IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 14 December 2022, accepted 27 December 2022, date of publication 4 January 2023, date of current version 9 January 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3234182

RESEARCH ARTICLE

A Simple Check Polytope Projection Penalized Algorithm for ADMM Decoding of LDPC Codes

HUIYANG LIU, PINGQUAN HE, YUN JIANG, QINGLIN ZHANG, AND QIAOQIAO XIA
College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China
Corresponding author: Qiaoqiao Xia (xiaqq@mail.ccnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62101204.

ABSTRACT ADMM penalized decoding method for Low-Density Parity-Check (LDPC) Codes can improve the frame error rate (FER) performance than the standard ADMM decoder by adding penalty terms to the objective function. However, penalty parameter optimization of the penalty terms is very difficult and extremely time-consuming. Additionally, the Euclidean projection onto the check polytope is the most complex part of ADMM-based decoding algorithms. In this paper, by dynamically choosing even-vertex in the check polytope closest to the input vector as the approximate projection, a simple and novel check polytope projection penalized algorithm for ADMM decoding is proposed which can avoid the tedious work of penalty parameter optimization and simplify the check polytope projection. The simulation results show that the proposed algorithm can substantially reduce the projection time while achieving better FER performance when compared with the existing penalized decoding. In particular, the proposed algorithm can save the decoding time by 23% to 41% compared with ADMM-PD-LSA algorithm.

INDEX TERMS Alternating direction method of multipliers (ADMM), low-density parity-check (LDPC) codes, penalty terms, check polytope projection.

I. INTRODUCTION

Low-density parity-check (LDPC) codes have been widely used in communication applications. Feldman et al. proposed a linear programming (LP) decoding algorithm for LDPC codes [1], and it also has be generalized to other codes and channels [2], [3], [4], [5]. It satisfies all-zeros assumption and maximum likelihood certification property, but the research on LP decoding was hampered by the excessive complexity. Therefore, the works [6], [7], [8], [9], [10] were proposed to reduce the complexity of LP decoding model. In [10], Barman et al. applied the framework of alternating direction multiplier method (ADMM) to LP decoding algorithm and proposed the ADMM-LP decoding algorithm to reduce the complexity of LP decoding algorithm. However, the FER performance of ADMM-LP was actually worse than the BP algorithm at the low SNRs. In addition, the Euclidean projection onto the check polytope was very complicated and time-consuming in ADMM decoding algorithms.

The associate editor coordinating the review of this manuscript and approving it for publication was Zesong Fei.

In order to improve the FER performance, Liu et al. proposed the ADMM penalized decoding algorithm by adding penalty terms to the objective function [11], which greatly improved the error correction performance at the low SNRs by penalizing the pseudo-codewords, making them more costly than codewords. Jiao et al. proposed to assign different penalty parameters to variable nodes with different degrees for irregular LDPC codes [12]. Wang et al. designed the improved piecewise penalty functions for ADMM penalized decoder by increasing the slope of the penalty function at the points near $x = 0$ and $x = 1$ [13]. The works [12] and [13] optimized the relevant parameters by differential evolution algorithm. However, the relevant parameters optimization is very difficult and extremely time-consuming. Recently, Wei et al. proposed a novel ADMM check node (CN) penalized decoding algorithm that codeword solutions which satisfy all parity-check equations will have smaller penalty values than non-codeword solutions [14]. The proposed CN-penalized decoder in [14] can improve the FER performance at the expense of increase in the decoding complexity.

To reduce the complexity of Euclidean projection in ADMM decoding algorithm, Zhang et al. proposed the cut search algorithm (CSA) [15] which is more computationally efficient than the "two-slice" projection algorithm proposed in [10]. Zhang et al. [16] demonstrated that the complex projection operation onto the check polytope can be transformed to the projection onto a simplex. Jiao et al have simplified the check polytope projection by establishing lookup table (LUT) and quantization input vector [17], which reduced the computational complexity of Euclidean projection at the expense of huge memory. Moreover, an iterative check polytope projection algorithm was proposed by Wei et al. to achieve the goal of reducing the complexity of projection [18]. However, the convergence speed of the algorithm is slow when keeping the FER performance. A fast iterative check polytope projection algorithm by bisection method was proposed to speed up the projection onto the parity check polytope without increasing computational complexity [19].

