

LETTER

An Investigation on LP Decoding of Short Binary Linear Codes With the Subgradient Method*

Haiyang LIU^{†a)}, Xiaopeng JIAO^{††b)}, *Nonmembers*, and Lianrong MA^{†††c)}, *Member*

SUMMARY In this letter, we investigate the application of the subgradient method to design efficient algorithm for linear programming (LP) decoding of binary linear codes. A major drawback of the original formulation of LP decoding is that the description complexity of the feasible region is exponential in the check node degrees of the code. In order to tackle the problem, we propose a processing technique for LP decoding with the subgradient method, whose complexity is linear in the check node degrees. Consequently, a message-passing type decoding algorithm can be obtained, whose per-iteration complexity is extremely low. Moreover, if the algorithm converges to a valid codeword, it is guaranteed to be a maximum likelihood codeword. Simulation results on several binary linear codes with short lengths suggest that the performances of LP decoding based on the subgradient method and the state-of-art LP decoding implementation approach are comparable.

key words: *binary linear codes, linear programming (LP) decoding, subgradient method*

1. Introduction

As a relaxation of maximum likelihood (ML) decoding, linear programming (LP) decoding [1] of linear codes is a promising technique that has satisfactory theoretical guarantee on decoding performance. However, solving the problem in LP decoding through general-purpose LP solvers is infeasible, especially for codes with high check node degrees. In order to address the problem, the implementation of LP decoding through the iterative alternate direction method of multipliers (ADMM) technique has been proposed in the recent years [2]–[11]. However, these implementation methods still involve complicated operations (e.g., the polytope projection).

The subgradient method [12] is an iterative algorithm for solving nondifferentiable optimization problems. In his seminal work [1], Feldman has introduced the idea of solving the LP decoding problem using the subgradient method.

However, with the exception of [13], this line of works do not seem to have been further considered to the best of our knowledge. (In fact, the authors in [13] have presented a subgradient-based approach for decoding a particular class of codes, i.e., single parity-check product codes.) From the optimization point of view, the subgradient method has the following advantages, which makes it attractive in LP decoding. First, compared with other optimization methods (e.g., interior-point method), the storage requirement of the subgradient method is much smaller. Besides, the operations involved in the method are very simple. Second, LP decoding with the subgradient method outputs a binary vector in each iteration. Hence, the decoding output for each bit is more interpretable compared with solving the LP decoding problem through general-purpose LP solvers.

In this letter, we focus on short binary linear codes, which are amicable in low-latency wireless communication systems where short packets are preferred [14]. Despite the importance, how to develop efficient soft-decision decoding algorithms that can achieve ML or near-ML performances for these codes is challenging. Although their lengths are short, the exact ML decoding is intractable for these codes in general. For some families of algebraic codes, we can design soft-decision decoding algorithms (e.g., Chase algorithm) whose performances are close to ML decoding [15]. However, these approaches do not work well for non-algebraic codes such as random low-density parity-check (LDPC) codes. Since the performance of LP decoding is close to ML decoding when the channel noise is low [1] and the subgradient method has advantages mentioned above, it is deserved to investigate the design of efficient LP decoding algorithm based on the subgradient method for short binary linear codes, which motivates our work.

It is known that the description complexity of the feasible region of the original LP decoding problem is exponential in the check node degrees of the code [1]. In order to address the problem, we develop a processing method whose complexity is linear in the check node degree for LP decoding with the subgradient method. The obtained decoding approach has lower per-iteration complexity compared with the ADMM-based LP decoding. Moreover, if the algorithm converges to a codeword, it is an ML codeword. Simulation results on several short binary codes suggest that the error rate performance of the proposed method is close to that of the ADMM-based LP decoding. In addition, the proposed method also has advantages in the average decoding time of a codeword for the tested codes, which suggests that

Manuscript received May 30, 2023.

Manuscript revised September 27, 2023.

Manuscript publicized November 21, 2023.

[†]Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China.

^{††}School of Computer Science and Technology, Xidian University, Xi'an 710071, China.

