Proximal-ADMM Decoder for Nonbinary LDPC Codes

Yongchao Wang, Member, IEEE, Jing Bai

Abstract—In this paper, we develop an efficient decoder via the proximal alternating direction method of multipliers (proximal-ADMM) technique for nonbinary linear block codes in the Galois field. Its main contents are as follows: first, exploiting the decomposition technique based on the three-variables check equation, we formulate the maximum likelihood (ML) decoding problem approximately to a non-convex quadratic program; second, an efficient algorithm based on the proximal-ADMM technique is proposed to solve the formulated QP problem. Exploiting the QP problem’s inherent structures, its variables can be updated in parallel; third, we prove that the proposed decoding algorithm can converge to some stationary point of the formulated QP problem. Moreover, we also show, for nonbinary low-density parity-check (LDPC) codes, its computational complexity in each proximal-ADMM iteration scales linearly with block length and the size of the considered Galois field. Simulation results demonstrate that the proposed proximal-ADMM decoder outperforms state-of-the-art nonbinary LDPC decoders in terms of either error correction performance or computational complexity.

Index Terms—Nonbinary Low-density Parity-check (LDPC) codes, Galois Field, Proximal Alternating Direction Method of Multipliers (Proximal-ADMM), Quadratic Programming (QP).

I. INTRODUCTION

Nonbinary linear block codes, such as nonbinary low-density parity-check (LDPC) codes [1], are favorable in high-data-rate communication systems and storage systems [2]–[4] since they possess many desirable merits from the viewpoints of practical applications. For example, nonbinary LDPC codes have greater ability to eliminate short cycles (especially 4-cycles) and display better error correction performance [5]. Moreover, nonbinary LDPC codes have good ability to resist burst errors by combining multiple burst bit errors into fewer nonbinary symbol errors [6]. Furthermore, nonbinary LDPC codes can provide a higher data transmission rate and spectral efficiency when combined with a higher-order modulation scheme [7].

Typical decoding algorithms for nonbinary LDPC codes, such as sum-product [1] [8], are based on the belief propagation (BP) strategy. In [1], authors first investigated nonbinary LDPC codes and the corresponding nonbinary BP algorithm. Moreover, authors in [8] optimized nonbinary BP algorithms by reducing the computational complexity of the check-node processing. However, nonbinary BP-like decoding algorithms are heuristic from a theoretical viewpoint since their theoretical performance, such as convergence, cannot be guaranteed. Meanwhile, analyzing the behavior of the nonbinary BP-like decoding algorithms is often difficult and the corresponding results are very limited.

In recent years, mathematical programming (MP) techniques, such as linear programming (LP) and quadratic programming (QP), are proposed to decode LDPC codes. Due to their theoretically-guaranteed decoding performance, MP decoding techniques have attracted significant attention from researchers in the error correction coding/decoding field. The first MP decoding technique was proposed by Feldman et al. [9], who relaxed the maximum-likelihood (ML) decoding problem to a linear program for binary LDPC codes. In comparison with the classical BP decoder, two issues must be considered: general LP solving algorithms’ complexity, such as interior point method [13] and simplex method [14], are prohibitive in practical applications and the other is its inferior error correction performance in low SNR regions. For the first issue, many new MP decoders for binary LDPC codes are proposed. In [15], authors applied the alternating direction method of the multipliers (ADMM) technique to solve the original LP decoding problem [9]. Its main concern is that each ADMM iteration involves expensive check-polytope projection operations. Later, authors in [16] and [17] independently optimized the check-polytope projection algorithm. Based on the works in [16] and [17], authors in [18] proposed a hardware-compatible projection algorithm. In [19], authors proposed an iterative check-polytope projection algorithm to reduce the complexity of LP decoding. Moreover, authors in [20] investigated a projection reduction technique to reduce the number of Euclidean projections onto the check polytope. Authors in [21] proposed an efficient ADMM-based LP decoding algorithm via the three-variables check-equations decomposition technique. For the second issue, authors in [22] proposed an adaptive LP decoder to improve LP decoding performance by adaptively adding necessary parity-check constraints. Authors in [23]–[25] designed different cut-generating algorithms to eliminate unexpected pseudo-codewords and improve the error correction performance of LP decoding. In addition, authors in [26] and [27] independently proposed improved ADMM-based penalized decoding algorithms to enhance the error correction performance of LP decoding in low SNR regions.

In comparison with MP decoding techniques for binary LDPC codes, the corresponding approaches for nonbinary cases are limited. Of particular relevance is the work [28] where LP decoding was first generalized to nonbinary LDPC codes. However, nonbinary LP decoding encounters a similar computational complexity problem as the binary case when using general LP solvers. To overcome this problem, authors in [29] and [30] independently extended works in [31] and
and applied the coordinate ascent method to solve an approximated dual problem of the original LP decoding problem. Authors in [35] were inspired by the binary LP decoding idea of [34] and proposed a trellis-based algorithm for check node processing to reduce the complexity of nonbinary LP decoding. In addition, another nonbinary LP decoding scheme was introduced in [35] by using constant-weight binary vectors to represent elements in the Galois field of characteristic two but no efficient algorithm was developed to solve the resulting LP problem. More recently, authors in [36, 37] extended the binary LP decoding idea of [35] to the nonbinary case and developed an LP decoding algorithm based on the ADMM technique for nonbinary LDPC codes in the Galois field of characteristic two. However, the proposed nonbinary ADMM decoder involves time-consuming Euclidean projections onto check polytopes.

In this paper, we focus on designing a convergence-guaranteed decoder with low complexity and favourable error correction performance for nonbinary LDPC codes. Specifically, the main contributions of this paper are summarized as follows.

- Based on the check-equation decomposition method, we first decompose a general multi-variables check equation into a set of three-variables check equations. Then, by exploiting the equivalent binary parity-check formulation of every three-variables check equation, we transform the nonbinary ML decoding problem to an equivalent linear integer program (LIP). Finally, by adding the quadratic penalty term into the objective of the LIP model, relaxing binary constraints to box constraints, and introducing extra linear constraints, a new quadratic programming (QP) decoding model is established for nonbinary linear block codes. The rest of this paper is organized as follows. In Section II, we briefly introduce the formulation of the ML decoding problem for nonbinary linear block codes. In Section III, we present the equivalent binary parity-check formulation of the three-variables check equation and establish a relaxed QP decoding problem for nonbinary linear block codes. Moreover, an efficient proximal-ADMM algorithm for solving the formulated QP problem is presented in Section IV. Section V shows the convergence and complexity analyses of the proposed proximal-ADMM decoding algorithm. Simulation results demonstrate the effectiveness of our proposed QP decoder in Section VI. Finally, Section VII concludes this paper.

Notations: in this paper, we just focus on LDPC codes in the Galois field of characteristic two, denoted by \( \mathbb{F}_{2^q} \). \( \mathbb{R} \) represents real numbers. Bold lowercase and uppercase letters denote vectors and matrices respectively. \((\cdot)^T\) and \(\|\cdot\|_2\) symbolize the transpose operation and the 2-norm operator respectively. \(\text{diag}(\cdot)\) indicates the operator of element diagonalization. Let \(\delta_A\) denote the spectral norm of matrix \(A\). The Euclidean projection operator onto a set \(X\) is denoted by \(\Pi_X\). \(\nabla_x f\) is the derivative of function \(f\) with respect to variable \(x\). \(\lambda_{\text{min}}(A^T A)\) represents the minimum eigenvalue of matrix \(A^T A\).

### II. ML Decoding Problem Formulation

Consider a nonbinary linear block codeword defined by an \(m\)-by-\(n\) check matrix \(H\). Its feasible code set is denoted by \(C \subseteq \{0, 1, \cdots, 2^q - 1\}^n\). All the elements in \(H\) belong to set \(\{0, 1, \cdots, 2^q - 1\}\). Let \(h_j^T, j \in J = \{1, 2, \cdots, m\}\), denote the \(j\)th row vector of the check matrix \(H\).

Assume that codeword \(u\) is transmitted through an additional white Gaussian noise (AWGN) channel and its corresponding output is denoted as \(r\). In the receiver, the aim of ML decoding is to determine which codeword has the largest a priori probability \(p(r|u)\) throughout the feasible codeword set \(C\). So the ML decoding problem can be formulated as

\[
\mathbf{x}^* = \arg\max_{\mathbf{x}\in C} p(r|\mathbf{u}), \tag{1}
\]

where codeword set \(C\) can be expressed as

\[
C = \left\{ u | (h_j^T u) \oplus \mathbf{2}^q = 0, j \in J, u \in \{0, 1, \cdots, 2^q - 1\}^n \right\}. \tag{2}
\]

According to the mapping rule [3], any nonzero element \(u_i \in u\) can be one-to-one correspondent to a \((2^q - 1)\)-length binary vector \(x_i = [x_{i,1}, \cdots, x_{i,\sigma}, \cdots, x_{i,2^q-1}]^T\), where

\[
x_{i,\sigma} = \begin{cases} 1, & \sigma = u_i, \\ 0, & \sigma \neq u_i. \end{cases} \tag{3}
\]

Besides, we map the zero element to a \((2^q - 1)\)-length all-zeros vector. Then, codeword \(u\) can be mapped one-to-one correspondent to a binary vector \(x = [x_1^T, \cdots, x_m^T]^T \in \{0, 1\}^{m(2^q - 1)}\). Here, we call \(x\) as the equivalent binary codeword to the nonbinary codeword \(u\). Let \(X\) denote the set consisting of all the equivalent binary codewords. Then, the ML decoding problem [11] is equivalent to

\[
\mathbf{x}^* = \arg\max_{\mathbf{x}\in X} p(r|x), \tag{4}
\]

which can be further derived as

\[
\arg\max_{\mathbf{x}\in X} p(r|x) = \arg\max_{\mathbf{x}\in X} \prod_{i=1}^m \prod_{\sigma=1}^{2^q-1} p(r_i|x_{i,\sigma}) = \arg\min_{\mathbf{x}\in X} \sum_{i=1}^m \sum_{\sigma=1}^{2^q-1} -p(r_i|x_{i,\sigma}). \tag{5}
\]

Plugging constant \(\sum_{i=1}^m \sum_{\sigma=1}^{2^q-1} \log p(r_i|x_{i,\sigma} = 0)\) into [5], we have
the following derivations

\[
\arg \max_{x \in \mathcal{X}} p(x|x) = \arg \min_{x \in \mathcal{X}} \sum_{i=1}^{n} \sum_{\sigma=1}^{2^q-1} \log \frac{p(r_i|x_i,\sigma) = 0}{p(r_i|x_i,\sigma) = 1}
\]

\[
= \arg \min_{x \in \mathcal{X}} \sum_{i=1}^{n} x_i \sigma \log \frac{p(r_i|x_i,\sigma) = 0}{p(r_i|x_i,\sigma) = 1}
\]

(6)

where \( \gamma \in \mathbb{R}^{n(2^q-1)} \times \mathcal{X} \) is called a cost vector defined by

\[
\gamma = \left[ \log \frac{p(r_1|x_{1,1} = 0)}{p(r_1|x_{1,1} = 1)}, \cdots, \log \frac{p(r_1|x_{1,2^q-1} = 0)}{p(r_1|x_{1,2^q-1} = 1)}, \right.
\]

\[
\cdots, \log \frac{p(r_n|x_{n,1} = 0)}{p(r_n|x_{n,2^q-1} = 0)}, \cdots, \log \frac{p(r_n|x_{n,2^q-1} = 0)}{p(r_n|x_{n,2^q-1} = 1)} \right].
\]

Then, the ML decoding problem can be cast as the following standard optimization model

\[
\min_{x} \quad \gamma^{T} x,
\]

s.t. \( x \in \mathcal{X} \).