Xia et al. have tried to further simplify the Euclidean projection with line segment projection algorithm (LSA) [20], which made a projection onto a line segment consisting of two even-vertices to obtain an approximate projection. Reference [21] recently proposed an efficient hybrid projection algorithm (HPA) by alternately using even-vertex projection algorithm (EVA) and other accurate projection algorithms to increase the percentage of unuseful projections.

In this paper, in order to avoid the tedious work of penalty parameter optimization and simplify the check polytope projection, a simple and novel check polytope projection penalized algorithm for ADMM decoding is proposed by dynamically choosing even-vertex in the check polytope closest to the input vector as the approximate projection. The simulation results show that the proposed algorithm can substantially reduce the projection time while achieving better FER performance when compared with the existing penalized decoding.

II. PRELIMINARIES

Consider an LDPC code C is defined by $m \times n$ parity check matrix $\mathbf{H}$. Let $j \in \mathcal{J} = \{1, 2, 3, \ldots, m\}$ and $i \in \mathcal{I} = \{1, 2, 3, \ldots, n\}$ be the indexes of the rows and columns of the check matrix $\mathbf{H}$, respectively. The degrees of the check node c_j and variable node v_i are the number of 1s in the corresponding row and column of $\mathbf{H}$, and are defined by d_j and d_i, respectively. Let $N_j(N_i)$ be the set of variable nodes(check nodes) adjacent to check node c_j(variable node v_i).

Suppose that a codeword $\mathbf{x} \in C$ is transmitted over a memoryless binary-input output-symmetric (MBIOS) channel, and the received vector is $\mathbf{y}$. The LP decoding model with ADMM can be described as follows:

$$\begin{aligned} &min \sum_i \gamma_i x_i \\ &s.t.\ \mathbf{P}_j\mathbf{x} = \mathbf{z}_j, \quad \mathbf{z}_j \in \mathbb{PP}_{d_j},\ \forall j \in \mathcal{J} \end{aligned} \tag{1}$$

where $\boldsymbol{\gamma} = \{\gamma_i | i \in \mathcal{I}\}$ is the vector of log-likelihood ratios (LLRs), and γ_i can be defined as $\gamma_i = \log(\frac{P_r(y_i/x_i=0)}{P_r(y_i/x_i=1)})$. $\mathbf{P}_j$ is the $d_j \times n$ transfer matrix which selects the d_j components of $\mathbf{x}$ involved in the $j-th$ check node. $\mathbf{z}_j$ is the auxiliary variable of the check node c_j. And $\mathbb{PP}_{d_j}$ represents the check polytope, implying the convex hull of all permutations of a $length - d_j$ binary vector with even number of ones. Furthermore, the augmented Lagrangian function corresponding to formulation (1) can be described as follows:

$$L_\mu(\mathbf{x},\mathbf{z},\boldsymbol{\lambda}) = \boldsymbol{\gamma}^T\mathbf{x} + \sum_{j=1}^{m}\boldsymbol{\lambda}_j{}^T(\mathbf{P}_j\mathbf{x}-\mathbf{z}_j) + \frac{\mu}{2}\sum_{j=1}^{m}\|\mathbf{P}_j\mathbf{x}-\mathbf{z}_j\|_2^2 \tag{2}$$

where $\boldsymbol{\lambda}_j \in \mathbb{R}^{d_j}$ represents the Lagrangian multiplier, and $\mu > 0$ is the penalty parameter. The iterative update rules of $\mathbf{x}$, $\mathbf{z}$ and $\boldsymbol{\lambda}$ can be described as follows:

$$\begin{aligned} x_i{}^{k+1} &= \prod\nolimits_{[0,1]} \frac{1}{|N_i|}\left(\sum_{j\in N_i}\left(\mathbf{z}_{j\to i}^k - \frac{1}{\mu}\boldsymbol{\lambda}_{j\to i}^k\right) - \frac{\gamma_i}{\mu}\right) \\ \mathbf{z}_j{}^{k+1} &= \prod\nolimits_{\mathbb{PP}_{d_j}}\left(\mathbf{P}_j\mathbf{x}^{k+1} + \boldsymbol{\lambda}_j^k/\mu\right) \\ \boldsymbol{\lambda}_j{}^{k+1} &= \boldsymbol{\lambda}_j^k + \mu\left(\mathbf{P}_j\mathbf{x}^{k+1} - \mathbf{z}_j^{k+1}\right) \end{aligned} \tag{3}$$

where $k \geq 0$ denotes the iteration number, and $\prod_{[0,1]}$ is the projection to the interval [0, 1], and $\prod_{\mathbb{PP}_{d_j}}$ represents the Eulidean projection onto the check polytope.