^{†††}Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.

*This work is supported by the National Natural Science Foundation of China (NSFC) under Grants 62271482, 61971322, and 61871376.

a) E-mail: liuhaiyang@ime.ac.cn

b) E-mail: jiaozi1216@126.com

c) E-mail: malian@tsinghua.edu.cn

DOI: 10.1587/transfun.2023EAL2045

the method is suitable for applications where low decoding latency is required.

2. Background

In this section, we introduce the background knowledge of our work. For any vector $\mathbf{a} = [a_i]$, the set $\{i : a_i \neq 0\}$ is called the *support* of \mathbf{a} . For a given set \mathcal{S} , denote $|\mathcal{S}|$ by the cardinality of \mathcal{S} .

An $[n, k]$ binary linear code C can be specified as the null space of a parity-check matrix $\mathbf{H} = [h_{ji}]$ of size $m \times n$ over $\mathbb{F}_2 = \{0, 1\}$, where $m \geq n - k$, with equality if and only if \mathbf{H} is full-rank. Suppose \mathcal{I} and \mathcal{J} are the indices for the columns and rows of \mathbf{H} . Define $\mathcal{M}(i) = \{j \in \mathcal{J} : h_{ji} = 1\}$ and $\mathcal{N}(j) = \{i \in \mathcal{I} : h_{ji} = 1\}$. In particular, if $|\mathcal{M}(i)| = d_v$ (resp., $|\mathcal{N}(j)| = d_c$) holds for each $i \in \mathcal{M}(i)$ (resp., $j \in \mathcal{N}(j)$), the code is said to be (d_v, d_c) -regular. For convenience, we associate \mathbf{H} with a bipartite graph $\mathcal{G}(\mathbf{H})$, called Tanner graph [15], which consists of two sets of nodes: The set of n bit nodes that correspond to the indices in \mathcal{I} and the set of m check nodes that correspond to the indices in \mathcal{J} . There is an edge $(i, j) \subseteq \mathcal{I} \times \mathcal{J}$ in $\mathcal{G}(\mathbf{H})$ connecting the i -th bit node and the j -th check node if and only if $h_{ji} = 1$.

Suppose a codeword $\mathbf{x} \in C$ is transmitted over a memoryless binary-input output-symmetric channel and \mathbf{r} is the received vector. It can be shown that the ML codeword $\hat{\mathbf{x}}_{\text{ML}}$ is the solution of the following minimization problem:

$$\min_{\mathbf{x} \in \text{conv}(C)} \boldsymbol{\gamma}^T \mathbf{x}, \quad (1)$$

where $\text{conv}(C)$ is the convex hull of C when C is embedded in \mathbb{R}^n and $\boldsymbol{\gamma}$ is the log-likelihood ratio vector whose i -th entry is $\gamma_i = \log \left[\frac{\Pr(r_i | x_i=0)}{\Pr(r_i | x_i=1)} \right]$ ($i \in \mathcal{I}$).

Due to the daunting complexity of describing $\text{conv}(C)$, the problem in (1) is practically intractable. Feldman has proposed an approximation of the problem [1]. Suppose C_j is the single parity-check code defined by the j -th row of \mathbf{H} . Define the polytope $\mathcal{P} = \bigcap_{j \in \mathcal{J}} \text{conv}(C_j)$ and construct the minimization problem:

$$\min_{\mathbf{x} \in \mathcal{P}} \boldsymbol{\gamma}^T \mathbf{x}. \quad (2)$$

Then LP decoding is to find the optimal solution $\hat{\mathbf{x}}_{\text{LP}}$ of the minimization problem in (2).

Every codeword of C corresponds to one vertex of \mathcal{P} . Conversely, not all vertices of \mathcal{P} are codewords of C in general, which is due to the fact that \mathcal{P} could have non-integral vertices. The optimum of an LP problem can always be reached at one vertex of the polytope \mathcal{P} . When the solution $\hat{\mathbf{x}}_{\text{LP}}$ is at an integer vertex, it must be the optimal solution of the problem in (1) and thus is an ML codeword. In contrast, if $\hat{\mathbf{x}}_{\text{LP}}$ is non-integral, then LP decoder detects a decoding error. We call this fact the *ML certificate property* of an LP decoder, which makes the performance of the decoder amenable to theoretical analysis [1].