Problem (8) is a binary integer program. The difficulty of solving it lies in how to formulate and handle the constraint \( x \in \mathcal{X} \). In the following sections, we first decompose check equations in (2) into a series of three-variables check equations and second transform the resulting constraints to binary constraints and show analytical expressions for \( x \in \mathcal{X} \), which leads to a nonconvex quadratic continuous optimization problem. Moreover, an efficient algorithm, named as proximal-ADMM, is proposed to solve the formulated optimization problem. The analyses on convergence and computational complexity of the designed proximal-ADMM algorithm are also presented.

### III. PROBLEM RELAXATION

#### A. Three-variables check equation and its equivalent binary parity-check formulation

First, we consider the following three-variables \( 2^q \)-ary check equation in \( \mathbb{F}_{2^q} \)

\[
\left( \sum_{k=1}^{3} h_{k} u_{k} \right) \oplus 2^q = 0,
\]

(9)

where \( h_{k} \) is some nonzero constant, \( u_{k} \) is a variable and the addition “\( \oplus \)” and multiplication are in \( \mathbb{F}_{2^q} \). Since \( h_{k} u_{k} \) can be expressed exactly as

\[
h_{k} u_{k} = \sum_{i=1}^{q} c_{i,k} 2^{i-1},
\]

(10)

where \( c_{i,k} \in \{0,1\} \), then the \( 2^q \)-ary check equation (9) is equivalent to the following \( q \) three-variables parity-check equations

\[
\left( \sum_{k=1}^{3} c_{i,k} \right) \oplus 2 = 0, \quad i = 1, 2, \cdots, q.
\]

(11)

Moreover, the above parity-check equations can be further equivalent to

\[
c_{i,1} \leq c_{i,2} + c_{i,3}, \quad c_{i,2} \leq c_{i,1} + c_{i,3},
\]

\[
c_{i,3} \leq c_{i,1} + c_{i,2}, \quad c_{i,1} + c_{i,2} + c_{i,3} \leq 2,
\]

(12)

Letting

\[
t = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},
\]

(13)

we can rewrite (12) as

\[
P c_i \leq t, \quad c_i \in \{0,1\}^q, \quad i = 1, 2, \cdots, q.
\]

(14)

where \( c_i = [c_{i,1}, c_{i,2}, c_{i,3}] \).

For any nonzero constant \( h_{k} \in \mathbb{F}_{2^q} \), it can be mapped one-to-one to a \( (2^q-1) \times (2^q-1) \) matrix, denoted by \( D(2^q, h_{k}) \), whose entries are determined by

\[
D(2^q, h_{k})_{ij} = \begin{cases} 1, & \text{if } i = j h_{k} \oplus 2^q, \\ 0, & \text{otherwise}. \end{cases}
\]

(15)

Then, we have the following lemma.

**Lemma 1**: For \( u_{k} \in \mathbb{F}_{2^q} \), let \( x_{k} \) be its corresponding binary vector codeword according to mapping rule (3). Then, \( D(2^q, h_{k}) x_{k} \) is the corresponding binary equivalent codeword for \( h_{k} u_{k} \).

**Proof** See proof in Appendix A

Moreover, we map any nonzero element \( \alpha \in \mathbb{F}_{2^q} \) to a \( q \)-length binary vector according to rule (10) and formulate the following \( q \)-by-(\( 2^q - 1 \)) matrix

\[
B = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}.
\]

(16)

It is easy to see that \( \alpha \) equals \( B \)’s column index. Letting \( b_i^T \) denote the \( i \)th row vector of matrix \( B \), then we have

\[
c_{i,k} = b_i^T D(2^q, h_{k}) x_{k}, \quad i = 1, 2, \ldots, q, \quad k = 1, 2, 3.
\]

(17)

Plugging it into (14), we can rewrite it as

\[
P T_i D x \leq t, \quad \forall i = 1, 2, \ldots, q,
\]

(18)

where

\[
T_i = \text{diag} (b_i^T, b_i^T, b_i^T),
\]

\[
D = \text{diag} (D(2^q, h_1), D(2^q, h_2), D(2^q, h_3)),
\]

\[
x = [x_1; x_2; x_3].
\]

(19a

(19b

(19c

Moreover, let

\[
W = [P T_1 D; \ldots; P T_q D],
\]

\[
w = 1 \otimes t,
\]

(20a

(20b

where “1” is a length-\( q \) all-ones vector. Then, based on (18), the three-variables parity-check equations (11) can be
equivalent to
\[ Wx \preceq w, \; Ex \preceq 1, \; x \in \{0, 1\}^{3(2^q - 1)}, \quad (21) \]
in the sense their solutions are one-to-one correspondent, where \( 1 = [1; 1; 1] \) and \( E = \text{diag}(1^T, 1^T, 1^T) \) in which “1” is a length-(2^q - 1) all-ones vector.

B. Equivalent ML decoding problem

Consider the \( j \text{th} \) check equation in (2). Without loss of generality, we assume it involves \( d_j \geq 3 \) variables, which are denoted by \( u_{\sigma_1}, \ldots, u_{\sigma_{d_j}} \) and their corresponding coefficients are \( h_{\sigma_1}, \ldots, h_{\sigma_{d_j}} \). Then, the check equation can be decomposed equivalently to \( d_j - 2 \) three-variables check equations by introducing \( d_j - 3 \) auxiliary variables. The detailed decomposing procedure is presented as the following three steps:

Step 1: for the first two variables \( u_{\sigma_1} \) and \( u_{\sigma_2} \), we introduce an auxiliary variable \( v_1 \) and let them satisfy the three-variables check equation (5), i.e.,
\[ (h_{\sigma_1}u_{\sigma_1} + h_{\sigma_2}u_{\sigma_2} + g_1) \oplus 2^q = 0. \quad (22) \]

Step 2: for the variables in set \( \{u_{\sigma_3}, \ldots, u_{\sigma_{d_j-2}}\} \), we introduce a corresponding auxiliary variables set \( \{g_2, \ldots, g_{d_j-3}\} \) and let them satisfy the following three-variables check equations
\[ (g_{t-1} + h_{\sigma_{t+1}}u_{\sigma_{t+1}} + g_t) \oplus 2^q = 0, \quad t = 2, \ldots, d_j - 4. \quad (23) \]

Step 3: let auxiliary variable \( g_{d_j-3} \) and the last two variables \( u_{\sigma_{d_j-1}} \) and \( u_{\sigma_{d_j}} \) satisfy
\[ (g_{d_j-3} + h_{\sigma_{d_j-1}}u_{\sigma_{d_j-1}} + h_{\sigma_{d_j}}u_{\sigma_{d_j}}) \oplus 2^q = 0. \quad (24) \]

Applying the above decomposing procedure to all the check equations in (2), one can find the total numbers of the three-variables check equations and the introduced 2^q-ary auxiliary variables are
\[ \Gamma_c = \sum_{j=1}^{m}(d_j - 2), \quad \Gamma_a = \sum_{j=1}^{m}(d_j - 3), \quad (25) \]
respectively.

Based on the discussion of the three-variables check equation in the previous subsection, we define
\[ v = [x; s], \quad (26) \]
where auxiliary variable \( s = [s_1; \ldots ; s_i; \ldots ; s_{\Gamma_a}] \) and \( s_i \in \{0, 1\}^{2^q - 1} \) correspond to auxiliary variable \( g_1 \). Define a variable-selecting matrix \( Q_\tau \in \{0, 1\}^{3 \times (n+1)} \), corresponding to the \( \tau \text{th} \) three-variables check equation. Its every row includes only one “1”, whose index corresponds to the variable in the check equation. It is easy to see that \( Q_\tau v \) are the variables in the \( \tau \text{th} \) three-variables check equation, where \( \tau = 1, \ldots, \Gamma_c \). Moreover, we define
\[ \lambda = [\gamma; 0], \quad (27a) \]
\[ F = [W_1(Q_1 \otimes I); \ldots; W_\tau(Q_\tau \otimes I); \ldots; W_{\Gamma_c}(Q_{\Gamma_c} \otimes I)], \quad (27b) \]
\[ f = 1 \otimes w. \quad (27c) \]

where \( I \) is a \((2^q-1) \times (2^q-1)\) identity matrix, symbols “1” and “0” are length-\( \Gamma_c \) all-ones vector and length-(2^q - 1)\( \Gamma_a \) all-zeros vector, respectively, and \( W_\tau = [PT_1; \ldots; PT_{d_j}] \) and \( D_\tau \) have the same expression as [19]. Another observation on the equivalent binary codeword of the nonbinary symbol is that it includes at most one 1 (see [1]). To exploit this structure, we define \( (n+\Gamma_a) \times (n+\Gamma_a)(2^q - 1) \) matrix \( S = \text{diag}(1^T, \ldots, 1^T) \) where “1” is a length-(2^q - 1) all-ones vector. Then, we have
\[ Sv \preceq 1. \quad (28) \]
where “1” is a length-\( n+\Gamma_a \) all-ones vector. Then, the ML decoding problem (9) is equivalent to the following linear integer program
\[ \min_v \lambda^T v \quad (29a) \]
\[ \text{s.t. } Fv \preceq f, \; Sv \preceq 1, \quad (29b) \]
\[ v \in \{0, 1\}^{(2^q-1)(n+\Gamma_a)}. \quad (29c) \]

Due to the binary constraints (29c), the above linear integral program (29) is NP-hard, i.e., its computational complexity scales exponentially with the number of variables. Therefore, it is prohibitive to solve problem (29) directly. In the following, we exploit relaxation and tightness techniques to formulate a tractable model.