III. A SIMPLE CHECK POLYTOPE PROJECTION PENALIZED ALGORITHM

In this section, we propose a simple check polytope projection penalized algorithm for ADMM decoding of LDPC codes. First, we briefly describe the LSA algorithm [20]. Second, we presents our sources of innovation and the proposed algorithm will be introduced in detail.

A. LINE SEGMENT PROJECTION ALGORITHM

The Euclidean projection onto the check polytope is the most complicated operation in the ADMM decoding algorithm. The precision of the approximate projection depends on the target FER and increases by decreasing the target FER [18], it indicates that the approximate projection does not need to always maintain a high level of projection accuracy and the precision of the projection can be increased as the number of decoding iterations increases. Therefore, Xia et al. proposed a line segment projection algorithm in [20]. The specific steps are shown in Algorithm 1.

According to Algorithm 1, the indicator vector $\boldsymbol{\theta}$ can be calculated as line 1-4. Next, in accordance with $\boldsymbol{\theta}$, we can determine the odd-vertex $\mathbf{O}$ (the vertex has an odd number of 1s) closest to $\mathbf{v}$ as shown in Algorithm 1,line 6. After that, we can find the index p of the element in $\mathbf{v}$ the closest to 0.5 and the index q of the element in $\mathbf{v}$ the second closest to 0.5. Then, we can obtain the two even-vertices $\mathbf{A}$ and $\mathbf{B}$ as shown in Algorithm 1, Line 8. At last, we can calculate the projection of $\mathbf{v}$ onto the line segment L_{AB} by Algorithm 1, Line 9-11.

Algorithm 1 Line Segment projection(LSA)

Input: Vector $\mathbf{v} \in \mathbb{R}^{d_j}$
Output: Vector $\mathbf{z}$

Initialize the indicator vector $\boldsymbol{\theta} : \theta_i = sgn\left(v_i - 0.5\right)$
if $|\{i|\theta_i = 1\}|$ is even **then**
  $i = argmin_{i \in d_j}\left(|v_i - 0.5|\right)$
  $\theta_i = -\theta_i$
end if
Odd-vertex $\mathbf{O} : O_i = \begin{cases} 1, if \ \ \theta_i = 1 \\ 0, if \ \ \theta_i = -1 \end{cases}$
$p = argmin_i\left(|v_i - 0.5|\right)$
$q = argmin_{i/p}\left(|v_i - 0.5|\right)$
Even-vertex $\mathbf{A} = \begin{cases} O_i, if \ \ i \neq p \\ 1 - O_i, if \ \ i = p \end{cases}$
Even-vertex $\mathbf{B} = \begin{cases} O_i, if \ \ i \neq q \\ 1 - O_i, if \ \ i = q \end{cases}$
$\mathbf{AB} = \left\{0, \cdots, B_p - A_p, B_q - A_q, \cdots, 0\right\}$
$\mathbf{Av} = \left\{\cdots, v_p - A_p, \cdots, v_q - A_q, \cdots\right\}$
$\mathrm{t} = \prod_{[0,1]} \frac{(B_p - A_p)(v_p - A_p) + (B_q - A_q)(v_q - A_q)}{2}$
$\mathbf{z} = \mathbf{A} + t \cdot \mathbf{AB}$
return $\mathbf{z}$

B. IMPROVEMENT STRATEGY

In last subsection, we refer to [20] to present the LSA algorithm, it simplifies the calculation of the Euclidean projection onto the check polytope. However, in order to obtain good FER performance, it need to introduce a penalty term to the objective function of the LP decoding problem, and the relevant parameters optimization is very difficult and extremely time-consuming. The result of check polytope projection $\mathbf{z}$ can be regarded as the *copy* of variable node $\mathbf{x}$ that means we can achieve better decoding performance by penalizing the check node. The projection of EVA will make z_j be 0 or 1, which is equivalent to a penalty for $\mathbf{x}$ to keep it away from the fractional vertices. In addition, the projection of EVA is the nearest even vertex, which means that the check node of this part satisfies the local parity-check equation. So, in this section, we try to introduce the idea of penalized decoding to the simplified approximate projection algorithm for check polytope. It can effectively improve the decoding performance without any penalty parameters, that means it will avoid the tedious parameter optimization. The projection operation of the proposed algorithm will find the vertex closest to the vector $\mathbf{v}$ on the unit hypercube, and we denote this vertex as $\mathbf{V_T}$. Then, if the vertex $\mathbf{V_T}$ has an even number of 1s, we will output $\mathbf{V_T}$ as the result of the approximate projection and we call this procedure "even-vertex projection". If $\mathbf{V_T}$ is an odd-vertex(the vertex has an odd number of 1s), we can then choose LSA algorithm or CSA algorithm to perform the Euclidean projection onto the check polytope. For convenience, we call the algorithm improved line segment algorithm (I-LSA) if we combine the proposed idea with LSA algorithm, and call it improved cut search algorithm (I-CSA) if we combine the proposed idea with CSA algorithm. The specific steps of I-LSA are shown in Algorithm 2.