3. LP Decoding with the Subgradient Method

3.1 Formulation

Now we review LP decoding with the subgradient method, which has been originally presented in [1]. The polytope \mathcal{P} can be reformulated by introducing auxiliary variables. Define

$$\mathcal{E}_j = \{\mathcal{S} \subseteq \mathcal{N}(j) : |\mathcal{S}| \text{ is even}\} \quad (3)$$

for each $j \in \mathcal{J}$. Note that $\emptyset \in \mathcal{E}_j$.

By inspection, we obtain that $|\mathcal{E}_j| = 2^{|\mathcal{N}(j)|-1}$. An auxiliary variable $w_{j,\mathcal{S}}$ is associated to each \mathcal{S} in \mathcal{E}_j , which is an indicator for the codeword in C_j corresponding to \mathcal{S} . Then $\text{conv}(C_j)$ can be represented using the following constraints:

$$x_i = \sum_{\mathcal{S} \in \mathcal{E}_j, \mathcal{S} \ni i} w_{j,\mathcal{S}}, \forall i \in \mathcal{N}(j), \quad (4)$$

$$0 \leq x_i \leq 1, \forall i \in \mathcal{I}, \quad (5)$$

$$\sum_{\mathcal{S} \in \mathcal{E}_j} w_{j,\mathcal{S}} = 1 \text{ and } w_{j,\mathcal{S}} \geq 0, \forall \mathcal{S} \in \mathcal{E}_j. \quad (6)$$

Using the Lagrange multiplier method [12], we construct

$$\mathcal{L}(\mathbf{m}) = \boldsymbol{\gamma}^T \mathbf{x} + \sum_{i,j} m_{ij} \left[\sum_{\mathcal{S} \in \mathcal{E}_j, \mathcal{S} \ni i} w_{j,\mathcal{S}} - x_i \right] \quad (7)$$

by moving the equality constraints in (4) into the objective function, where $\mathbf{m} = [m_{ij}]$, and the coefficients m_{ij} are Lagrange multipliers. Denote Q by the set of points (\mathbf{x}, \mathbf{w}) satisfying the constraints in (5) and (6). Note that the projection of Q onto \mathbf{x} is the polytope \mathcal{P} . Define

$$\mathcal{L}^*(\mathbf{m}) = \min_{(\mathbf{x}, \mathbf{w}) \in Q} \mathcal{L}(\mathbf{m}). \quad (8)$$

From the duality theory [12], we have

$$\min_{(\mathbf{x}, \mathbf{w}) \in Q} \boldsymbol{\gamma}^T \mathbf{x} = \max_{\mathbf{m}} \mathcal{L}^*(\mathbf{m}) \quad (9)$$

The problem $\max_{\mathbf{m}} \mathcal{L}^*(\mathbf{m})$ can be solved in an iterative manner by using the subgradient method, i.e., a sequence $\{\mathbf{m}^{(k)}\}$ can be generated, where k is the iteration number. The iterative process can be treated as a message passing process, and the Lagrange multiplier $m_{ij}^{(k)}$ can be viewed as the message from the i -th bit node to the j -th check node in the k -th iteration. This can facilitate the implementation of the decoding process. In general, we perform the following steps until a prescribed maximum number K_{max} of iterations is reached [1].

Step 1: Check message processing.

Calculate $m_{ji}^{(k)}$, the message from the j -th check node to the i -th bit node in the k -th iteration, by

$$m_{ji}^{(k)} = \begin{cases} 1, & i \in \mathcal{S}_j^-, \\ 0, & i \notin \mathcal{S}_j^-, \end{cases} \quad (10)$$

where

$$\mathcal{S}_j^- = \operatorname{argmin}_{\mathcal{S} \in \mathcal{E}_j} \sum_{i \in \mathcal{S}} m_{ij}^{(k)}. \quad (11)$$

Step 2: Bit message processing.

