IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024CIP0016

Publicized:2024/11/14

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

LETTER

Bisection Method Assisted Affine Projection Algorithm in ADMM-LP Decoding of LDPC Codes

Rui CHENG†, **Yun JIANG**†, **Qinglin ZHANG**†, *and* **Qiaoqiao XIA**†a), *Nonmembers*

SUMMARY Many researchers have proposed optimization methods to reduce the computational complexity of the Euclidean projection onto check polytope in the alternating direction method of multipliers (ADMM) decoding for Low-Density Parity-Check (LDPC) codes. Existing the sparse affine projection algorithm (SAPA) projects the vector to be projected onto an χ-dimension affine hull and the dimension χ is fixed, resulting in deteriorating decoding performance. In this letter, bisection method assisted affine projection algorithm is proposed to determine the correct projection dimension for each the vector to be projected with the bisection method iterative algorithm. Simulation results show that the proposed algorithm can improve the accuracy of projection results by 68.2%. The FER performance of the proposed algorithm is almost the same as that of the exact projection algorithm, and compared with the sparse affine projection algorithm (SAPA), it can improve the FER performance by 0.14dB as well as save average number of iterations by 3.2%.

key words: *Alternating direction method of multipliers (ADMM), low-density parity-check (LDPC) codes, check polytope projection, bisection method, sparse affine projection algorithm.*

1. Introduction

The performance of Low-Density Parity-check (LDPC) codes is close to Shannon capacity, have been widely used in the field of channel coding and decoding research. Linear programming (LP) decoding of low-density parity check (LDPC) codes was first proposed by Feldman in [1], having all-zeros assumption and the maximum likelihood (ML) certificate property. However, LP decoding was not well developed because of its high complexity. Recently, Barman et al. applied the alternating direction multiplier method (ADMM) [2] in the field of decoding and proposed an LP decoding algorithm model based on ADMM (ADMM-LP) [3]. The ADMM-LP decoding algorithm can reduce the complexity of LP decoding and eliminate the error floor phenomenon that occurs in the traditional belief propagation (BP) decoding algorithm. However, the ADMM-LP decoding algorithm still had relatively high complexity, and its decoding performance at the low SNRs was inferior to that of the BP decoding algorithm. In order to improve the decoding performance, Liu et al. proposed the ADMM penalized decoding algorithm by adding penalty terms to make the decoding result closer to the integer codeword [4], thereby improving the decoding performance. Jiao et al. proposed a method for irregular LDPC codes, using different penalty parameters for variable nodes of different degrees [5]. Wang et al. designed the improved piecewise penalty functions for ADMM penalized decoder [6]. As the most complex and time-consuming operation in ADMM-LP, Euclidean projection has been studied by many scholars. To simplify the Euclidean projection operation and reduce the complexity in ADMM decoding algorithm, Zhang et al. proposed a projection algorithm based on cut search algorithm (CSA) [7]. Wasson et al. proposed to combine CSA with simplex projection algorithm and implement in hardware [8]. G. Zhang et al. replaced the projection onto the check polytope with the projection onto the simplex [9]. Jiao et al. proposed to simplify the projection operation by using simple table lookup operations [10]. Subsequently, Jiao et al. reduced storage resources and facilitate hardware implementation by applying a non-uniform quantization method of projection vector [11]. Moreover, Wei et al. proposed an iterative check polytope projection algorithm [12]. However, this algorithm requires more iterations to achieve convergence. Inspired by [12] Lin et al. proposed a fast iterative check polytope projection algorithm by bisection method [13]. Xia et al. proposed the even-vertex projection algorithm (EVA) [14], [15], projecting onto the closest even-vertex. Reference [16] proposed the line-segment algorithm which made a projection onto a line segment consisting of two closest even-vertices. Asadzadeh et al. proposed the sparse affine projection algorithm (SAPA) [17], projecting onto the affine hull of a small number of vertices of the polytope. However, experiments show that the FER performance of these approximation algorithms can be deteriorated in low iteration regime.