C. Relaxation and Tightness

The typical way to handle the binary constraint (29c) is to relax it to the box constraint \( 0 \leq v \leq 1 \), which can simplify the NP-hard problem (29) to a convex one. However, the resulting optimization problem’s optimal solution could be fractional especially when the decoder works in low SNR regions. To overcome this drawback, we deploy the following two techniques to tighten the relaxation.

One is to add a quadratic penalty term into the objective, i.e., \( \lambda^T v - \frac{\alpha}{2} ||v - 0.5||^2_2 \), where \( \alpha > 0 \) is a preset constant. Intuitively, the quadratic penalty can make the optimal integer solutions more favorable.

The other is to introduce extra linear constraints to cut possible fractional solutions from feasible space. For (11), defining \( c_k = [c_{1,k}, \ldots, c_{q,k}]^T \), we can rewrite it as \( \sum_{k=1}^{3} c_k \oplus 2 = 0 \). Letting \( B^T \) (see (16)) multiply the left side of the above equation, we can obtain
\[ \left(B^T \sum_{k=1}^{3} c_k \right) \oplus 2 = 0, \quad (30) \]
where “0” is a \((2^q - 1)\)-length all-zeros vector. Similar derivations to (12–21), we can obtain
\[ Wx \preceq w, \; x \in \{0, 1\}^{3(2^q-1)}, \quad (31) \]
where
\[ w = 1 \otimes t, \quad (32a) \]
\[ W = (B^T \otimes I)W \otimes 2. \quad (32b) \]
where “1” is a length-2^q - 1 all-ones vector and “T” is a \( 4 \times 4 \) identity matrix. Since \( w \) is a \((2^q - 1)\)-length vector and \( W \) is a \((2^q - 1)\)-by-(2^q - 1) matrix, one can find that \( (31) \) consists of \( 4(2^q - 1) \) inequalities. Moreover, besides \( 4q \) inequalities in (31), other inequalities in (31) can be cast as redundant
ones since they are combined by some inequalities in (21). However, when the binary constraint \( x \in \{0, 1\}^{3(2^n-1)} \) is relaxed to the box constraint \( x \in [0, 1]^{3(2^n-1)} \), these redundant inequalities can play a role in tightening the relaxation.

Based on the above tightening techniques and through a similar formulation procedure (22)-(29), the MP decoding problem (29) can be relaxed to the following optimization model

\[
\min_v \lambda^T v - \frac{\alpha}{2} \|v - 0.5\|_2^2, \\
\text{s.t.} \quad Av \preceq b, \\
0 \preceq v \preceq 1,
\]

where symbols “1” and “0” in (33a) are length-(2\(^n\)-1)(n+\(\Gamma\)) 
all-ones vector and length-(2\(^n\)-1)(n + \(\Gamma\)) all-zeros vector, respectively, and

\[
A = [\hat{W}_1(Q_1 \circ I) I \cdots \hat{W}_n(Q_n \circ I) I \cdots \hat{W}_{\Gamma_n}(Q_{\Gamma_n} \circ I) I], \quad b = [1 \circ \hat{w}_1]. \tag{34a}
\]

where symbols “1”s in (34a) are length-\(\Gamma\)c and length-(n+\(\Gamma\)) 
all-ones vectors, respectively.

In sequel, we will present an efficient solving algorithm via the proximal-ADMM technique for the above optimization problem (33). Moreover, we also prove that the proposed proximal-ADMM decoding algorithm converges to some stationary point of problem (33) in theory. Furthermore, by exploiting inherent structures of problem (33), we show that the computational complexity of the proposed proximal-ADMM algorithm is linear to the length of the LDPC codes.

IV. PROXIMAL-ADMM SOLVING ALGORITHM

A. Proximal-ADMM algorithm framework

By introducing two auxiliary variables, \( e_1 \) and \( e_2 \), we transform the decoding problem (33) to

\[
\min_{v, e_1, e_2} \lambda^T v - \frac{\alpha}{2} \|v - 0.5\|_2^2, \\
\text{s.t.} \quad Av + e_1 = b, \quad e_1 \succeq 0, \\
v = e_2, \quad 0 \preceq e_2 \preceq 1,
\]

where symbol “0” in (35b) is a length-(4(2\(^n\)-1)\(\Gamma\) + n + \(\Gamma\)) 
all-zeros vector, and symbols “1” and “0” in (35c) are length-(2\(^n\)-1)(n + \(\Gamma\)) 
all-ones vector and length-(2\(^n\)-1)(n + \(\Gamma\)) all-zeros vector, respectively. The augmented Lagrangian function for problem (35) can be written as

\[
\mathcal{L}_\mu(v, e_1, e_2, y_1, y_2) = \lambda^T v - \frac{\alpha}{2} \|v - 0.5\|_2^2 + y_1^T (Av + e_1 - b) \\
+ y_2^T (v - e_2) + \frac{\mu}{2} \|Av + e_1 - b\|_2^2 + \frac{\mu}{2} \|v - e_2\|_2^2, \tag{36}
\]

where \( y_1 \) and \( y_2 \) are Lagrangian multipliers corresponding to the two equality constraints in (35) respectively and \( \mu > 0 \) is a penalty parameter. Based on (36), the proximal-ADMM iteration algorithm for solving (35) can be described as follows

\[
v^{k+1} = \arg \min_v \mathcal{L}_\mu(v, e_1^k, e_2^k, y_1^k, y_2^k) + \frac{\mu}{2} \|v - p^k\|_2^2, \tag{37a}
\]

\[
e_1^{k+1} = \arg \min_{e_1 \succeq 0} \mathcal{L}_\mu(v^{k+1}, e_1, e_2^k, y_1^k, y_2^k) + \frac{\mu}{2} \|e_1 - z_1^k\|_2^2, \tag{37b}
\]

\[
e_2^{k+1} = \arg \min_{0 \preceq e_2 \preceq 1} \mathcal{L}_\mu(v^{k+1}, e_1^k, e_2, y_1^k, y_2^k) + \frac{\mu}{2} \|e_2 - z_2^k\|_2^2, \tag{37c}
\]

where \( k \) is the iteration number, \( \|v - p^k\|_2^2, \|e_1 - z_1^k\|_2^2 \), and \( \|e_2 - z_2^k\|_2^2 \) are so-called proximal terms, \( \mu \) is the corresponding penalty parameter, and \( \beta \) belongs to (0, 1). In the following, we show that subproblems (37a)-(37c) can be solved efficiently by exploiting their inherent structures.

B. Solving subproblem (37a)

Obviously, choosing their values of parameters \( \rho \), \( \mu \), and \( \alpha \) properly, problem (37a) can be reduced to a strongly quadratic convex one with respect to \( v \). In this case, its solving procedure can be described as follows: by setting the gradient of the function \( \mathcal{L}_\mu(v, u_1, u_2, y_1, y_2) + \frac{\mu}{2} \|v - p^k\|_2^2 \) to be zero and solving the corresponding linear equation, we update \( v \) as

\[
v^{k+1} = (A^T A + \epsilon I)^{-1} \varphi^k \tag{38}
\]

where “\( T \) is a (2\(^n\)-1)(n+\(\Gamma\)) \times (2\(^n\)-1)(n+\(\Gamma\)) identity matrix, and

\[
\epsilon = 1 + \frac{\rho}{\mu} - \frac{\alpha}{\mu}, \\
\varphi^k = A^T (b - e_1^k - \frac{y_1^k}{\mu}) + e_2^k - \frac{y_2^k}{\mu} \frac{\rho}{\mu} - \frac{\lambda + 0.5\alpha}{\mu}. \tag{39}
\]

Note that \((A^T A + \epsilon I)^{-1}\) is fixed for a given code. Thus, it only needs to be calculated only once throughout the ADMM iterations. Therefore, the main computational cost lies in \((A^T A + \epsilon I)^{-1}\)\( \varphi^k \). It is easy to see that calculating it directly requires \(O((2^n-1)^2(n+\Gamma)^2)\) complexity, which is prohibitive for large-scale problems. In the following, we show a much more efficient way to perform the computational procedure. Firstly, we present the following lemma.

**Lemma 2**: Matrix \((A^T A + \epsilon I)^{-1}\) is a block diagonal. Specifically, it can be denoted by

\[
(A^T A + \epsilon I)^{-1} = \text{diag}\left((A_i^T A_i + \epsilon I)^{-1}, \ldots, (A_n^T A_n + \Gamma_n + \epsilon I)^{-1}\right), \tag{40}
\]

where sub-matrix \( A_i, i = 1, \ldots, n + \Gamma_n \), is formed by column vectors indexed from \(((2^n-1)(i-1)+1)\) to \(((2^n-1)i)\) in matrix \( A \) and “\( T \)” is a \((2^n-1) \times (2^n-1)\) identity matrix.
Moreover,

\[
(A_i^T A_i + \epsilon I)^{-1} = \begin{bmatrix}
\theta_i & \omega_i & \cdots & \omega_i \\
\omega_i & \theta_i & \cdots & \omega_i \\
\vdots & \vdots & \ddots & \vdots \\
\omega_i & \cdots & \cdots & \theta_i
\end{bmatrix},
\]

(41)

where

\[
\theta_i = \omega_i + \frac{1}{2m_d + \epsilon}, \\
\omega_i = \frac{-2m_d}{(2m_d + \epsilon)[(2m_d + \epsilon + 1) + (2m_d + 1)(2m_d - 2)]}. 
\]

(42)

Here, \(d_i\) denotes the degree of the \(i\)th information symbol, i.e., the number of check equations which it participates in.

**Proof** See Appendix [B]

(40) indicates \(\phi^{k+1}\) in (38) can be obtained through the following implementations in parallel

\[
v_i^{k+1} = (A_i^T A_i + \epsilon I)^{-1} \phi^k_i, \quad i = 1, 2, \ldots, n + \Gamma_a, 
\]

(43)

where \(\phi^k_i\) is the corresponding \(2^n - 1\)-length sub-vector in \(\phi^k\). Moreover, based on (41), (43) can be further derived as follows

\[
v_i^{k+1} = \begin{bmatrix}
\omega_i & \omega_i & \cdots & \omega_i \\
0 & \omega_i & \cdots & \omega_i \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & \omega_i
\end{bmatrix} \phi^k_i + \begin{bmatrix}
\theta_i - \omega_i & 0 & \cdots & 0 \\
0 & \theta_i - \omega_i & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & \theta_i - \omega_i
\end{bmatrix} k_i + 1, i = 1, 2, \ldots, n + \Gamma_a, 
\]

(44)

where \(\phi^k_{i,a}\) denotes the \(i\)th entry in \(\phi^k\).