Algorithm 2 Improved Line Segment algorithm(I-LSA)

Input: Vector $\mathbf{v} \in \mathbb{R}^{d_j}$
Output: Vector $\mathbf{z}$

1: Initialize the indicator vector $\boldsymbol{\theta} : \theta_i = sgn\left(v_i - 0.5\right)$
2: Vertex $\mathbf{V_T} : V_{T\,i} = \begin{cases} 1, if \ \ \theta_i = 1 \\ 0, if \ \ \theta_i = -1 \end{cases}$
3: **if** $|\{i|\theta_i = 1\}|$ is even **then**
4:   $\mathbf{z} = \mathbf{V_T}$
5: **else**
6:   $p = argmin_i\left(|v_i - 0.5|\right)$
  $q = argmin_{i/p}\left(|v_i - 0.5|\right)$
7:   Even-vertex $\mathbf{A} = \begin{cases} V_{T\,i}, if \ \ i \neq p \\ 1 - V_{T\,i}, if \ \ i = p \end{cases}$
  Even-vertex $\mathbf{B} = \begin{cases} V_{T\,i}, if \ \ i \neq q \\ 1 - V_{T\,i}, if \ \ i = q \end{cases}$
8:   $\mathbf{AB} = \left\{0, \cdots, B_p - A_p, B_q - A_q, \cdots, 0\right\}$
  $\mathbf{Av} = \left\{\cdots, v_p - A_p, \cdots, v_q - A_q, \cdots\right\}$
9:   $\mathrm{t} = \prod_{[0,1]} \frac{(B_p - A_p)(v_p - A_p) + (B_q - A_q)(v_q - A_q)}{2}$
10:   $\mathbf{z} = \mathbf{A} + t \cdot \mathbf{AB}$
11: **end if**
12: return $\mathbf{z}$

According to Algorithm 2, in accordance with $\boldsymbol{\theta}$, the vertex $\mathbf{V_T}$ closest to $\mathbf{v}$ can be determined. Subsequently, we will confirm the number of 1s in vertex $\mathbf{V_T}$, and if the vertex $\mathbf{V_T}$ has even 1s, we will output it as the projection result, as shown in Algorithm 2, line 3-4. Otherwise, we will use LSA algorithm to calculate the line segment projection, such as described in line 5-12.

IV. SIMULATION RESULTS

In the simulations, the additive white Gaussian noise (AWGN) channel with binary phase shift keying (BPSK) modulation is assumed. Moreover, the adopted codes are the irregular (576,288) rate 1/2 code C_1, irregular (576,432) rate 2/3 code C_2 from ieee802.16e standard [22], and regular (2640,1320) rate 1/2 Margulis code C_3. The check node degree of $C_1 - C_3$ is {6, 7}, {14, 15} and 6, respectively.

The ADMM-PD denotes the ADMM algorithm with the l_1 penalty method, and the parameters μ and α were optimized according to the method in [12]. Besides, the over-relaxation operation is not adopted and the over-relaxation parameter is set to 1. We adopt the early-termination technology based on $\mathbf{H}^{\mathbf{T}}\mathbf{x} = \mathbf{0}$. The maximum number of iterations number is set to 500, and the points plotted in all FER curves are obtained by generating at least 100 error frames.