In this letter, bisection method assisted affine projection algorithm is proposed in order to improve the FER performance in low-iteration regime. Different from the sparse affine projection algorithm (SAPA), the proposed algorithm selects the different dimension χ of the affine hull for different vectors to be projected. Simulation results show that the algorithm can maintain the FER performance and convergence rate of the exact projection algorithm (CSA), the accuracy of projection results is as high as 99.6%. Compared with the sparse affine projection algorithm (SAPA), the proposed algorithm can improve the accuracy of projection results by 68.2% and improve the FER performance by 0.14 dB as well as save average number of iterations by 3.2%.

†The authors are with the College of Physical Science and Technology, Central China Normal University, Wuhan, 430079, China.

a) E-mail: xiaqq@ccnu.edu.cn (Corresponding author)

2. Preliminaries

Consider an LDPC code $\boldsymbol{C}$ defined by an $m \times n$ parity check matrix $\mathbf{H}$. Let $i \in I = \{1, 2, 3..., n\}$ and $j \in J = \{1, 2, 3..., m\}$ be the set for variable nodes and check nodes of $\boldsymbol{C}$, respectively. Let d_i (d_j) denotes the degree of variable node v_i (check node c_j). The set of check nodes (variable nodes) adjacent to variable node v_i (check node c_j) is denoted by $N_{v(i)}$ ($N_{c(j)}$).

Suppose that a codeword $\boldsymbol{x} \in \boldsymbol{C}$ is transmitted over a symmetric memoryless channel and $\boldsymbol{y}$ is the received vector. The LP decoding model with ADMM can be described as follows:

$$\begin{aligned} &\min \sum_{i=1}^{n} \gamma_i x_i \\ &s.t. \boldsymbol{P}_j x = z_j, z_j \in P_{d_j}, \forall j \in J \end{aligned} \tag{1}$$

Where $\boldsymbol{\gamma}$ is the vector of log-likelihood ratios (LLRs), and the i-th entry of $\boldsymbol{\gamma}$ can be defined as $\gamma_i = \log(\frac{\Pr(y_i|x_i=0)}{\Pr(y_i|x_i=1)})$. $\boldsymbol{P}_j$ is the $d_j \times n$ transfer matrix which selects the d_j components of $\mathbf{x}$ involved in the j-th check node. z_j is the auxiliary variable of the check node c_j. P_{d_j} is the check polytope, implying the convex hull of all permutations of a length-d_j binary vector with even number of ones.

The augmented Lagrangian function corresponding to formulation (1) can be described as follows:

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

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 $\boldsymbol{x}$, z and $\boldsymbol{\lambda}$ can be described as follows:

$$\boldsymbol{x}_i{}^{k+1} = \prod\nolimits_{[0,1]} \frac{1}{d_i}\Big(\sum_{j\in N_i} \big((z_j^k)_i - \frac{1}{\mu}(\boldsymbol{\lambda}_j^k)_i\big) - \frac{1}{\mu}\gamma_i\Big) \tag{3}$$

$$z_j{}^{k+1} = \prod\nolimits_{P_{d_j}} (\boldsymbol{P}_j\boldsymbol{x}^{k+1} + \boldsymbol{\lambda}_j^k/\mu) \tag{4}$$

$$\boldsymbol{\lambda}_j{}^{k+1} = \boldsymbol{\lambda}_j{}^k + \mu(\boldsymbol{P}_j\boldsymbol{x}^{k+1} - z_j^{k+1}) \tag{5}$$

where $k \geq 0$ is the iteration number, $\prod_{[0,1]}$ is the projection to the interval [0,1], and $\prod_{P_{d_j}}$ is the check polytope projection operation.

3. ADMM-LP Decoding Algorithm With Bisection Method Assisted Affine Projection Algorithm

Projection onto the parity polytope is considered to be the most complicated and time-consuming operation in the ADMM-LP decoding algorithm, as it requires sorting a vector of size d_j and finding the proper shift to yield a convex combination of the closest even-weight vertices of P_{d_j}. Hence, many scholars have conducted research on this issue, proposing various exact projection algorithms and approximate projection algorithms and enhancing the decoding performance of ADMM-LP. In this section, we will propose the bisection method assisted affine projection algorithm (BM-SAPA) inspired by SAPA and illustrate it in detail.