Estimate the transmitted codeword $\mathbf{y}^{(k)} = [y_i^{(k)}]$ according to

$$y_i^{(k)} = \begin{cases} 1, & \gamma_i - \sum_{j \in \mathcal{M}(i)} m_{ij}^{(k)} < 0, \\ 0, & \gamma_i - \sum_{j \in \mathcal{M}(i)} m_{ij}^{(k)} \geq 0. \end{cases} \quad (12)$$

Update $m_{ij}^{(k)}$ for each edge (i, j) such that $y_i^{(k)} \neq m_{ji}^{(k)}$ using the following equation:

$$m_{ij}^{(k+1)} = \begin{cases} m_{ij}^{(k)} + \alpha^{(k)}, & y_i^{(k)} = 0, \\ m_{ij}^{(k)} - \alpha^{(k)}, & y_i^{(k)} = 1, \end{cases} \quad (13)$$

where $\{\alpha^{(k)}\}$ is a sequence of prescribed positive real numbers satisfying $\lim_{k \rightarrow \infty} \alpha^{(k)} = 0$ and $\sum_{k=1}^{\infty} \alpha^{(k)} = \infty$ [12].

Step 3: Termination decision.

If $y_i^{(k)} = m_{ji}^{(k)}$ for all the edges (i, j) , then terminate the iterative process and output $\mathbf{y}^{(k)}$.

It is known from the above description that the algorithm outputs a binary vector in each iteration. Moreover, if the output of LP decoder (2) is integral, $\mathbf{y}^{(k)}$ is the same output for sufficiently large k , in which the algorithm converges to an ML codeword [1].

3.2 Proposed Check Message Processing Procedure

From the previously described algorithm, we know that the most complicated procedure is to find the set \mathcal{S}_j^- in (11), whose complexity is exponential in $|\mathcal{N}(j)|$ using traditional methods since we have $|\mathcal{E}_j| = 2^{|\mathcal{N}(j)|-1}$. This is particularly troublesome for check nodes with high degrees. In the following, we develop a low-complexity procedure (Algorithm 1) for finding \mathcal{S}_j^- . At first our algorithm constructs the set $\mathcal{I}_j^- = \{i \in \mathcal{N}(j) : m_{ij}^{(k)} < 0\}$. If the size of \mathcal{I}_j^- is even, we have $\mathcal{S}_j^- = \mathcal{I}_j^-$ and the algorithm terminates. If the size of \mathcal{I}_j^- is odd, we compare $|m_{i_p j}^{(k)}|$ and $|m_{i_n j}^{(k)}|$ and let $\mathcal{S}_j^- = \mathcal{I}_j^- \setminus \{i_n\}$ if $|m_{i_p j}^{(k)}| > |m_{i_n j}^{(k)}|$ and $\mathcal{S}_j^- = \mathcal{I}_j^- \cup \{i_p\}$ otherwise, where $m_{i_p j}^{(k)}$ (resp., $m_{i_n j}^{(k)}$) has the minimum absolute value among all $m_{ij}^{(k)}$ that are positive (resp., negative). The steps of our procedure are provided in Algorithm 1, whose correctness is proved in Proposition 1.

Proposition 1: The output of Algorithm 1 is \mathcal{S}_j^- given in (11).

Proof: Let \mathcal{T} be a subset of $\mathcal{N}(j)$. Assume $\mathcal{I}_j^- = \{i \in \mathcal{N}(j) : m_{ij}^{(k)} < 0\}$ and $\mathcal{I}_j^+ = \mathcal{N}(j) \setminus \mathcal{I}_j^-$. By inspection, we know that $\sum_{i \in \mathcal{T}} m_{ij}^{(k)}$ achieves the minimum when $\mathcal{T} = \mathcal{I}_j^-$. We distinguish between two cases.

Case 1: $|\mathcal{I}_j^-|$ is even. In this case, we have $\mathcal{I}_j^- \in \mathcal{E}_j$.