### C. Solving subproblems (37b) and (37c)

Solving (37b) is equivalent to solving the following M subproblems in parallel

\[
\min_{e_{1,j}} y_{1,j}^k e_{1,j} + \frac{\mu}{2} (a_j^T v^{k+1} + c_{1,j} - b_j)^2 + \frac{\rho}{2} (e_{1,j} - z_{1,j})^2 \\
\text{s.t.} \quad e_{1,j} \geq 0, \quad j = 1, \ldots, M, 
\]

(45)

where \(M = 4(2^n - 1) \Gamma_a + n + \Gamma_a\), \(a_j^T\) denotes the \(j\)th row vector of matrix \(A\). Obviously, the optimal solution of the above problem (45) can be obtained by setting the gradient of its objective function to zero and then projecting the solution of the corresponding equation to region \([0, +\infty]\).

Then, we can obtain

\[
e_{1,j}^{k+1} = \Pi_{[0, +\infty)} \frac{\mu}{\rho + \mu} \left(b_j - a_j^T v^{k+1} - \frac{y_{1,j}^k}{\mu} + \frac{\rho}{\mu} z_{1,j}^k\right).
\]

(46)

Similar to (37b), problem (37c) can be separated into the following \(N\) independent subproblems

\[
\min_{e_{2,\ell}, z_{2,\ell}} y_{2,\ell}^k e_{2,\ell} + \frac{\mu}{2} (v_{2,\ell}^k - e_{2,\ell})^2 + \frac{\rho}{2} (e_{2,\ell} - z_{2,\ell})^2, \\
\text{s.t.} \quad 0 \leq e_{2,\ell} \leq 1, \quad \ell = 1, \ldots, N, 
\]

(47)

where \(N = 2^n - 1 \Gamma_a + n + \Gamma_a\). Their optimal solutions can be expressed as

\[
e_{2,\ell}^{k+1} = \Pi_{[0, +\infty)} \frac{\mu}{\rho + \mu} \left(b_j - a_j^T v^{k+1} - \frac{y_{2,\ell}^k}{\mu} + \frac{\rho}{\mu} z_{2,\ell}^k\right).
\]

(48)

In Algorithm 7, we summarize the proposed proximal-ADMM decoding algorithm for nonbinary LDPC codes in \(\mathbb{F}_{2^m}\). In the next section, we will discuss its convergence and computational complexity.

### V. Performance Analysis

#### A. Convergence

Before presenting the main convergence result, we have the following lemma to show that the gradient of the considered augmented Lagrangian \(L_\mu(\cdot)\) is Lipschitz continuous.

**Lemma 3:** Suppose \(\alpha > 0\) and \(\mu > 0\) and let \(X = \{v \mid Av \preceq b, 0 \preceq v \preceq \delta_\mu\}\). Then, the gradient of the augmented Lagrangian \(L_\mu\) with respect to variable \(v\) is Lipschitz continuous, i.e., for any \(v, v' \in X\),

\[
\|\nabla_v L_\mu(v, e_1, e_2, y_1, y_2) - \nabla_v L_\mu(v', e_1, e_2, y_1, y_2)\|_2 \leq L \|v - v'\|_2,
\]

(49)

where \(L \geq \alpha + \mu + \mu \delta_\mu^2\) and “\(\delta_A\)” denotes the spectral norm of matrix \(A\).

**Proof** See Appendix [C]

Based on Lemma 3, we have the following theorem to characterize the convergence property of the proposed proximal-ADMM decoding algorithm.

**Theorem 1:** Assume that \(\rho > \alpha > 0\) and \(\delta_{\min}(A - A_1) \leq \mu \leq \frac{\rho(\rho - \alpha)}{\delta_{\min}(A + A_1)}\), where \(\delta_{A_1}\) is the spectral norm of matrix \(A_1 = \begin{bmatrix} A & 1 \end{bmatrix}\) and symbol “\(I_\kappa\)” denotes a \(\kappa\)-by-\(\kappa\) identity matrix. Let \(\{v, e^k_1, e^k_2, p^k_1, z^k_1, z^k_2, y^k_1, y^k_2\}\) be the tuples generated by Algorithm 7. Then, we have the following
Moreover, observing (25), we have
\[ \Gamma \]
in every proximal-ADMM iteration is roughly
\[ d \]
since Algorithm conclude that the total computational complexity of (33), i.e.,
\[ \text{Algorithm 1} \]
and \[ v \]
and \[ p \]
are \[ O \]
and \[ \mu \]
respectively [39]. For \( C_1 \), we use the same parity-check matrix as the binary case but each binary nonzero check value is replaced by \( 1 \in \mathbb{F}_4 \). For \( C_2 \), its parity-check matrix is the same as the binary case with each nonzero check value \( 1 \in \mathbb{F}_{16} \). The corresponding modulations are quadrature phase shift keying (QPSK) and sixteen quadrature amplitude modulation (16QAM) respectively. The modulated symbols are transmitted over the additive white Gaussian noise (AWGN) channel. The considered decoders include the proposed proximal-ADMM algorithm, logarithm-domain fast-fourier-transforms-based Q-ary sum-product algorithm (Log-FFT-QSPA) [6], and the nonbinary ADMM-based LP (ADMM-LP) decoding algorithm [5]. The parameters of Algorithm [7] are set as follows: penalty parameter \( \mu \) is chosen as 0.8 and 0.6 for codes \( C_1 \) and \( C_2 \) respectively; parameters \( \alpha \), \( \rho \) and \( \beta \) are set to be 0.5, 0.52, and 0.9 respectively for both of the two codes; we stop the iteration when both \( \| a v^k + e^k_1 - b \|_2^2 \leq 10^{-5} \) and \( \| v^k - e^k_2 \|_2^2 \leq 10^{-5} \) are satisfied, or the maximum iteration number \( t_{\text{max}} = 500 \) is reached.

Fig. 1 shows the FER/SER performance of code \( C_1 \) in \( \mathbb{F}_4 \) and code \( C_2 \) in \( \mathbb{F}_{16} \) when different decoders are applied. From Fig. 1(a), it can be seen that our proposed decoder attains better FER/SER performance than Log-FFT-QSPA [8] and the nonbinary ADMM-LP decoder [5]. Observing Fig. 1(b), it can be seen that the proposed decoder outperforms the ADMM-LP decoder [5] in terms of either FER or SER. We can also see that the proposed decoder displays comparable FER/SER performance to the Log-FFT-QSPA in low SNR regions and performs superiorly to the Log-FFT-QSPA in terms of FER and BER at \( E_s/N_0 = 9dB \), where the FER/SER curves of the proposed proximal-ADMM decoder continue to drop in a waterfall manner while the FER/SER curves of the Log-FFT-QSPA drop slowly and the corresponding error-correction performance is inferior to the proximal-ADMM decoder. Therefore, we can conclude that the proposed proximal-ADMM decoder outperforms Log-FFT-QSPA [6] and the ADMM-LP decoder [5] in terms of error-correction performance for the considered nonbinary LDPC codes.

Fig. 2 shows the average iteration number of the proximal-ADMM decoder and the ADMM-LP decoder in [5]. From the figures, one can find that the proposed proximal-ADMM decoder requires fewer iterations than the ADMM-LP decoder [5] in low SNR regions and they are comparable in high SNR.

VI. Simulation Results

In this section, several numerical results are presented for the proposed proximal-ADMM decoder in Algorithm [7]. First, we show its error-correction performance (frame error rate (FER) and symbol error rate (SER)) and decoding efficiency, which is compared with several state-of-the-art nonbinary LDPC decoders. Second, we present how to select the proper value of the parameters in the proximal-ADMM decoder, which can improve the proximal-ADMM decoder’s error-correction performance and convergence rate.

A. Performance of the proposed proximal-ADMM decoder

We consider two codes, named \( C_1 \) and \( C_2 \), which are Tanner (1055,424) LDPC code and Tanner [155,64] LDPC code respectively [39]. For \( C_1 \), we use the same parity-check matrix as the binary case but each binary nonzero check value is replaced by \( 1 \in \mathbb{F}_4 \). For \( C_2 \), its parity-check matrix is the same as the binary case with each nonzero check value \( 1 \in \mathbb{F}_{16} \). The corresponding modulations are quadrature phase shift keying (QPSK) and sixteen quadrature amplitude modulation (16QAM) respectively. The modulated symbols are transmitted over the additive white Gaussian noise (AWGN) channel. The considered decoders include the proposed proximal-ADMM algorithm, logarithm-domain fast-fourier-transforms-based Q-ary sum-product algorithm (Log-FFT-QSPA) [6], and the nonbinary ADMM-based LP (ADMM-LP) decoding algorithm [5]. The parameters of Algorithm [7] are set as follows: penalty parameter \( \mu \) is chosen as 0.8 and 0.6 for codes \( C_1 \) and \( C_2 \) respectively; parameters \( \alpha \), \( \rho \) and \( \beta \) are set to be 0.5, 0.52, and 0.9 respectively for both of the two codes; we stop the iteration when both \( \| a v^k + e^k_1 - b \|_2^2 \leq 10^{-5} \) and \( \| v^k - e^k_2 \|_2^2 \leq 10^{-5} \) are satisfied, or the maximum iteration number \( t_{\text{max}} = 500 \) is reached.