Figure 1 shows the ratio of even-vertex projection of C_1, C_2 and C_3 codes for the ADMM-LP decoder with the proposed algorithm under the 500 iterations. The points plotted in the curves are obtained by generating at least 100000 frames. As can be seen from the figure, for the three codes, the ratio of even-vertex projection are all over 70% for

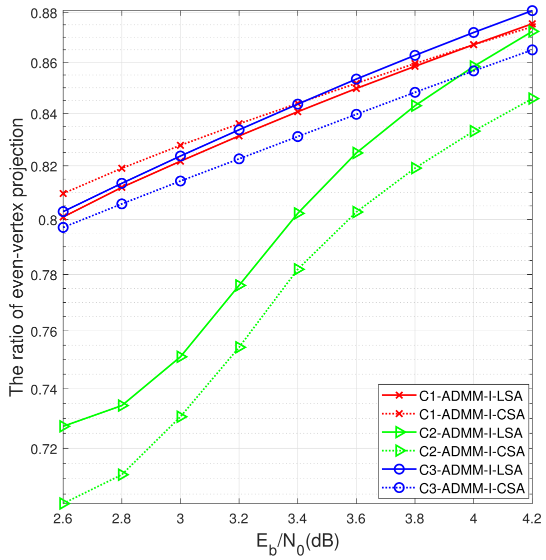


FIGURE 1. **The ratio of even-vertex projection in different SNR.**

the ADMM-LP decoder with the proposed check polytope projection penalized algorithm, and it presents an upward trend with the increase of SNR. It means that most of the vector **v** was projected onto the closest even-vertex $\mathbf{V_T}$, and the operation is very convenient, so the algorithm proposed in this paper can greatly reduce the time of projection calculation and further reduce the complexity of the algorithm.

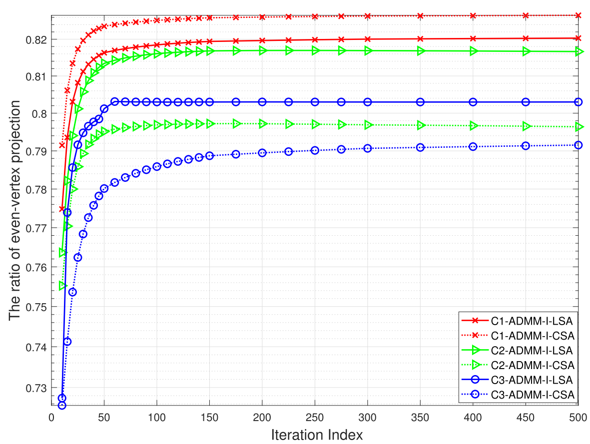
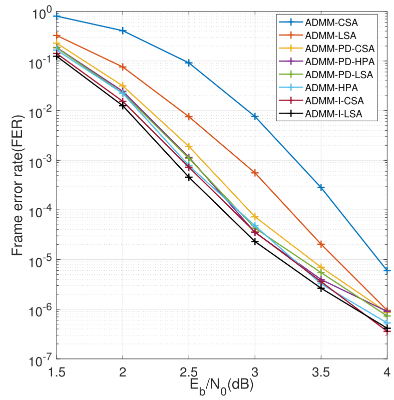
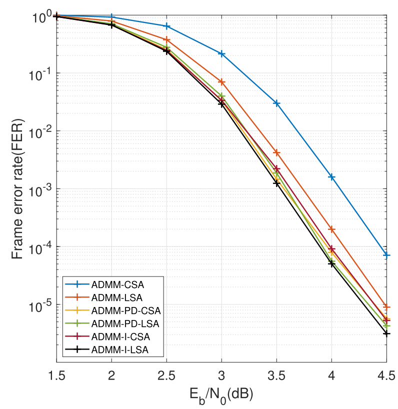


FIGURE 2. **The ratio of even-vertex projection under different number of iterations.**

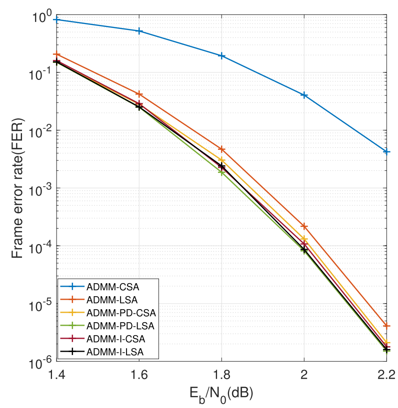
Figure 2 shows the ratio of even-vertex projection of C_1, C_2 and C_3 codes for the ADMM-LP decoder with the proposed algorithm under 3.0 dB, 3.5dB and 2.6dB, respectively, and the points plotted in the curves are obtained by generating at least 100000 frames. As shown in this figure, we can draw a conclusion, as the increase of the iterations, the ratio of even-vertex projection will be raised rapidly and then become converge. For the three codes, the ratio of even-vertex projection will be over 79% when the number of iterations is over 150.