Algorithm 1 Procedure of finding \mathcal{S}_j^- .

```

1: Initialize  $m_p = +\infty$ ,  $m_n = +\infty$ , and  $\mathcal{S}_j^- = \emptyset$ .
2: for  $i \in \mathcal{N}(j)$  do
3:   if  $m_{ij}^{(k)} < 0$  then
4:      $\mathcal{S}_j^- \leftarrow \mathcal{S}_j^- \cup \{i\}$ .
5:     if  $|m_{ij}^{(k)}| < m_n$  then
6:        $m_n \leftarrow |m_{ij}^{(k)}|$ ,  $i_n \leftarrow i$ .
7:     end if
8:   else
9:     if  $m_{ij}^{(k)} < m_p$  then
10:       $m_p \leftarrow m_{ij}^{(k)}$ ,  $i_p \leftarrow i$ .
11:    end if
12:   end if
13: end for
14: if  $|\mathcal{S}_j^-|$  is odd then
15:   if  $m_p > m_n$  then
16:      $\mathcal{S}_j^- \leftarrow \mathcal{S}_j^- \setminus \{i_n\}$ .
17:   else
18:      $\mathcal{S}_j^- \leftarrow \mathcal{S}_j^- \cup \{i_p\}$ .
19:   end if
20: end if
21: Output  $\mathcal{S}_j^-$ .

```

Hence, $\mathcal{S}_j^- = \mathcal{I}_j^-$.

Case 2: $|\mathcal{I}_j^-|$ is odd. Let $i_n = \operatorname{argmin}_{i \in \mathcal{I}_j^-} |m_{ij}^{(k)}|$ and $i_p = \operatorname{argmin}_{i \in \mathcal{I}_j^+} |m_{ij}^{(k)}|$. In this case, the sets $\mathcal{I}_j^- \setminus \{i_n\}$ and $\mathcal{I}_j^- \cup \{i_p\}$ are both in \mathcal{E}_j . If $\sum_{i \in \mathcal{I}_j^-} m_{ij}^{(k)} - m_{i_n j}^{(k)} < \sum_{i \in \mathcal{I}_j^-} m_{ij}^{(k)} + m_{i_p j}^{(k)}$, the sum in (11) achieves the minimum $\sum_{i \in \mathcal{I}_j^-} m_{ij}^{(k)} - m_{i_n j}^{(k)}$ and $\mathcal{S}_j^- = \mathcal{I}_j^- \setminus \{i_n\}$. Otherwise, the sum in (11) achieves the minimum $\sum_{i \in \mathcal{I}_j^-} m_{ij}^{(k)} + m_{i_p j}^{(k)}$ and $\mathcal{S}_j^- = \mathcal{I}_j^- \cup \{i_p\}$. \square

We can check that our Algorithm 1 needs at most $2|\mathcal{N}(j)| + 1$ comparisons to find \mathcal{S}_j^- for a check node j of degree $|\mathcal{N}(j)|$. Hence, the complexity of the procedure is linear in the check node degree. Moreover, the addition operations are avoided in the check message processing.

3.3 Complexity Analysis

In this subsection, the per-iteration complexity of LP decoding with the subgradient method is analyzed for a (d_v, d_c) -regular code. Note that we have $md_c = nd_v$ in this case. As previously discussed, at most $m(2d_c + 1)$ real comparisons are needed to find \mathcal{S}_j^- given by (11) in one iteration. For bit message processing, we need nd_v real additions to calculate $\mathbf{y}^{(k)}$ in (12) in one iteration. Then we need nd_v integer comparisons and at most nd_v real additions to update the Lagrange multiplier $m_{ij}^{(k)}$ as well as the termination decision in one iteration. Hence, the per-iteration complexity of the proposed method in the worst case is linear in d_c and n if d_v and d_c are independent of n .