Fig. 1 shows the FER/SER performance of code \( C_1 \) in \( \mathbb{F}_4 \) and code \( C_2 \) in \( \mathbb{F}_{16} \) when different decoders are applied. From Fig. 1(a), it can be seen that our proposed decoder attains better FER/SER performance than Log-FFT-QSPA [8] and the nonbinary ADMM-LP decoder [5]. Observing Fig. 1(b), it can be seen that the proposed decoder outperforms the ADMM-LP decoder [5] in terms of either FER or SER. We can also see that the proposed decoder displays comparable FER/SER performance to the Log-FFT-QSPA in low SNR regions and performs superiorly to the Log-FFT-QSPA in terms of FER and BER at \( E_s/N_0 = 9dB \), where the FER/SER curves of the proposed proximal-ADMM decoder continue to drop in a waterfall manner while the FER/SER curves of the Log-FFT-QSPA drop slowly and the corresponding error-correction performance is inferior to the proximal-ADMM decoder. Therefore, we can conclude that the proposed proximal-ADMM decoder outperforms Log-FFT-QSPA [6] and the ADMM-LP decoder [5] in terms of error-correction performance for the considered nonbinary LDPC codes.
present the FER performance and number of iterations for and then transmitted over the AWGN channel. Fig. 3 and Fig. 4 replaced by the simple Euclidean projections onto the positive Euclidean projections onto the check/simplex polytopes are codes. Here, we presented several simulation results for binary ADMM decoder is also suitable for decoding binary LDPC codes. Moreover, it can also be observed that the proposed proximal-ADMM decoder is less than 200 in high SNR regions for the nonbinary ADMM-LP decoder [36] in terms of decoding regions. Furthermore, compared with other competing binary SPAs at low SNRs, but achieves better error-correction performance in high SNR regions. Specifically, the SPA decoder displays error floor behavior while our proposed decoder continuously drops in a waterfall manner. Moreover, one can also find that the proposed proximal-ADMM decoding algorithm attains comparable FER performance to state-of-the-art binary ADMM-based penalized decoding (ADMM-PD) algorithms [26] [27] and outperforms the binary ADMM-LP decoding algorithm [21] in terms of error-correction performance. Fig. 4 shows that the average iteration numbers required to realize decoding are comparable between the proposed proximal-ADMM decoder and the ADMM-PD methods [26] [27]. At last, we should also note that the proposed proximal-ADMM decoder is more efficient than the ADMM-PD decoder [26] since it eliminates time-consuming check-polytope projections.

B. Parameter choices of the proposed proximal-ADMM decoder

There are several parameters in the proposed proximal-ADMM decoding Algorithm [7], including the penalty parameter $\mu$, the 2-norm penalty parameter $\alpha$, the ending tolerance $\xi$, the maximum number of iterations $t_{\text{max}}$, and parameters $\rho$ and $\beta$. Proper parameters can make Algorithm [7] achieve favourable error-correction performance and reduce the iteration number. However, an exhaustive search over all possible parameters is not practical and necessary. First, it is easy to see that fewer iterations are required when $\rho > \alpha$ and $\beta$ parameter is favorable because it can make the updates for variables $v$, $e_1$, and $e_2$ in every proximal-ADMM iteration not deviate too much from the stabilized iterate $p$, $z_1$, and $z_2$ respectively (cf. [41]). Therefore, we set $\beta$ to be 0.9 in the simulations. Moreover, Algorithm [7] is sensitive to the settings of parameters $\mu$, $\alpha$ and $\rho$. In order to guarantee that subproblem [37a] is strongly convex with respect to variable $x$, we let $\rho > \alpha$.

Next, we focus on how to choose parameters $\mu$ and $\alpha$. Fig. 5 and Fig. 6 plots FER performance and iteration numbers for code $C_3$ as a function of parameters $\mu$ and $\alpha$ at $E_b/N_0 = 5$ dB, respectively. In the figures, we set $\rho = \alpha + 0.2$ to ensure that $\rho > \alpha$ always holds. Observing Fig. 5, one can find that Algorithm [7] achieves better FER performance as long as $\mu \in [0.3, 1]$ and $\alpha \in [0.2, 0.5]$. Moreover, from Fig. 6 one can see that fewer iterations are required when $\mu \in [0.3, 1]$ and $\alpha \in [0.2, 0.5]$. This means that $\mu \in [0.3, 1]$ and $\alpha \in [0.2, 0.5]$ are good choices in terms of error-correction performance and decoding efficiency of Algorithm [7].

VII. CONCLUSION

In this paper, we propose an efficient proximal-ADMM decoding algorithm for nonbinary LDPC codes in $F_{2^q}$. We
prove that the proposed decoding algorithm is guaranteed to converge to some stationary point of the formulated QP decoding problem. Moreover, it is shown that the complexity of the proposed decoder in each iteration is roughly $O(2^{qn})$. Simulation results demonstrate that the proposed proximal-ADMM
decoding algorithm attains better error-correction performance and decoding efficiency than state-of-the-art nonbinary LDPC decoders.

APPENDIX A
PROOF OF LEMMA 1

Proof Let \( \mathbb{I}(\cdot) \) is an indicator function defined in \( \mathbb{F}_{2^n} \), i.e., \( \mathbb{I}(u, v) = 1 \) when \( u = v \), otherwise \( \mathbb{I}(u, v) = 0 \). Then, from the mapping rule defined in (3), binary vector codeword \( x_k \), corresponding to \( u_k \in \mathbb{F}_{2^n} \), can be expressed by

\[
x_k = \mathbb{I}(1, u_k) ; \ldots ; \mathbb{I}(j, u_k) ; \ldots ; \mathbb{I}(2^n - 1, u_k).
\]

Assuming \( j = u_k \), one can find that \( \mathbb{I}(j, u_k) = 1 \) and other elements in \( x_k \) are zeros. Moreover, based on the definition of \( D(2q, h_k)_{ij} \) in (15), matrix \( D(2^q, h_k) \) can be rewritten as

\[
D(2^q, h_k) = \begin{bmatrix}
\mathbb{I}(1, h_k) & \mathbb{I}(1, jh_k) & \mathbb{I}(1, (2^q - 1)h_k) \\
\vdots & \vdots & \vdots \\
\mathbb{I}(i, h_k) & \mathbb{I}(i, jh_k) & \mathbb{I}(i, (2^q - 1)h_k) \\
\mathbb{I}(2^q - 1, h_k) & \mathbb{I}(2^q - 1, jh_k) & \mathbb{I}(2^q - 1, (2^q - 1)h_k)
\end{bmatrix}.
\]

Then, we have

\[
D(2^q, h_k)x_k = \begin{bmatrix}
\mathbb{I}(1, jh_k)\mathbb{I}(j, u_k) \\
\vdots \\
\mathbb{I}(i, jh_k)\mathbb{I}(j, u_k) \\
\mathbb{I}(2^q - 1, jh_k)\mathbb{I}(j, u_k)
\end{bmatrix} = \begin{bmatrix}
0 \\
\vdots \\
0
\end{bmatrix}.
\]

Notice that \( \mathbb{I}(i, jh_k) = 1 \), which completes the proof.

APPENDIX B
PROOF OF LEMMA 2

Proof According to the definition of matrix \( A \) in (54), we have

\[
A^T A = \left( \sum_{\tau=1}^{\Gamma_c} (\hat{W}_\tau (Q_\tau \otimes I) )^T (\hat{W}_\tau (Q_\tau \otimes I) ) \right) + S^T S.
\]

Plugging (16) and (20a) into (32b), we can derive \( \hat{W}_\tau \) as follows

\[
\hat{W}_\tau = (B^T \otimes I) \hat{W}_\tau \oplus 2 = \left[ P \left( \sum_{i \in K_1} T_i \right) D_\tau ; \ldots ; P \left( \sum_{i \in K_{2^{n-1}}} T_i \right) D_\tau \right] \oplus 2,
\]

where \( K_1 = \{1\}, K_2 = \{2\}, K_3 = \{1, 2\}, \ldots, K_{2^n-1} = \{1, \ldots, q\} \). Then, we have the following derivations

\[
\hat{W}_\tau^T \hat{W}_\tau = \sum_{\ell=1}^{2^{n-1}} D_\tau^T \left( \left( \sum_{i \in K_{2^{n-1}}} T_i \right) \left( \sum_{i \in K_{2^{n-1}}} T_i \right) \right) D_\tau \\
= 4 \sum_{\ell=1}^{2^{n-1}} D_\tau^T \left( \left( \sum_{i \in K_{2^{n-1}}} T_i \right) \left( \sum_{i \in K_{2^{n-1}}} T_i \right) \right) D_\tau \\
= 4 D_\tau^T \left( \left( \sum_{i \in K_{2^{n-1}}} T_i \right) \left( \sum_{i \in K_{2^{n-1}}} T_i \right) \right) D_\tau,
\]

where the second equality holds since \( P^T P = 4I \). Moreover, since \( T_i = \text{diag}(b_i^T, b_i^T, b_i^T) \) (see (19)), we can define

\[
\Phi = \sum_{i \in K_{2^{n-1}}} \left( \sum_{i \in K_{2^{n-1}}} b_i \right) T_i \oplus 2,
\]

and derive (58) as

\[
\hat{W}_\tau^T \hat{W}_\tau = 4D_\tau^T \text{diag}(\Phi, \Phi, \Phi) D_\tau.
\]

Furthermore, since \( b_i^T \) is the \( i \)th row vector in matrix \( B \) (see (16), \( \Phi \) can be expressed as

\[
\Phi = \begin{bmatrix}
2q-1 & 2q-2 & \ldots & 2q-2 \\
2q-2 & 2q-1 & \ldots & 2q-2 \\
2q-2 & 2q-1 & \ldots & 2q-1
\end{bmatrix}.
\]

From (15), we can see that each row vector or column vector in matrix \( D(2^q, h_{rk}) \) only includes one “1”. Therefore, we can get

\[
D(2^q, h_{rk})^T \Phi D(2^q, h_{rk}) = \Phi,
\]

which implies that (60) can be further simplified to

\[
\hat{W}_\tau^T \hat{W}_\tau = 4 \text{diag}(\Phi, \Phi, \Phi).
\]

Plugging (63) into (56) and noting \( S = \text{diag}(1^T, \ldots, 1^T) \), we have the following derivations

\[
A^T A = \left( \sum_{\tau=1}^{\Gamma_c} (Q_\tau \otimes I) \right)^T \text{diag}(\Phi, \Phi, \Phi) (Q_\tau \otimes I) + S^T S
\]

\[
= \text{diag}(4d_i \Phi + 1^T 1, \ldots, 4d_n + \Gamma_c \Phi + 1^T 1),
\]

where \( d_i, i = 1, \ldots, n + \Gamma_c \), denotes the number of three-variables check equations that the \( i \)th information symbol participates in. From (64), one can conclude that matrix \( (A^T A + cI)^{-1} \) is block diagonal. Letting

\[
A_i^T A_i + cI = 4d_i \Phi + 1^T 1 + cI,
\]