(a) $C_1 - (576, 288)$.



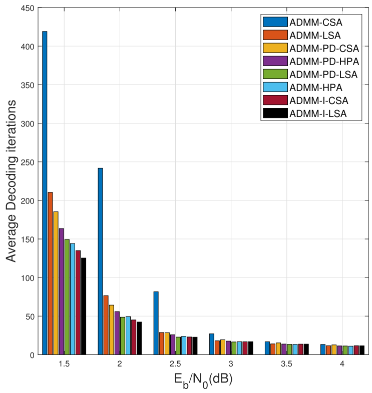
(b) $C_2 - (576, 432)$.



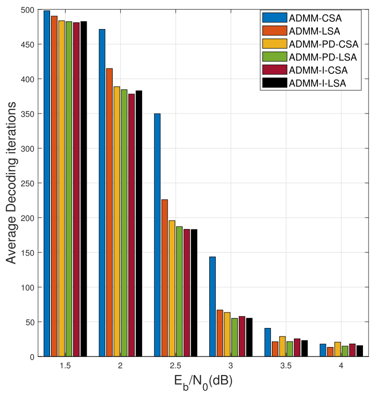
(c) $C_3 - (2640, 1320)$.

FIGURE 3. **The FER performance of C_1, C_2 and C_3 with different algorithms.**

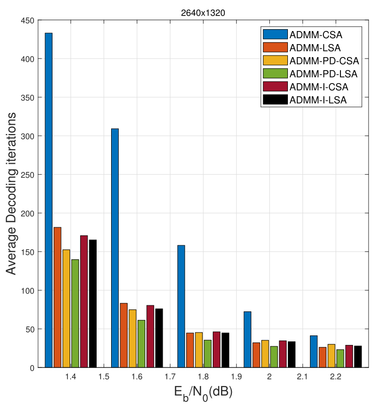
Figure 3 shows the FER performance of C_1, C_2 and C_3 for the ADMM-LP decoder with CSA, LSA, HPA and the proposed algorithms. As shown in the figure, for the three codes, the proposed ADMM-I-CSA and ADMM-I-LSA will achieve better or similar FER performance compared with other algorithms. For example, for C_1 code, when the $FER = 10^{-4}$, performance gains for the proposed ADMM-I-LSA are about 0.1dB to ADMM-HPA and ADMM-PD-LSA.



(a) $C_1 - (576, 288)$.



(b) $C_2 - (576, 432)$.

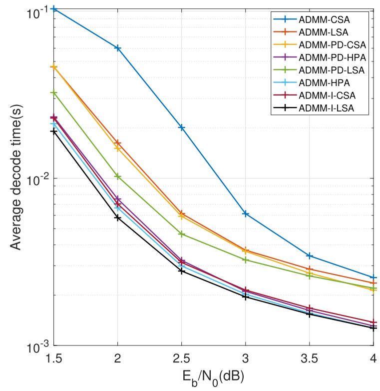


(c) $C_3 - (2640, 1320)$.

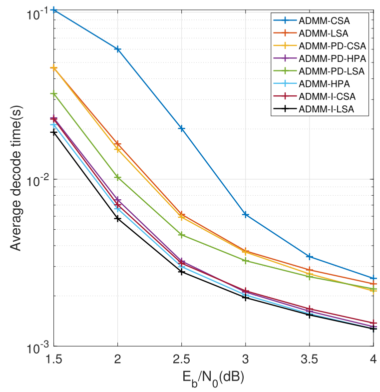
FIGURE 4. **Average number of iterations of C_1, C_2 and C_3 with different algorithms.**

Besides, performance gains for the proposed ADMM-I-CSA are about 0.15dB to ADMM-PD-CSA.

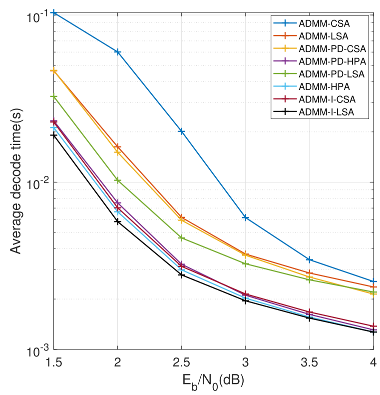
Figure 4 shows the average number of iterations for $C_1 - C_3$ with various projection algorithms. It can be seen from the figure, for C_1, when SNR $= 2.0$dB, compared with ADMM-HPA, ADMM-PD-LSA, ADMM-PD-HPA and ADMM-PD-CSA, the average number of iterations of the proposed ADMM-I-LSA is reduced by 14%, 13%, 24% and 34% respectively. And the average number of iterations of the proposed ADMM-I-CSA is also reduced compared with existing algorithms. For C_2, when SNR $= 2.5$dB, the average iteration of the proposed ADMM-I-LSA and ADMM-I-CSA are basically the same, which is reduced by 2% compared with ADMM-PD-LSA, and 6% compared with ADMM-PD-CSA.