3.1 SAPA

Affine Projection Algorithm (APA) is to project a d_j-dimensional vector onto the affine hull of the d_j vertices instead of the check polytope projection, avoiding sorting operations and complex comparison operations. However, simulation results show that APA reduces the decoding performance. On this basis, A. Asadzadeh et al. further proposed the sparse affine projection algorithm (SAPA), projecting onto the affine hull of the χ even-vertices closest to the vector to be projection and $\chi \in \{1, 2, 3, ..., d_j\}$. Based on the value of χ, SAPA is denoted as χ-SAPA. The specific steps of χ-SAPA are shown in Algorithm 3 of reference [17].

As shown in line 1-2 of Algorithm 3 in reference [17], SAPA only involves a partial sorting on the χ minimum elements in the vector to be projected and calculates the sum of those elements as the affine shift. It is worthy of mention that χ-SAPA does not need to perform projection operations on the unit cube because its projection region is the affine hull composed of even vertices.

Reference [17] shows that the outcome of 1-SAPA is equal to that of EVA. The difference between 2-SAPA and LSA is that LSA projects on a line segment, while 2-SAPA projects on an entire straight line. In addition, the experimental results in reference [17] show that 3-SAPA has the best FER performance. Therefore, 3-SAPA is adopted for experimental simulation in the following experiments.

3.2 BM-SAPA

Although SAPA can achieve similar decoding performance to CSA in the high-iteration regime, some projection results of SAPA may not be on the check polytope in low-iteration regime, resulting in the deterioration of decoding performance.

In order to make the projection more accurate for low-iteration regime, we try to propose the bisection method assisted affine projection algorithm (BM-SAPA). Asadzadeh proves that there exists at least one value $\chi \in \{1, 2, 3, ..., d_j\}$ for which χ-SAPA will reproduce the exact projected point. For different vectors to be projected, their corresponding correct projection dimensions are different, so the dimension χ of the affine hull that can achieve accurate projection is also different. In BM-SAPA, according to the description of the bisection method iterative algorithm (BMIA) in reference[13], we can reduce the value range $[\beta_{low}, \beta_{up}]$ of coefficients η in the exact projection calculation $z = \prod_{[0,1]} (v - \eta\theta_V)$ by the bisection method. The BM-SAPA aims to select the correct projection dimension for different vectors to be projected.

The speci c process of BM-SAPA is shown in Algorithm 1.

Algorithm 1 The Bisection Method Assisted A ne Projection Algorithm (BM-SAPA)

Input: Vector $\boldsymbol{v} \in \mathbb{R}^{d_j}$, indicator vector $\boldsymbol{\theta}_V$, I_{max}
Output: Projection $\boldsymbol{z}$

1: $\beta_{max} \leftarrow \frac{1}{2}\left(\min_{\theta_{V,i}=1} v_i - \max_{\theta_{V,j}=-1} v_j\right), p \leftarrow |\{i|\theta_{V,i}=1\}| - 1$
2: Initialize $\beta_{low} \leftarrow 0, \beta_{up} \leftarrow \beta_{max}, iter \leftarrow 0$
3: **for** $iter = 1$ **to** I_{max} **do**
4: $\quad \beta \leftarrow \frac{1}{2}(\beta_{up} + \beta_{low})$
5: $\quad \boldsymbol{z} \leftarrow \prod_{[0,1]^{d_j}} (\boldsymbol{v} - \beta\theta_V)$
6: $\quad iter \leftarrow iter + 1$
7: $\quad$ **if** $\theta_V^T \boldsymbol{z} < p$ **then**
8: $\quad\quad \beta_{up} \leftarrow \beta$
9: $\quad$ **else**
10: $\quad\quad \beta_{low} \leftarrow \beta$
11: $\quad$ **end if**
12: **end for**
13: $\chi = |\{i|0 < v_i - \beta_{low}\theta_{V,i} < 1\}|$
14: $\boldsymbol{z} = \chi - SAPA(\boldsymbol{v})$
15: **return** $\boldsymbol{z}$

The bisection method iteration can be described in lines 1-11 of Algorithm 1, the number of iterations is $I_{\max}$. The values of β_{low} and β_{up} are very close through $I_{\max}$ iteration, β_{low} is selected to judgment the number of elements in the interval [0,1] in $v - \beta_{low}\theta$ and the number is taken as the dimension of the a ne hull.