(65)
we can obtain \( (A^T A + \epsilon I)^{-1} \), i.e.,
\[
(A^T A + \epsilon I)^{-1} = \text{diag}\left( (A^T_1 A_1 + \epsilon I)^{-1}, \ldots, (A^T_n A_n + \epsilon I)^{-1} \right).
\]
Plugging (61) into (65), we have
\[
A^T A + \epsilon I = \begin{bmatrix}
4d_2 2^{\alpha - 1} + 1 + \epsilon & 4d_2 2^{\beta - 1} + 1 + \epsilon & \cdots & 4d_2 2^{\beta - 1} + 1 + \epsilon \\
4d_2 2^{\alpha - 1} + 1 + \epsilon & 4d_2 2^{\beta - 1} + 1 + \epsilon & \cdots & 4d_2 2^{\beta - 1} + 1 + \epsilon \\
\vdots & \vdots & \ddots & \vdots \\
4d_2 2^{\alpha - 1} + 1 + \epsilon & 4d_2 2^{\beta - 1} + 1 + \epsilon & \cdots & 4d_2 2^{\beta - 1} + 1 + \epsilon
\end{bmatrix} \tag{66}
\]
Observing (66), it is easy to see that the diagonal elements in the matrix are the same and other elements are also the same. So its inverse matrix can be written as \( (A^T A + \epsilon I)^{-1} = \)
\[
\begin{bmatrix}
\theta_1 & \omega_1 & \cdots & \omega_1 \\
\omega_1 & \theta_1 & \cdots & \omega_1 \\
\vdots & \vdots & \ddots & \vdots \\
\omega_1 & \omega_1 & \cdots & \theta_1
\end{bmatrix} \tag{67}
\]
Multiplying the right sides of (65) and (67), the corresponding equality leads to the following equations
\[
(4d_2 2^{\alpha - 1} + 1 + \epsilon) \theta_i + (4d_2 2^{\beta - 1} + 1)(2^\gamma - 2) \omega_i = 1,
\]
\[
(4d_2 2^{\alpha - 1} + 1 + \epsilon) \omega_i + (4d_2 2^{\beta - 1} + 1)(2^\gamma - 2) \theta_i + (4d_2 2^{\beta - 1} + 1)(2^\gamma - 3) \omega_i = 0.
\]
Solving the above equations, we obtain
\[
\theta_i = \omega_i + \frac{1}{2^\gamma d_i + 1},
\]
\[
\omega_i = \frac{2^\gamma d_i}{(2^\gamma d_i + 1)(2^\gamma d_i + 1 + 1) + (2^\gamma d_i + 1)(2^\gamma - 2)}. \]
This completes the proof.

**APPENDIX C**

**Proof of Lemma 3**

Based on the definition of \( L_\mu \) in (56), its gradient, with respect to variable \( v \), can be calculated by
\[
\nabla_v L_\mu(v, e_1, e_2, y_1, y_2) = (\lambda - \alpha(v - 0.5)) + A^T y_1 + y_2 + \mu A^T(Av + e_1 - b) + \mu(v - e_2).
\]
Then, we have the following derivations
\[
||\nabla_v L_\mu(v, e_1, e_2, y_1, y_2) - \nabla_v L_\mu(v', e_1, e_2, y_1, y_2)||_2
= ||-\alpha(v - v') + \mu(v - v') + \mu A^T(A(v - v'))||_2
\leq (\alpha + \mu + \delta_\mu A)||v - v'||_2,
\]
where \( \delta_\mu A \) is the spectral norm of matrix \( A \). Letting \( L = \alpha + \mu + \delta_\mu A \), we obtain \( (49) \).

**APPENDIX D**

**Proof of Theorem 7**

Before we show its proof, we give one definition and two lemmas that are used to establish Theorem 7.

**Definition** Define the following local functions
\[
D(p, z_1, z_2, y_1, y_2) = \min_{0 \leq e_2 \leq 1} \mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2), \tag{68}
\]
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \min_{0 \leq e_2 \leq 1} \mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2), \tag{69}
\end{align*}
\]
\[
\mathcal{P}(p, z_1, z_2) = \min_{0 \leq e_2 \leq 1} \mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2), \tag{70}
\]
\[
\begin{align*}
\mathcal{V}(p, z_1, z_2) &= \min_{0 \leq e_2 \leq 1} \mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2), \tag{71}
\end{align*}
\]
where function \( \mathcal{F}(\cdot) \) is expressed by
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \frac{p}{2} ||v - p||^2 + \frac{p}{2} ||e_1 - z_1||^2 + \frac{p}{2} ||e_2 - z_2||^2, \tag{72}
\end{align*}
\]
Based on Definition 1, the following inequalities hold.

**Lemma 4:** Suppose \( \rho > \alpha > 0 \), then we have
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \frac{p}{2} ||v - p||^2 + \frac{p}{2} ||e_1 - z_1||^2 + \frac{p}{2} ||e_2 - z_2||^2, \tag{73}
\end{align*}
\]
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \frac{p}{2} ||v - p||^2 + \frac{p}{2} ||e_1 - z_1||^2 + \frac{p}{2} ||e_2 - z_2||^2, \tag{74}
\end{align*}
\]
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \frac{p}{2} ||v - p||^2 + \frac{p}{2} ||e_1 - z_1||^2 + \frac{p}{2} ||e_2 - z_2||^2, \tag{75}
\end{align*}
\]
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \frac{p}{2} ||v - p||^2 + \frac{p}{2} ||e_1 - z_1||^2 + \frac{p}{2} ||e_2 - z_2||^2, \tag{76}
\end{align*}
\]
\[
\begin{align*}
\mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2) &= \frac{p}{2} ||v - p||^2 + \frac{p}{2} ||e_1 - z_1||^2 + \frac{p}{2} ||e_2 - z_2||^2, \tag{77}
\end{align*}
\]
where
\[
\begin{align*}
\epsilon_1 &= \frac{(\rho - \alpha)^2}{(\rho - L + 2c)^2}, \epsilon_2 = \frac{(\rho - \alpha)^2}{(\rho - L + 2c)^2}, \epsilon_3 = \frac{(\rho - \alpha)^2}{\rho^2}, \epsilon_4 = \frac{(\rho - \alpha)^2}{\rho^2}, \epsilon_5 = \frac{(\rho - \alpha)^2}{\rho^2}.
\end{align*}
\]
Moreover, if
\[
\begin{align*}
\mathcal{P}(p, z_1, z_2, y_1, y_2) &= \min_{0 \leq e_2 \leq 1} \mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2), \tag{78}
\end{align*}
\]
\[
\begin{align*}
\mathcal{V}(p, z_1, z_2, y_1, y_2) &= \min_{0 \leq e_2 \leq 1} \mathcal{F}(v, e_1, e_2, p, z_1, z_2, y_1, y_2), \tag{79}
\end{align*}
\]
where \( \Delta > 0 \) is some constant, then there exists \( \epsilon_6 > 0 \) such that
\[
\begin{align*}
\mathcal{V}(p, z_1, z_2, y_1, y_2) - \mathcal{P}(p, z_1, z_2, y_1, y_2) \leq \epsilon_6,
\end{align*}
\]
where \( y^*_1(p, z_1, z_2) \) and \( y^*_2(p, z_1, z_2) \) are the solution sets of dual multipliers for problem (70).

**Proof** See Appendix E.

To save space, throughout the whole proof we denote
functions $F$, $D$ and $P$ at the $k$th iteration by
\[F^k := F(v^k, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k),\]
\[D^k := D(p^k, z_1^k, z_2^k, y_1^k, y_2^k),\]
\[P^k := P(p^k, z_1^k, z_2^k),\]
respectively. Using the above abbreviations, we further introduce the following lemma.

**Lemma 5:** Let $\alpha \leq \mu \lambda_{\min}(AT^TA)$. Then, the following inequalities hold
\[
F^k - F^{k+1} \geq \frac{\mu + \mu}{2} \left[ \frac{v^k - v^{k+1}}{2} + \frac{\rho}{2} \left[ \frac{e_1^k - e_1^{k+1}}{2} + \frac{e_2^k - e_2^{k+1}}{2} \right] \right] \|\frac{z_1^k - z_1^{k+1}}{2} + \frac{z_2^k - z_2^{k+1}}{2}\|_2^2.
\]
(80)

\[D^{k+1} - D^k \geq \mu \left[ \frac{A v^k + e_1^k - b}{v^k - e_2^k} \right] T \left[ \frac{p^{k+1} - p^k}{2} \right] \left[ \frac{z_1^k - z_1^{k+1}}{2} + \frac{z_2^k - z_2^{k+1}}{2} \right] \psi^k,
\]
(81)

\[P^k - P^{k+1} \geq \mu \left[ \frac{v^k - v^{k+1}}{2} + \frac{\rho}{2} \left[ \frac{z_1^k - z_1^{k+1}}{2} + \frac{z_2^k - z_2^{k+1}}{2} \right] \right] \|\frac{p^{k+1} - p^k}{2} \|_2^2.
\]
(82)

where $\eta$ and "$\phi^k$" and "$\psi^k$" are defined as follows
\[
\eta = 1 + \frac{1}{\sqrt{\rho}},
\]
(83a)

\[\phi^k = \begin{bmatrix} A v^k + e_1^k - b \\ \sqrt{\frac{\rho}{2}} \left[ \frac{z_1^k - z_1^{k+1}}{2} + \frac{z_2^k - z_2^{k+1}}{2} \right] \right] \left[ \frac{v^k - v^{k+1}}{2} + \frac{\rho}{2} \left[ \frac{e_1^k - e_1^{k+1}}{2} + \frac{e_2^k - e_2^{k+1}}{2} \right] \right] \psi^k,
\]
(83b)

\[\psi^k = \begin{bmatrix} \frac{p^{k+1} - p^k}{2} \\ \frac{z_1^k - z_1^{k+1}}{2} + \frac{z_2^k - z_2^{k+1}}{2} \end{bmatrix} \left[ \frac{v^k - v^{k+1}}{2} + \frac{\rho}{2} \left[ \frac{e_1^k - e_1^{k+1}}{2} + \frac{e_2^k - e_2^{k+1}}{2} \right] \right] \|\frac{p^{k+1} - p^k}{2} \|_2^2.
\]
(83c)

**Proof:** See Appendix [G]

Now we are ready to prove Theorem [I]

**Proof** First, we define a potential function as follows
\[
\Psi = F - 2D + 2P.
\]
(84)

The key to proving convergence of the proposed proximal-ADMM algorithm is to verify that the function $\Psi$ not only decreases sufficiently in each iteration but also is lower-bounded.