The ADMM approach is also an iterative method for the implementation of LP decoding. The most complicated part of the ADMM-based LP decoding in each iteration is the polytope projection, which usually involves a sorting

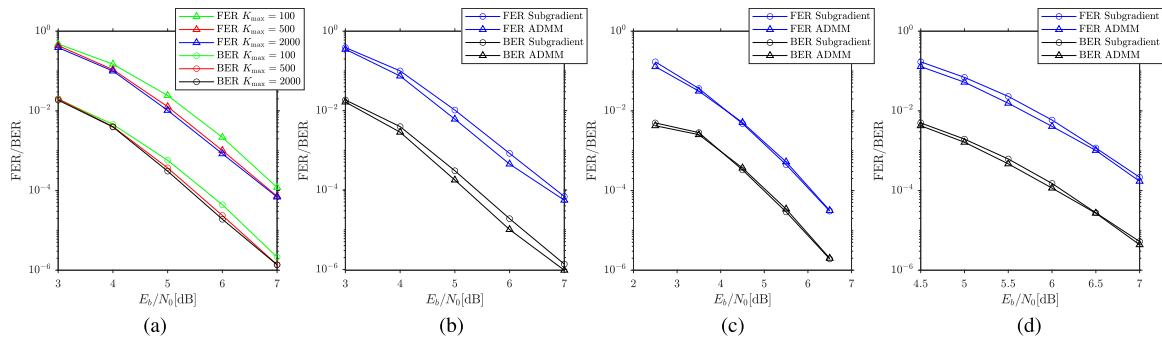


Fig. 1 Error rate performances of the three codes. (a) Performances of C_1 for the proposed method under different values of K_{\max} . (b) Performances of C_1 for the proposed method and the ADMM-based LP decoding method. (c) Performances of C_2 for the proposed method and the ADMM-based LP decoding method. (d) Performances of C_3 for the proposed method and the ADMM-based LP decoding method.

Table 1 Comparisons of ANI values and ADT (ms) of a codeword at different SNR values for the three codes.

Code		C_1				C_2				C_3			
SNR (dB)		4.0	5.0	6.0	7.0	3.5	4.5	5.5	6.5	5.5	6.0	6.5	7.0
ANI	Subgradient	175.08	33.20	12.96	9.01	57.04	16.79	8.67	6.09	104.14	61.04	43.10	35.25
	ADMM	15.93	8.42	4.48	2.40	6.57	4.56	3.10	2.08	18.62	10.41	5.93	3.40
ADT (ms)	Subgradient	0.92	0.17	0.069	0.048	0.18	0.050	0.028	0.018	7.14	4.29	3.06	2.51
	ADMM	5.45	0.61	0.13	0.055	0.20	0.094	0.055	0.035	10.67	6.08	3.57	1.98

operation whose complexity depends on the distribution of the input values. It is believed that the worst-case complexity of the exact polytope projection is quadratic in the check node degree d_c [3], [4]. (We mention that some recent works have considered approximate polytope projection methods, e.g., [8], [11], whose complexity can be linear in the check node degree.) In addition to the polytope projection, we need about $nd_v + 2md_c$ real additions and $2n$ real multiplications in one iteration of the ADMM-based LP decoding [2], [6]. Hence, we conclude that the proposed method has lower per-iteration complexity[†].

4. Simulation Results

In this section, we verify the performance of the proposed method through simulations, which are performed with the Microsoft Visual C++ 6.0 development tool on computers whose configurations are i5-3470 3.2 GHz CPU and 4 GB RAM. The parameters $\alpha^{(k)}$ in (13) are chosen as $\alpha^{(k)} = 1/k$. We choose the algorithm in [2], i.e., the ADMM LP decoding without penalty, for comparison with the proposed method. There are two main reasons for our choice. First, the algorithm in [2] has the lowest per-iteration complexity among all the ADMM-based LP decoding methods. Hence, we compare the per-iteration complexities of the proposed method and this algorithm. Second, it has been shown in [5] that the ADMM LP decoding without penalty can achieve better performances than certain ADMM penalized LP decoding

method when the channel noise is low.