$\prod_{[0,1]}(v_i - \beta_{low}\theta_{V,i})$ is the estimate of the projection result. Step 12 indicates that there are χ-element in [0,1], and the remaining $d_j - \chi$-element is outside [0,1]. The value of χ has no e ect on the projection results outside the interval [0,1], so we choose χ as the dimension of a ne projection. $I_{\max}$ is the pre-determined value in the algorithm. Simulation experiments are used to illustrate the $I_{\max}$ in the next section.

4. Simulation Results

In the simulations, the additive white Gaussian noise (AWGN) channel with binary phase shift keying (BPSK) modulation is assumed. Moreover, we consider three LDPC codes with di erent rates: the regular (2640, 1320) rate-1/2 Margulis code C_1, the regular (1920, 640) rate-1/3 Gallager code C_2, the irregular (576, 432) rate-3/4 code C_3 from IEEE 802.16e standard [18]. The check node degrees of C_1, C_2 and C_3 are 6, 4 and $\{14,15\}$, respectively.

The ADMM-LP decoding algorithm with L2 penalty combined with over-relaxation technique is adopted in the simulations, and the relaxation coe cient is 1.9. Penalty coe cient μ are set to 4.0, 5.5 and 5.5 for C_1, C_2 and C_3, respectively. Parameter α are set to 0.9, 0.8 and 1.9 for C_1, C_2 and C_3, respectively. The maximum number of iterations for ADMM-LP decoder is set to 20. The points plotted in all FER curves are obtained by generating at least 100 error frames.

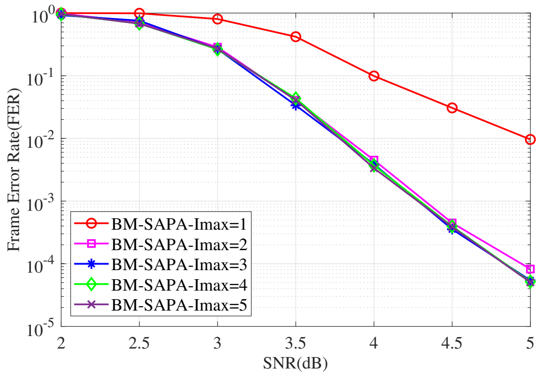


Fig. 1 The FER performance of C_3 for di erent $I_{\max}$ in BM-SAPA.

Table 1 The average error of ξ_j between BMIA, SAPA, BM-SAPA and CSA for C_1, C_2 and C_3

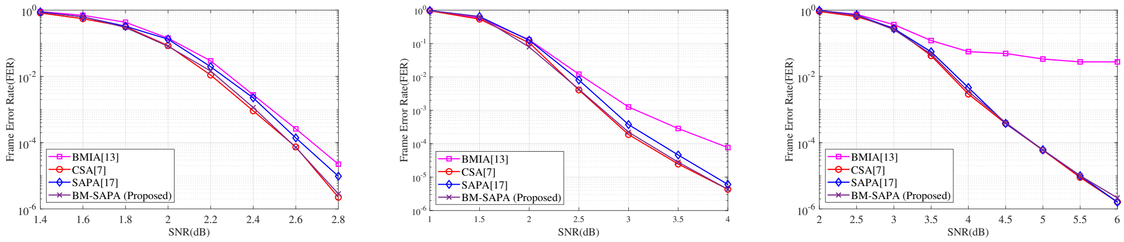
Code	SNR(dB)	BMIA[13]	SAPA[17]	BM-SAPA (Proposed)
C_1	2.0	6.54×10^{-4}	1.33×10^{-2}	2.02×10^{-4}
	2.2	6.55×10^{-4}	1.44×10^{-2}	2.86×10^{-4}
	2.4	6.60×10^{-4}	1.53×10^{-2}	3.36×10^{-4}
	2.6	6.50×10^{-4}	1.65×10^{-2}	4.07×10^{-4}
	2.8	6.65×10^{-4}	1.78×10^{-2}	4.33×10^{-4}
C_2	1.5	3.61×10^{-4}	1.30×10^{-2}	2.07×10^{-5}
	2.0	3.67×10^{-4}	1.52×10^{-2}	2.37×10^{-5}
	2.5	3.66×10^{-4}	1.84×10^{-2}	2.39×10^{-5}
	3.0	3.73×10^{-4}	2.27×10^{-2}	2.73×10^{-5}
	3.5	3.86×10^{-4}	2.74×10^{-2}	2.89×10^{-5}
C_3	2.5	6.99×10^{-4}	1.11×10^{-2}	3.26×10^{-5}
	3.0	7.24×10^{-4}	1.10×10^{-2}	3.57×10^{-5}
	5.5	7.59×10^{-4}	1.07×10^{-2}	4.01×10^{-5}
	4.0	7.80×10^{-4}	1.02×10^{-2}	4.71×10^{-5}
	4.5	7.89×10^{-4}	1.02×10^{-2}	5.39×10^{-5}

