(\boldsymbol{s}_1^{i-1} = \mathbf{0}|w_i = 0)}{p(\boldsymbol{s}_1^{i-1} = \mathbf{0})}.$$

Since the fraction in RHS of the latter equation is upper-bounded by 1 then (we skip the details due to the space limitations)

$$p(s_i = 0|\boldsymbol{s}_1^{i-1} = \mathbf{0}) \leq \frac{1}{2}\left(1 + p(w_i = 0)\right) .$$

Since the strips are obtained by the independent random permutations, we have

$$p(\boldsymbol{s} = \mathbf{0}) = p(\boldsymbol{s}_1^M = \mathbf{0})^J \leq p(s_1 = 0)^{MJ} = p(s_1 = 0)^r .$$

The probability that the row i in $H_{I(\boldsymbol{e})}$ has only zeros can be bounded from above by

$$p(w_i = 0) = \frac{\binom{n-\nu}{K}}{\binom{n}{K}} \leq \left(\frac{n-\nu}{n}\right)^K . \tag{3}$$

The probability that the entire sequence of length ν is a codeword (all r components of the syndrome are equal to zero) is

$$p(\boldsymbol{s} = \mathbf{0}) \leq 2^{-r}\left(1 + \left(1 - \frac{\nu}{n}\right)^K\right)^r ; . \tag{4}$$

Denote by $\alpha = \nu/r$ the normalized number of erasures. Asymptotic exponent of list size is determined by

$$\begin{aligned} \varphi(\alpha, J, K) \quad &= \quad \lim_{r\to\infty}\frac{\mathbf{E}[L]}{r} \leq \lim_{r\to\infty}\frac{\log_2 \mathbf{E}[T]}{r} \\ &\leq \quad \alpha - 1 + \log_2\left(1 + \left(1 - \alpha\frac{J}{K}\right)^K\right) . \end{aligned} \tag{5}$$

It is interesting to find a critical (largest) value of α, such that $\varphi(\alpha, J, K) = 0$.

We expect that for sparse matrices $\alpha < 1$. Examples of critical values of α for different code rates $R = 1 - J/K$ and K are given in Table I. We can see that α is close to 1 even for rather sparse parity-check matrices. It means that the allowable fraction of erasures ν/n can be chosen close to $1 - R$.

V. SIMULATION RESULTS

All simulated parity-check matrices were obtained from the same matrix of the irregular LDPC convolutional code used in [14] as a competitor for the standard code from the WiMAX standard. To facilitate low complexity encoding, the degree matrix of the LDPC code [14] has the form

$$D = \begin{pmatrix} D_{\mathrm{bd}} & D_{\mathrm{I}} \end{pmatrix} , \tag{6}$$

where the submatrix D_{bd} of size 12×11 is

$$D_{\mathrm{bd}} = \begin{pmatrix} 0 & 0 & -1 & \ldots & -1 & -1 \\ -1 & 0 & 0 & \ldots & -1 & -1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ -1 & -1 & -1 & \ldots & 0 & 0 \end{pmatrix} \tag{7}$$

and

$$D_{\mathrm{I}} = \begin{pmatrix} 0 & -1 & -1 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & 16 & 0 & 5 & -1 \\ -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 12 & 1 & -1 & 7 \\ -1 & -1 & -1 & -1 & 6 & -1 & -1 & 0 & -1 & 15 & 7 & 11 & -1 \\ 13 & -1 & -1 & -1 & -1 & -1 & -1 & 11 & 22 & -1 & 1 & 21 & 4 \\ -1 & -1 & 8 & -1 & -1 & -1 & 21 & -1 & -1 & 12 & 14 & -1 & 19 \\ -1 & -1 & -1 & 0 & -1 & 10 & -1 & -1 & 15 & 21 & -1 & 12 & 3 \\ -1 & 10 & -1 & -1 & -1 & -1 & -1 & 18 & -1 & 5 & 14 & 21 & 23 \\ -1 & -1 & -1 & 3 & -1 & -1 & 8 & -1 & -1 & 18 & 23 & 16 & 11 \\ -1 & -1 & 14 & -1 & 23 & -1 & -1 & -1 & 20 & -1 & 11 & 22 & 7 \\ 0 & -1 & -1 & -1 & -1 & 19 & -1 & -1 & -1 & 18 & 6 & 5 & 22 \\ -1 & 20 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 5 & 11 & 23 & 19 \end{pmatrix} .$$