Three codes are used in our simulations. The first code C_1 is a $[120, 100]$ random LDPC code described by a $(3, 18)$ -regular parity-check matrix \mathbf{H}_1 with size 20×120 , which is constructed using the progressive edge-growth algorithm [16]. The second code C_2 is a $[64, 45]$ EG-LDPC code described by a $(3, 8)$ -regular parity-check matrix \mathbf{H}_2 with size 24×64 , which is constructed from EG(2, 8), the two-dimensional Euclidean geometry (EG) over GF(8), according to the methods provided in [15, Example 17.22]. The code C_3 is a $[127, 120]$ Hamming code. It is known that its dual C_3^\perp is a $[127, 7]$ simplex code whose 127 nonzero codewords all have weight 64 [15]. We use all the nonzero codewords in C_3^\perp to construct a $(64, 64)$ -regular parity-check matrix \mathbf{H}_3 with size 127×127 for C_3 . It is worth mentioning that these three codes have relatively large check node degrees.

Figure 1(a) illustrates the frame error rate (FER) and bit error rate (BER) performances of C_1 under different values of K_{\max} . It is known from the figure that the performance gap between K_{\max} being 500 and 2000 is smaller than that between K_{\max} being 100 and 500 for the proposed method. Figure 1(b) compares the performances of C_1 for the proposed algorithm and the ADMM-based LP decoding algorithm, where the maximum numbers of iterations are set as 2000 for both decoding algorithms. As shown in the figure, the FER or BER performance of the proposed method is close to that of the ADMM-based LP decoding method. It is worth mentioning that the performances of the proposed method and the ADMM-based LP decoding method should be identical if K_{\max} goes to infinity. In our simulations, there are some differences in the performances of the two methods,

[†]Note that the proposed method has lower per-iteration complexity can also be verified by our simulation results, which are provided in the next section.

which is due to the fact that K_{\max} is finite.

In order to evaluate the complexities of the two methods for decoding C_1 , we compare the average number of iterations (ANI) values and the average decoding time (ADT) of a codeword, where the maximum numbers of iterations are set as 2000 for both methods. The results are listed in Table 1. We can see from the table that the ANI values of the proposed method are larger than those of the ADMM-based LP decoding method. However, the ADT of a codeword for the proposed method is lower than that for the ADMM-based LP decoding method. Due to this fact, the maximum numbers of iterations are set as 2000 for both decoding methods in the evaluations of C_2 and C_3 .

Figure 1(c) and (d) illustrate the performances of C_2 and C_3 , respectively. Using the results in [17], [18], we can conclude that H_2 and H_3 have good pseudocodeword property, namely, the minimum weight of pseudocodewords of H_2 (resp., H_3) is equal to the minimum distance of C_2 (resp., C_3), which indicates that LP decoding is asymptotically optimal for C_2 and C_3 [17], [18]. We can see from the figures that the performances of the two methods are comparable for each code. This indicates that the parity-check matrix with good pseudocodeword property may be helpful to the proposed method. In addition, it is known from Table 1 that the ADT of a codeword for the proposed method is lower than that for the ADMM-based LP decoding method for all the test cases except for one signal-to-noise ratio (SNR) value of C_3 . This, together with the ANI values, indicates that the per-iteration complexity of the proposed method is much lower than that of the ADMM-based LP decoding method.

5. Conclusion and Future Work

In this letter, we investigated the application of the subgradient algorithm for LP decoding and proposed a processing technique with linear complexity in the check node degrees. The per-iteration complexity of the obtained decoding algorithm is extremely low. In each iteration, the proposed method outputs a binary vector. If the output is a valid codeword, it is an ML codeword. Simulation results on several codes with short lengths suggest that the performance of the proposed method is close to that of the ADMM-based LP decoding. Moreover, the proposed method has advantages in the average decoding time of a codeword.

A major problem of the proposed method is that it has relatively large ANI values. As a future work, we will try to design effective schemes to reduce the ANI values of the proposed method. Moreover, it is interesting to investigate the performance of the proposed method as the code length increases. Finally, it is deserved to further improve the performance of LP decoding with the subgradient method and compare with the ADMM penalized LP decoding methods.