Based on (80)-(82) in Lemma 5 we obtain
\[
\Psi^k - \Psi^{k+1} = (F^k - F^{k+1} + 2(D^{k+1} - D^k) + 2(P^k - P^{k+1}) \geq \frac{\mu + \mu}{2} \left[ \frac{v^k - v^{k+1}}{2} + \frac{\rho}{2} \left[ \frac{e_1^k - e_1^{k+1}}{2} + \frac{e_2^k - e_2^{k+1}}{2} \right] \right] \|\frac{z_1^k - z_1^{k+1}}{2} + \frac{z_2^k - z_2^{k+1}}{2}\|_2^2.
\]
(85)

For the last term of (85), we have the following derivations
\[
\|A v^k + e_1^k - b\|_2^2 - 2\|Av^k + e_1^k - b\|^T \phi^k
\]
= \left(\begin{bmatrix} A I_{M} 0 \\ I_{N} 0 - I_{N} \end{bmatrix} \right) v^k - v^{k+1} - \frac{e_1^k - e_1^{k+1}}{2} + \frac{e_2^k - e_2^{k+1}}{2} \right) \|\phi^k\|_2.
\]
(86)

From (73) in Lemma 3, we can further get
\[
\|A v^k + e_1^k - b\|_2^2 - 2\|Av^k + e_1^k - b\|^T \phi^k
\]
\[
\|e_1^k - e_1^{k+1}\|_2^2 - \|\phi^k\|_2^2.
\]
(87)

To facilitate derivations later, we define
\[
\chi^k := \begin{bmatrix} v(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}) - v(p^k, z_1^k, z_2^k) \\ e_1(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}) - e_1(p^k, z_1^k, z_2^k) \\ e_2(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}) - e_2(p^k, z_1^k, z_2^k) \end{bmatrix}.
\]

Then, the last term in (87) can be rewritten as (88). Applying property $2ab \leq a^2 + b^2$, we derive "(88) - (a)" as
\[
2 \left( \begin{bmatrix} p^{k+1} - p^k \\ z_1^{k+1} - z_1^k \\ z_2^{k+1} - z_2^k \end{bmatrix} \right)^T \chi^k \leq \|\chi^k\|_2^2 \leq \|z_1^{k+1} - z_1^k\|^2 + \chi^k_2^2 + \frac{\rho}{2} \|z_1^{k+1} - z_1^k\|^2_2.
\]
(89)

Moreover, according to Cauchy-Schwarz inequality and (77) in Lemma 2, inequality (90) holds. Then, plugging (89) and
Furthermore, plugging (91) into (87), we have

\[
\begin{align*}
\left\| e_2 \left( p^k, z_1^k, z_2^k \right) - z_2^k \right\|_2^2 \leq \lambda^2 \| \chi^k \|_2^2.
\end{align*}
\]

Moreover, we define

\[
\zeta := \min \{ \Delta, \sigma (\Delta/\sqrt{6} \omega) \},
\]

where \( \sigma(\cdot) \) is some function satisfying \( \lim \sigma(\epsilon) = 0 \). Since \( 0 < \beta \leq 1 \) (see (37) below), we can denote \( \beta \)'s upper-bound as

\[
\beta < \min \left\{ 1, \left( \frac{\rho + \mu}{2} \right)^2, \frac{\beta^2}{\rho \omega v^2} \right\},
\]

Moreover, we define the following inequalities

\[
\left\| \phi \right\|_2 \leq 2 \sqrt{\rho \omega v^2} \beta, \quad \left\| v - v^{k+1} \right\|_2 \leq 8 \rho \omega v^2 \beta.
\]

Now, we are ready to check the boundness of \( \chi^k \). First, we assume all of the inequalities (97a)-(97c) hold. Then, plugging (94)-(96) into (97b), we can obtain (98) and (99) simultaneously.

In the following, we show that term \( \rho \omega \beta \| \chi^k \|_2^2 \) can be bounded by the previous three terms.

First, since \( 0 \leq v \leq 1 \), \( Av \) is bounded. Moreover, since \( e_1 \geq 0 \) and \( Av + e_1 - b = 0 \), there exists some positive vector \( \theta \) such that \( e_1 \leq \theta \). Then, we can define

\[
V := \max_{0 \leq v \leq e_1, e_2 \leq \theta} \left\| \begin{bmatrix} v & e_1^T \\ e_2 & e_2^T \end{bmatrix} \right\|_2.
\]

1 See proofs in Appendix E.
Combining it with (98), we can see (79) holds.

Moreover, noticing $X^k$ and $\phi^k$ are on the right side of the inequalities (79) and (75) respectively, we have the following inequality chain

$$
\|X^k\|^2 \leq \frac{1}{\varepsilon_3} \begin{bmatrix} y^{k+1} - y^*_1 \{ p^k, z^k_1, z^k_2 \} \\
y^{k+1} - y^*_2 \{ p^k, z^k_1, z^k_2 \} \end{bmatrix}^{\frac{p}{2}} \leq \frac{\varepsilon_3}{\varepsilon_3} \|\phi^k\|_2^2,
$$

where the first inequality comes from (75), and the second inequality comes from (79). Moreover, according to (96), (102) can be further derived to

$$
\rho \omega \beta \|X^k\|^2 \leq \frac{\mu}{2} \|\phi^k\|_2^2.
$$

Next, we consider the case that at least one of the inequalities (97a)-97c does not hold. There are three scenarios:

1) (97a) does not hold i.e., \[ \begin{bmatrix} v^k - v^{k+1} \\
e^k_1 - e^{k+1}_1 \\
e^k_2 - e^{k+1}_2 \end{bmatrix} \leq \frac{\rho + \mu}{8} \begin{bmatrix} v^k - v^{k+1} \\
e^k_1 - e^{k+1}_1 \\
e^k_2 - e^{k+1}_2 \end{bmatrix} \leq \frac{8 \rho \omega \beta}{\rho + \mu} \|X^k\|_2^2. \]

Then, we have the following derivations

$$
\frac{\rho + \mu}{4} \begin{bmatrix} v^k - v^{k+1} \\
e^k_1 - e^{k+1}_1 \\
e^k_2 - e^{k+1}_2 \end{bmatrix} \leq \rho \omega \beta \|X^k\|_2^2.
$$

Since $V \geq \|X^k\|_2$ (see (94)), we can further get

$$
\frac{\rho + \mu}{4} \begin{bmatrix} v^k - v^{k+1} \\
e^k_1 - e^{k+1}_1 \\
e^k_2 - e^{k+2}_2 \end{bmatrix} \leq \frac{8 \rho \omega \beta}{\rho + \mu} \|X^k\|_2^2.
$$

2) (97b) does not hold i.e., \[ \|\phi^k\|^2 \leq \frac{2 \rho \omega \beta \rho + \mu}{\mu} \|X^k\|^2. \]

By exploiting the above inequality and $V \geq \|X^k\|_2$, we can get

$$
\mu \|\phi^k\|^2 \leq \rho \omega \beta \|X^k\|^2 \leq \frac{\mu}{2} \|\phi^k\|^2.
$$

3) (97c) does not hold i.e., \[ \begin{bmatrix} 0 \\cdots \ 0 \\
\phi^k - \phi^{k+1} \end{bmatrix} \leq \frac{2 \rho \omega \beta \rho + \mu}{\mu} \|X^k\|_2^2. \]

Through similar derivations to (105) and (106), we can obtain

$$
\frac{\rho}{3 \beta} \begin{bmatrix} p^k - p^{k+1} \\
z^k_1 - z^{k+1}_1 \\
z^k_2 - z^{k+1}_2 \end{bmatrix} \leq \frac{6 \rho \omega \beta}{\rho + \mu} \begin{bmatrix} p^k - p^{k+1} \\
z^k_1 - z^{k+1}_1 \\
z^k_2 - z^{k+1}_2 \end{bmatrix} \leq \frac{6 \rho \omega \beta}{\rho + \mu} \|X^k\|_2^2.
$$

Then, from (103), (105), (106), and (107), we can derive (93) as

$$
\Psi^k - \Psi^{k+1} \geq \frac{\rho + \mu}{8} \begin{bmatrix} v^k - v^{k+1} \\
e^k_1 - e^{k+1}_1 \\
e^k_2 - e^{k+1}_2 \end{bmatrix} \leq \frac{\rho}{6 \beta} \begin{bmatrix} p^k - p^{k+1} \\
z^k_1 - z^{k+1}_1 \\
z^k_2 - z^{k+1}_2 \end{bmatrix} \leq \frac{\rho}{6 \beta} \frac{\|X^k\|_2^2}{2}.
$$

Adding both sides of the above inequality from $k = 1, 2, \ldots$, we can get

$$
\lim_{k \to +\infty} \Psi^k - \Psi^{k+1} \geq \frac{\rho + \mu}{8} \sum_{k=1}^{+\infty} \frac{\|e^k_1 - e^{k+1}_1\|_2^2}{2} + \frac{\rho}{6 \beta} \sum_{k=1}^{+\infty} \frac{\|e^k_2 - e^{k+1}_2\|_2^2}{2}.
$$

(109)

According to Definition 2 we can see that $\mathcal{F}^k \geq \mathcal{D}^k$, $\mathcal{P}^k \geq \mathcal{D}^k$ and $\mathcal{P}^k$ are lower-bounded. Therefore, $\Psi^k = (\mathcal{F}^k - \mathcal{D}^k) + (\mathcal{P}^k - \mathcal{D}^k) + \mathcal{P}^k$ means that $\Psi$ is also lower-bounded. Therefore, we can obtain

$$
\lim_{k \to +\infty} v^k - v^{k+1} = 0, \quad \lim_{k \to +\infty} e^k_1 - e^{k+1}_1 = 0,
$$

(110)

$$
\lim_{k \to +\infty} e^k_2 - e^{k+1}_2 = 0, \quad \lim_{k \to +\infty} z^k_1 - z^{k+1}_1 = 0,
$$

(112)

and

$$
\lim_{k \to +\infty} e^k_2 - z^{k+1}_2 = 0.
$$

Plugging (110) into (37d), we can obtain

$$
\lim_{k \to +\infty} v^{k+1} - p^k = 0, \quad \lim_{k \to +\infty} e^k_1 - z^{k+1}_1 = 0,
$$

(113)

and

$$
\lim_{k \to +\infty} e^k_2 - z^{k+1}_2 = 0.
$$

Plugging (110) into (73) and (77) respectively, we have

$$
\begin{bmatrix} v^k - v(p^k, z^k_1, z^k_1, y^k_1, y^k_2) \\
e^k_2 - e_2(p^k, z^k_1, z^k_1, y^k_1, y^k_2) \end{bmatrix} = 0,
$$

(114)

respectively. From (37a), we have

$$
\begin{bmatrix} y^k_1 - y^{k+1}_1 \\
y^k_2 - y^{k+1}_2 \end{bmatrix} = A_{IM} \begin{bmatrix} e^k_1 + 1 \\
e^k_2 + 1 \end{bmatrix} - b.
$$

(115)
Plugging (83b) into (115), we have
\[
\begin{align*}
\left[ y_{k+1}^{k+1} - y_1^k \right] \\
\left[ y_{k+1}^1 - y_2^k \right]
\end{align*}
\]
\[
= \begin{bmatrix} A I_M & 0 \\ N & 0 \end{bmatrix} \begin{bmatrix} v_{k+1}^1 - v(p_{k+1}, z_{k+1}, z_{k+1}^1, y_1^k, y_2^k) \\ e_{k+1}^1, e_1^1(p_{k+1}, z_{k+1}, z_{k+1}^1, y_1^k, y_2^k) \end{bmatrix}
\]
\[
+ \begin{bmatrix} A I_M & 0 \\ N & 0 \end{bmatrix} \begin{bmatrix} v(p_k^k, z_{k+1}^k, z_{k+1}^1, y_1^k, y_2^k) \\ e_1(p_k^k, z_{k+1}^k, z_{k+1}^1, y_1^k, y_2^k) \end{bmatrix}
\]  

(116)

Plugging (111), (113), and (114) into (116), we can obtain
\[
\lim_{k \to \infty} y_{k+1}^1 - y_1^k = 0, \quad \lim_{k \to \infty} y_{k+1}^2 - y_2^k = 0. \quad (117)
\]

From (35b) and (35c), we can see clearly that \{e_k^1\} and \{v^k\} are bounded sequences since \{e_k^1\} is bounded by \([0, 1]^N\). Plugging these bounded results into (112), we can find that \{p^k\}, \{z_1^k\}, \{z^k_2\} are also bounded sequences. Furthermore, based on the above bounded results, (26) and (38) imply that \{y_k^1\} and \{y^k_2\} are also bounded sequences.