(a) $C_1 - (576, 288)$.



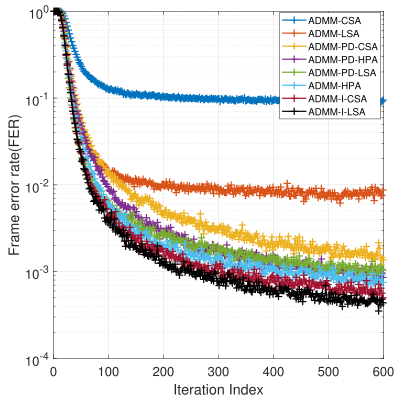
(b) $C_2 - (576, 432)$.



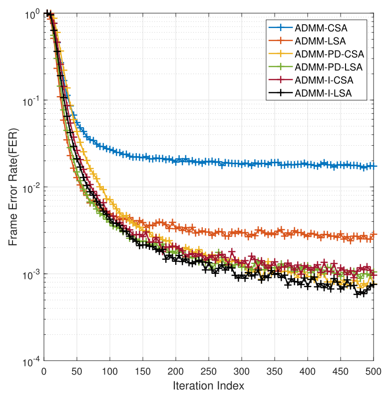
(c) $C_3 - (2640, 1320)$.

FIGURE 5. **Average decoding time of C_1, C_2 and C_3 with different algorithms.**

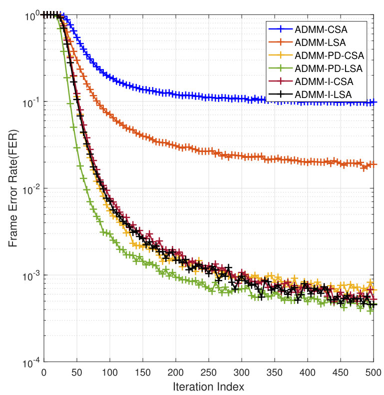
Figure 5 shows the average decoding time for $C_1 - C_3$ with various projection algorithms. As can be seen from the figure, for the three codes, the average decoding time of the proposed



(a) $C_1 - (576, 288)$.



(b) $C_2 - (576, 432)$.



(c) $C_3 - (2640, 1320)$.

FIGURE 6. The frame error rate of C_1, C_2 and C_3 with different algorithms at different number of iterations.

algorithms in this paper is decreased compared with other algorithms. For instance, for C_1 code, the proposed ADMM-I-LSA can save the average decoding time with regard to ADMM-PD-LSA, by roughly 40% at 3.0 dB. Besides, the proposed ADMM-I-CSA can save the average decoding time with regard to ADMM-PD-CSA, by roughly 41% at 3.0 dB.

Figure 6 shows the FER of each algorithm for the three codes at the different number of iterations, and the points plotted in the curves are obtained by generating at least 100000 frames and 100 error frames. As shown in the figure, for the three codes, the convergence behavior of these algorithms are similar. For C_1 and C_2 codes, the proposed ADMM-I-LSA and ADMM-I-CSA have better FER performance than other algorithms at the same number of iterations. For C_3 code, the proposed ADMM-I-LSA and ADMM-I-CSA have a similar FER performance compared to the ADMM-PD-CSA with the increase of the number of iterations.

V. CONCLUSION

To summarize, we propose a simple and novel check polytope projection penalized algorithm for ADMM decoding. On the one hand, compared with the existing penalized ADMM decoding algorithms, the algorithm we proposed can not only avoid the tedious work of parameters optimization, but also have an improved performance of FER. On the other hand, according to the experimental reseults, the proposed ADMM-I-LSA can save 23% to 41% average decoding time compared with other existing decoding algorithm with achieving better FER performance.

REFERENCES

[1] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.

[2] M. F. Flanagan, V. Skachek, E. Byrne, and M. Greferath, "Linear-programming decoding of nonbinary linear codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4134–4154, Sep. 2009.