Acknowledgments

The authors greatly appreciate the reviewers' valuable com-

ments.

References

- [1] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. Dissertation, MIT, Cambridge, MA, 2003.
- [2] X. Zhang and P.H. Siegel, "Efficient iterative LP decoding of LDPC codes with alternating direction method of multipliers," Proc. IEEE Int. Symp. Inf. Theory, Istanbul, Turkey, pp.1501–1505, July 2013.
- [3] G. Zhang, R. Heusdens, and W.B. Kleijn, "Large scale LP decoding with low complexity," IEEE Commun. Lett., vol.17, no.11, pp.2152–2155, Nov. 2013.
- [4] S. Barman, X. Liu, S.C. Draper, and B. Recht, "Decomposition methods for large scale LP decoding," IEEE Trans. Inf. Theory, vol.59, no.12, pp.7870–7886, Dec. 2013.
- [5] X. Liu and S.C. Draper, "The ADMM penalized decoder for LDPC codes," IEEE Trans. Inf. Theory, vol.62, no.6, pp.2966–2984, June 2016.
- [6] X. Jiao, J. Mu, Y.-C. He, and C. Chen, "Efficient ADMM decoding of LDPC codes using lookup tables," IEEE Trans. Commun., vol.65, no.4, pp.1425–1437, April 2017.
- [7] J. Bai, Y. Wang, and F. Lau, "Minimum-polytope-based linear programming decoder for LDPC codes via ADMM approach," IEEE Wireless Commun. Lett., vol.8, no.4, pp.1032–1035, Aug. 2019.
- [8] Q. Xia, Y. Lin, S. Tang, and Q. Zhang, "A fast approximate check polytope projection algorithm for ADMM decoding of LDPC codes," IEEE Commun. Lett., vol.23, no.9, pp.1520–1523, Sept. 2019.
- [9] Y. Wei, M.-M. Zhao, M.-J. Zhao, and M. Lei, "ADMM-based decoder for binary linear codes aided by deep learning," IEEE Commun. Lett., vol.24, no.5, pp.1028–1032, May 2020.
- [10] F. Gensheimer, T. Dietz, K. Kraft, S. Ruzika, and N. Wehn, "A reduced-complexity projection algorithm for ADMM-based LP decoding," IEEE Trans. Inf. Theory, vol.66, no.8, pp.4819–4833, Aug. 2020.
- [11] A. Asadzadeh, M. Barakatain, S.C. Draper, and J. Mitra, "SAPA: Sparse affine projection algorithm in ADMM-LP decoding of LDPC codes," Proc. Canadian Workshop on Information Theory, pp.1–6, 2022.
- [12] D. Bertsimas and J.N. Tsitsiklis, Introduction to Linear Optimization, Athena Scientific, Belmont, MA, 1997.
- [13] K. Yang, X. Wang, and J. Feldman, "A new linear programming approach to decoding linear block codes," IEEE Trans. Inf. Theory, vol.54, no.3, pp.1061–1072, March 2008.
- [14] M.C. Coskun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, and F. Steiner, "Efficient error-correcting codes in the short blocklength regime," Physical Communication, vol.34, pp.66–79, June 2019.
- [15] S. Lin and D.J. Costello Jr., Error Correcting Coding: Fundamentals and Applications, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 2004.
- [16] X.-Y. Hu, E. Eleftheriou, and D.M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," IEEE Trans. Inf. Theory, vol.51, no.1, pp.386–398, Jan. 2005.
- [17] R. Smarandache and P.O. Vontobel, "Pseudo-codeword analysis of Tanner graphs from projective and Euclidean planes," IEEE Trans. Inf. Theory, vol.53, no.7, pp.2376–2393, July 2007.
- [18] S.-T. Xia and F.-W. Fu, "Minimum pseudoweight and minimum pseudocodewords of LDPC codes," IEEE Trans. Inf. Theory, vol.54, no.1, pp.480–485, Jan. 2008.