In Figs. 3 and 4, the FER performance of the BP decoding with 50 iterations and the LED-based decoding for $R = 1/2$ codes of length $n = 288$, 576, 864 and 1152 are shown. For comparison, in Fig. 4 the FER performance of the BP decoding of the WiMAX standard code of length $n = 576$ is presented. The parameters in all simulations were the same: the total number of erased symbols $\alpha n = 0.5n$. Among them, $(\alpha - \beta)n = 0.35n$ positions were selected according to the reliabilities, estimated by the BP decoding, and $\beta n = 0.15n$ positions were selected pseudo-randomly from the next $2\beta = 0.3n$ less reliable positions. The number of trials was chosen to be equal $N = 5$. All the simulations were run until at least 50 LED-decoding block errors occurred.

The abbreviations LED-8 and LED-16 denote decoding with the list size $L_{\max} = 2^8$ and $L_{\max} = 2^{16}$ in Step 5 of Algorithm 2, respectively. As expected, LED-16 yields lower FER than LED-8, yet the gain is not large enough to justify much higher decoding complexity.

From the presented results, we can conclude that the energy gain grows with the SNRb. In all four cases the gain from using the LED at FER $\approx 10^{-5}$ is about 0.5 dB.

The estimated decoding time is shown in Fig. 5. Execution time was measured at SNRb=2.25 dB for Step 4 and Step 5 in Algorithm 2 separately. The simulations were performed on desktop computer with Intel Core i5 processor. Only the blocks, where the BP decoding has failed, were taken into account.

By the dashed lines we show the polynomial approximations

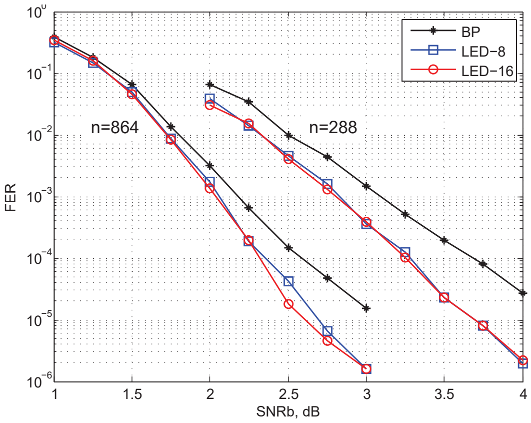

Fig. 3. FER performance of codes of rate $R = 1/2$ and length $n = 288$ and $n = 864$.

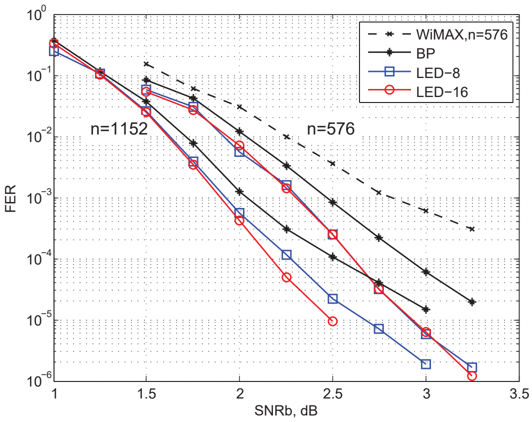

Fig. 4. FER performance of codes of rate $R = 1/2$ and length $n = 576$ and $n = 1152$.

$$\kappa_4 = 1.07 \times 10^{-6} n^2 + 4.27 \times 10^{-4} n - 0.00385\,, \qquad (8)$$

$$\kappa_5 = 4.73 \times 10^{-6} n^2 + 6.74 \times 10^{-3} n - 0.0674\,, \qquad (9)$$

for computation time of Steps 4 and 5, respectively.

As it was mentioned in the Introduction, the expected decoding complexity for Step 4 is a cubic function of the code length n. Nevertheless, the observed average decoding time grows approximately linearly with code length. Approximation (8) shows that contribution of the quadratic term is negligible.