[3] M. Punekar, P. O. Vontobel, and M. F. Flanagan, "Low-complexity LP decoding of nonbinary linear codes," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3073–3085, Aug. 2013.

[4] N. Goela, S. B. Korada, and M. Gastpar, "On LP decoding of polar codes," in *Proc. IEEE Inf. Theory Workshop*, Dublin, Ireland, Aug. 2010, pp. 1–5.

[5] B.-H. Kim and H. D. Pfister, "Joint decoding of LDPC codes and finite-state channels via linear-programming," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1563–1576, Dec. 2011.

[6] P. O. Vontobel and R. Koetter, "Towards low-complexity linear-programming decoding," in *Proc. 4th Int. Symp. Turbo Codes Rel. Topics (ISTC)*, Munich, Germany, Apr. 2006, pp. 1–9.

[7] 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, Mar. 2008.

[8] M. H. Taghavi, A. Shokrollahi, and P. H. Siegel, "Efficient implementation of linear programming decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5960–5982, Sep. 2011.

[9] D. Burshtein, "Iterative approximate linear programming decoding of LDPC codes with linear complexity," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4835–4859, Nov. 2009.

[10] 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.

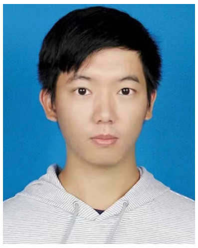
[11] X. Liu and S. C. Draper, "The ADMM penalized decoder for LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 2966–2984, Jun. 2016.

[12] X. Jiao, H. Wei, J. Mu, and C. Chen, "Improved ADMM penalized decoder for irregular low-density parity-check codes," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 913–916, Jun. 2015.

[13] B. Wang, J. Mu, X. Jiao, and Z. Wang, "Improved penalty functions of ADMM penalized decoder for LDPC codes," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 234–237, Feb. 2017.

[14] H. Wei and A. H. Banihashemi, "ADMM check node penalized decoders for LDPC codes," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3528–3540, Jun. 2021.

[15] X. Zhang and P. H. Siegel, "Efficient iterative LP decoding of LDPC codes with alternating direction method of multipliers," in *Proc. IEEE Int. Symp. Inf. Theory*, Turkey, Jul. 2013, pp. 1501–1505.
[16] 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.
[17] 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, Apr. 2017.
[18] H. Wei and A. H. Banihashemi, "An iterative check polytope projection algorithm for ADMM-based LP decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 29–32, Jan. 2018.
[19] Y. Lin, Q. Xia, W. He, and Q. Zhang, "A fast iterative check polytope projection algorithm for ADMM decoding of LDPC codes by bisection method," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 102, no. 10, pp. 1988–1991, Oct. 2019.
[20] 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, Sep. 2019.
[21] Q. Xia, X. Wang, H. Liu, and Q. L. Zhang, "A hybrid check polytope projection algorithm for ADMM decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 108–112, Jan. 2021.
[22] *LDPC Coding for OFDMA PHY*, Standard IEEE Std. C802.16e-05/0066r3, 2005.

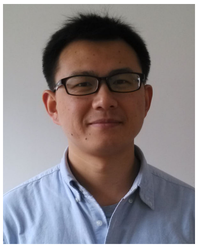


HUIYANG LIU was born in Hubei, China, in 1995. He received the B.Eng. degree from Hubei Minzu University. He is currently pursuing the M.E. degree in engineering with Central China Normal University. His research interest includes channel code theory.

PINGQUAN HE was born in Hubei, China, in 1997. He received the B.Eng. degree from Wuhan Textile University, Wuhan, China, and the M.E. degree in engineering from Central China Normal University. His research interest includes channel code theory.

YUN JIANG was born in Hubei, China, in 1999. She received the B.Eng. degree from Central China Normal University, China, where she is currently pursuing the M.E. degree in engineering. Her research interest includes channel code theory.

QINGLIN ZHANG received the Ph.D. degree from Wuhan University, in 2010. From 2010 to 2013, he worked as a Postdoctoral Researcher at the Physics Postdoctoral Mobile Station, Wuhan University. Since 2013, he has been working with the College of Physical Science and Technology, Central China Normal University. His research interests include EDA technology, embedded systems, and image processing.



QIAOQIAO XIA received the Ph.D. degree from Wuhan University, in 2014. He is currently an Associate Professor with the College of Physical Science and Technology, Central China Normal University. His research interests include coding theory, high-speed digital design, and deep learning.