Table 2 The accuracy of ξ_j for SAPA, BM-SAPA for C_1, C_2 and C_3

Code	SNR(dB)	SAPA[17]	BM-SAPA (Proposed)
C_1	2.0	24.0%	96.9%
	2.2	24.6%	97.3%
	2.4	26.1%	97.4%
	2.6	27.0%	97.6%
	2.8	28.7%	97.9%
C_2	1.5	30.1%	99.6%
	2.0	31.5%	99.7%
	2.5	33.5%	99.7%
	3.0	35.5%	99.8%
	3.5	37.5%	99.8%
C_3	2.5	36.2%	88.3%
	3.0	34.6%	88.4%
	3.5	33.6%	88.8%
	4.0	30.6%	89.6%
	4.5	27.7%	90.7%

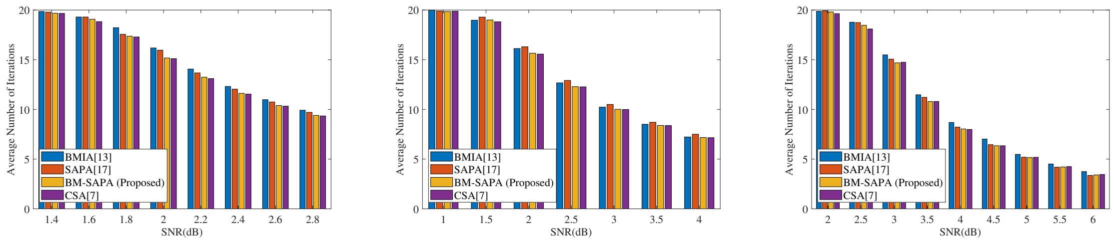
Figure 1 shows the FER performance of di erent $I_{\max}$ for C_3 in BM-SAPA. When $I_{\max} = 1$, BM-SAPA has the worst FER performance, when $I_{\max} \geq 3$, the FER performance is no longer improved with the increase of $I_{\max}$. Therefore, in our simulations, $I_{\max}$ in bisection method iterative is set to 3.

Table 1 describes the average error of ξ_j between approximate projection algorithm (BMIA, SAPA and BM-SAPA) and exact projection algorithm (CSA) for C_1, C_2 and C_3. The speci c de nition of average error : average er-



(a) C_1 (b) C_2 (c) C_3

Fig. 2 The FER performance for BMIA, CSA, SAPA and BM-SAPA for C_1, C_2 and C_3.



(a) C_1 (b) C_2 (c) C_3

Fig. 3 The average number of iterations for BMIA, SAPA, BM-SAPA and CSA for C_1, C_2 and C_3.

ror = $(\sum \| z_{jacc} - z_{japprox} \|)/N$, where z_{jacc} represents the projection result of the exact projection (CSA), $z_{japprox}$ represents the projection result of approximate projection (BMIA, SAPA and BM-SAPA), N is the total number of projections.

In the experiment, at least 1000000 projections, the distances between the di erent approximate projection and the exact projection are recorded in each projection operation, and calculate the average error, respectively. The smaller the value of error, the closer the result of approximate projection is to the exact projection. As can be seen from the table, the average error of z_j between BM-SAPA and CSA is much smaller than that between SAPA and CSA. For example, the average error of z_j between BM-SAPA and CSA is only 2.37×10^{-5} for C_2 at 2.0 dB, and it is about 1 of SAPA.