By contrast, the linear complexity of Step 5 for a fixed number of decoding attempts can be readily explained. Complexity of each codeword reconstruction and computing its metric is proportional to the list dimension L (see Fig.1).

Although complexity grows approximately linearly, the algorithm loses efficiency for large n since a typical number L of AA positions grows as well. To maintain the decoding efficiency, parameters N and $L_{\max}$ should also be increased, which leads to impractically high computational complexity for length above 2000.

VI. Conclusions

In this paper, we presented a novel decoding algorithm for LDPC codes used in conjunction with the AWGN channel. This algorithm consists of a standard BP decoder followed by list erasure decoder.

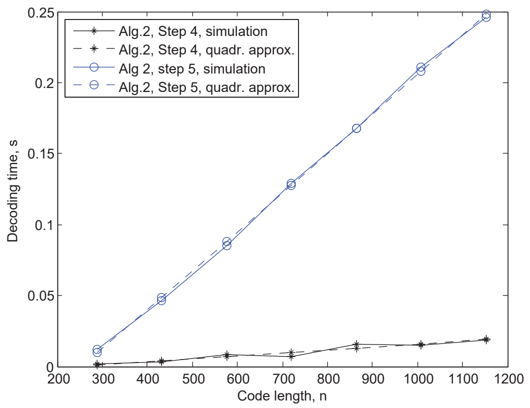

Fig. 5. Complexities of Steps 4 (LED) and 5 (search for the best candidate from the list) and their polynomial approximations.

We simulated the proposed algorithm for the irregular LDPC codes of rate $R = 1/2$ optimized for BP decoding. It is shown that the new algorithm outperforms the standard BP decoding for the chosen codes. We observed that the growth of the average decoding complexity of the most heavy steps in the algorithm is almost linear with the code length.

References

[1] Y. Han and W. E. Ryan, "Low-floor decoders for LDPC codes," *IEEE Trans. Comm.*, vol. 57, no. 6, pp. 1663–1673, 2009.

[2] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, "Lowering the error floor of LDPC codes using cyclic liftings," *IEEE Trans. Inform. Theory*, vol. 57, no. 4, pp. 2213–2224, 2011.

[3] E. Prange, "The use of information sets in decoding cyclic codes," *IRE Trans. Inform. Theory*, vol. 8, no. 5, pp. 5–9, 1962.

[4] A. Valembois and M. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. Inform. Theory*, vol. 50, no. 5, pp. 796–810, 2004.

[5] M. P. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[6] V. V. Zyablov and M. S. Pinsker, "Decoding complexity of low-density codes for transmission in a channel with erasures," *Problemy Peredachi Informatsii*, vol. 10, no. 1, pp. 15–28, 1974.

[7] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 3, pp. 439–454, 2004.

[8] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Tree-structure expectation propagation for decoding LDPC codes over binary erasure channels," in *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 799–803.

[9] D. Burshtein and G. Miller, "An efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2837–2844, 2004.

[10] E. Paolini, G. Liva, B. Matuz, and M. Chiani, "Maximum likelihood erasure decoding of LDPC codes: Pivoting algorithms and code design," *IEEE Trans. Comm.*, vol. 60, no. 11, pp. 3209–3220, 2012.

[11] Y. Fang, J. Zhang, L. Wang, and F. Lau, "BP-Maxwell decoding algorithm for LDPC codes over AWGN channels," in *6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2010, pp. 1–4.

[12] H. Pishro-Nik and F. Fekri, "Results on punctured low-density parity-check codes and improved iterative decoding techniques," *IEEE Trans. Inform. Theory*, vol. 53, no. 2, pp. 599–614, 2007.

[13] R. G. Gallager, *Low-density parity-check codes*. M.I.T. Press: Cambridge, MA, 1963.

[14] I. Bocharova, B. Kudryashov, and R. Johannesson, "Searching for binary and nonbinary block and convolutional LDPC codes," *IEEE Trans. Inform. Theory*, vol. 62, no. 1, pp. 163–183, Jan. 2016.