Table 2 shows the accuracy of z_j for SAPA, BM-SAPA for C_1, C_2 and C_3, that is , the proportion of z_j obtained by approximate projection that is equal to z_j obtained by CSA. BMIA approaches the exact value of the approximate projection result in an iterative manner, therefore, BMIA was not included in the comparison in this experiment. The accuracy of z_j for BM-SAPA is much higher than that of SAPA. For example, for C_2, when SNR = $2dB$, the accuracy of z_j for BM-SAPA is 99.7%, which is 68.2% more than that of SAPA.

Figure 2 shows the FER performance for di erent projection algorithms for C_1, C_2 and C_3 for low-iteration regime (maximum set to 20). It is suggested that the FER performance of BM-SAPA is almost the same as that of the exact projection algorithm CSA and outperforms that of SAPA. Speci cally, for C_2, when FER = 2×10^{-4}, the FER performance of SAPA is worse than that of BM-SAPA about 0.14 dB.

Figure 3 plots the average number of iterations for BMIA, SAPA, BM-SAPA and CSA for C_1, C_2 and C_3 for low-iteration regime. In the simulations, we adopt the early-termination technology based on $H^T x = 0$, the actual number of iterations during decoding may be less than the maximum number of iterations set. Therefore, fewer actual iterations mean faster convergence. It can be seen from the gure, the average number of iterations of the decoder with the proposed BM-SAPA is almost the same as that of the decoder with CSA and less than that of the decoder with SAPA, which means that the proposed algorithm converges quickly. For instance, for C_1, when SNR = $2.2dB$, the average number of iterations of the proposed BM-SAPA is reduced by 3.2%.

5. Conclusion

To summarize, we propose a bisection method assisted a ne projection algorithm (BM-SAPA), by replacing check polytope projection with projection onto the a ne hull of the χ even-vertices closest to the vector to be projected. For di erent vectors to be projected, we select di erent correct projection dimensions χ through bisection method iteration. Many existing approximate projection algorithms sacri ce certain decoding performance at low iterations in order to reduce the complexity of decoding and save time. The FER performance and convergence rate of BM-SAPA are almost the same as that of the exact projection algorithm for low iteration. Compared with the sparse a ne projection algorithm (SAPA), the proposed algorithm can improve the accuracy of projection results by 68.2%, the FER performance by 0.14 dB as well as save average number of iterations by 3.2%.

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] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, \Distributed optimization and statistical learning via the alternating direction method of multipliers," Found. Trends Mach. Learn., vol. 3, no.1, pp. 1{122, Jan. 2011.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] X. Zhang and P. H. Siegel, \E cient iterative LP decoding of LDPC codes with alternating direction method of multipliers," Proc. IEEE Int. Symp. Inf. Theory, Jul. 2013, pp. 1501{1505.
[8] M. Wasson, M. Milicevic, S. C. Draper and G. Gulak, \Hardware-Based Linear Program Decoding With the Alternating Direction Method of Multipliers," IEEE Trans. Signal Process., vol. 67, no. 19, pp. 4976-4991, Oct. 2019.
[9] 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.
[10] X. Jiao, J. Mu, Y. He, and C. Chen, \E cient ADMM decoding of LDPC codes using lookup tables," IEEE Trans. Commun., vol. 65, no. 4, pp. 1425{1437, Apr. 2017.
[11] X. Jiao, Y. He, and J. Mu, \Memory-Reduced lookup tables for e cient ADMM decoding of LDPC codes," IEEE Sig. Proc. Lett., vol. 25, no. 1, pp. 110{114, Jan. 2018.
[12] 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.
[13] 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 Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 102, no. 10, pp. 1406{1410, 2019.
[14] 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, 2020.
[15] Y. Zheng, Y. Lin, Z. Zhang, Q. Zhang, and Q. Xia, \An enhanced HDPC-EVA decoder based on ADMM," IEICE Trans. Fundamentals, vol. 104, no. 10, pp. 1425{1429, 2021.
[16] 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.
[17] A. Asadzadeh, M. Barakatain, S. C. Draper and J. Mitra, \SAPA: Sparse A ne Projection Algorithm in ADMM-LP Decoding of LDPC Codes," 2022 17th Canadian Workshop on Information Theory (CWIT), Ottawa, ON, Canada, 2022, pp. 27-32.
[18] LDPC Coding for OFDMA PHY. IEEE Standard C802. 16e-05 / 0066r3, 294[S] . New York: IEEE, 2005.