Combining these bounded results with (110) and (117), we can conclude
\[
\begin{align*}
\lim_{k \to \infty} v^k &= v^*, \quad \lim_{k \to \infty} e_1^k = e_1^*, \quad \lim_{k \to \infty} e_2^k = e_2^*, \\
\lim_{k \to \infty} p^k &= p^*, \quad \lim_{k \to \infty} z_1^k = z_1^*, \quad \lim_{k \to \infty} z_2^k = z_2^*, \\
\lim_{k \to \infty} y^k_1 &= y_1^*, \quad \lim_{k \to \infty} y^k_2 = y_2^*.
\end{align*}
\]

(118)

Plugging (118) into (35b) and (112), we can get
\[
Av^* + e_1^* - b = 0, \quad v^* = e_2^*, \quad v^* = p^*, \quad e_1^* = z_1^*, \quad e_2^* = z_2^*.
\]

(119)

which completes the proof of the first part of Theorem 1.

Next, we prove that \(v^*\) is a stationary point of the original problem (33). Letting \(g(v) = \lambda^T v - \frac{\rho}{\mu}(v - 0.5) - \frac{\rho}{\mu} [v - 0.5]^2\), we can obtain, \(\forall x \in X\),
\[
(v - v^*)^T \nabla_v g(v^*) = (v - v^*)^T (\lambda - \alpha(v - 0.5)). \quad (120)
\]

Moreover, since \(v_{k+1} = \text{argmin}_v F(v, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k)\), then we have
\[
0 = \nabla_v F(v^*, e_1^*, e_2^*, p^*, z_1^*, z_2^*, y_1^*, y_2^*) = \lambda - \alpha(v^* - 0.5) + \rho(v^* - p^*) + A^T y_1^* + y_2^*
\]
\[
+ \mu A^T (Av^* + e_1^* - b) + \mu (v^* - e_2^*)
\]
\[
= \nabla_v g(v^*) + A^T y_1^* + y_2^*,
\]

i.e., \(\nabla_v g(v^*) = -A^T y_1^* - y_2^*\), where the last equality follows from (119). Then, we can further obtain
\[
(v - v^*)^T \nabla_v g(v^*) = (v - v^*)^T (A^T y_1^* + y_2^*)
\]
\[
= (y_1^*)^T A(v - v^*) - (y_2^*)^T (v - v^*).
\]

(122)

Obviously, if \((y_1^*)^T A(v - v^*) \leq 0\) and \((y_2^*)^T (v - v^*) \leq 0\), then \((v - v^*)^T \nabla_v g(v^*) \geq 0\). In the following, we will prove that both of them hold.

First, we have
\[
(y_1^*)^T A(v - v^*) = (y_1^*)^T ((b - e_1^*) - (b - e_1))
\]
\[
= (y_1^*)^T (e_1^* - e_1). \quad (123)
\]

Then, since \(e_1^* \geq 0\), we have the following derivations
\[
(y_1^*)^T e_1^* = \sum_{i_1^*, j_1^*>0} y_{1, j}^* e_{1, j}^*
\]
\[
= \sum_{i_1^*, j_1^*>0} \left[ \frac{\mu}{\rho + \mu} y_{1, j}^* (y_j - a_j^* v^* - \frac{y_{1, j}^*}{\rho + \mu}) + \rho y_{1, j}^* \right]
\]
\[
= \sum_{i_1^*, j_1^*>0} y_{1, j}^* (y_j - a_j^* v^* - \frac{y_{1, j}^*}{\rho + \mu}),
\]

where the last equality holds since \(z_1^* = b - Av^*\), which follows from \(e_1^* = z_1^*\) and \(Av^* + e_1^* - b = 0\). Moreover, since \(Av^* + e_1^* - b = 0\), we also have
\[
(y_1^*)^T e_1^* = (y_1^*)^T (b - Av^*)
\]
\[
= \sum y_{1, j}^* (y_j - a_j^* x^*) + \sum y_{1, j}^* (b - a_j^* x^*)
\]
\[
= \sum y_{1, j}^* (b - a_j^* v^*).
\]

Comparing (124) and (125), we can see that when \(e_{1, j}^* > 0\),
\[
y_{1, j}^* = 0.
\]

(126)

Therefore, we can obtain
\[
(y_1^*)^T e_1^* = 0, \quad (127)
\]

since \(e_1^* \geq 0\).

On the other hand, if \(e_{1, j}^* = 0\), there exists \(\frac{\mu}{\rho + \mu} (y_j - a_j^* v^* - \frac{y_{1, j}^*}{\rho + \mu}) \leq 0\) (see (140)). Moreover, since \(e_1^* = z_1^* = b - Av^*\), one can see that \(y_{1, j}^* \geq \frac{1}{\rho + \mu} \sum_{i_1^*, j_1^*>0} e_{1, j}^* \geq 0\). Besides, since \(\rho > 0\) and \(\mu > 0\), thus we have
\[
y_{1, j}^* \geq 0,
\]
when \(e_{1, j}^* = 0\). From (127) and (128), we conclude
\[
y_1^* \geq 0.
\]

(129)

Plugging (127) and (129) into (123), we obtain \((y_1^*)^T (e_1^* - e_1) = -(y_1^*)^T e_1 \leq 0\), which means
\[
(y_1^*)^T A(v - v^*) \leq 0. \quad (130)
\]

Similar to the above derivations for (130), we can also have
\[
(y_2^*)^T (v - v^*) \leq 0. \quad (131)
\]

This completes the proof.
**APPENDIX E**

**Proof of (100)**

We prove by contradiction. Suppose (100) does not hold. Since \( \lim_{\epsilon \to 0} \sigma(\epsilon) = 0 \), and suppose there exists a sequence of tuples

\[
\{ \tilde{p}^k, v^k_1, v^k_2, y_2^{k+1}, y_1^{k+1} \} = \{ \tilde{p}, \tilde{v}_1, \tilde{v}_2, \tilde{y}_1, \tilde{y}_2 \},
\]

such that

\[
\lim_{k \to +\infty} ||\phi^k||_2 = 0.
\]

\[
\lim_{k \to +\infty} ||\lambda^k||_2 > 0.
\]

Next we prove (134) and (135) cannot hold simultaneously.

First, plugging the convergence result (133) into (134) and noticing \( e_1(\tilde{p}^k, \tilde{z}_1^k, \tilde{z}_2^k, y_2^{k+1}, y_1^{k+1}) \) is continuous (see (69) and \( \phi^k \) in (335), we can see

\[
\left[ \begin{array}{c}
Av(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2) + e_1(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2) - b \\
v(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2) - e_2(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2)
\end{array} \right] = 0.
\]

Plugging (136) into KKT equations of problem (69) and notice \( \tilde{y}_1 \) and \( \tilde{y}_2 \) are the corresponding optimal Lagrangian multipliers (problem (69) is strongly convex), we can see

\[
\begin{align*}
&v(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2) \\
e_1(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2) = e_1(\tilde{p}, \tilde{z}_1, \tilde{z}_2) \\
e_2(\tilde{p}, \tilde{z}_1, \tilde{z}_2, \tilde{y}_1, \tilde{y}_2) = e_2(\tilde{p}, \tilde{z}_1, \tilde{z}_2)
\end{align*}
\]

which indicates \( \lim_{k \to +\infty} ||\lambda^k||_2 = 0 \). This is a contradiction.

**APPENDIX F**

**Proof of Lemma 7**

First, we prove (73). To simplify the proof, \( F(v, e_1, e_2, \bullet^k) \) is used to denote function \( F(v, e_1, e_2, p^k, z_1^k, z_2^k, y_1^k, y_2^k) \). Moreover, we give the following definitions relative to \( v, e_1 \) and \( e_2 \) respectively

\[
r_{v}^k = v(\tilde{p}, \tilde{v}_1, \tilde{v}_2, \bullet^k) - v, \]

\[
r_{e_1}^k = e_1(\tilde{p}, \tilde{v}_1, \tilde{v}_2, \bullet^k) - e_1, \]

\[
r_{e_2}^k = e_2(\tilde{p}, \tilde{v}_1, \tilde{v}_2, \bullet^k) - e_2.
\]

Applying the triangle inequality to \( r_{v}^k \), we obtain

\[
||r_{v}^k||_2 \leq ||v(\tilde{p}, \tilde{v}_1, \tilde{v}_2, \bullet^k) - v|| + ||v - v||_2,
\]

\[
\leq (\rho + L + 1)||v + \rho - v||_2,
\]

where \( L \) is the Lipschitz constant defined in (49). Following similar derivations to (138), we can get

\[
\begin{align*}
||r_{e_1}^k||_2 &\leq ||e_1(\tilde{p}, \tilde{v}_1, \tilde{v}_2, \bullet^k) - e_1|| + ||e_1 - e_1||_2 \\
&= (\rho + \mu + 1)||e_1 + \mu - e_1||_2.
\end{align*}
\]

Moreover, according to the fixed point theorem, we have

\[
\begin{align*}
e_{1}^{k+1} &= e_{1}^{k} - \frac{1}{\mu + 1} e_{1}^{k} = e_{1}^{k} - \frac{1}{\mu + 1} e_{1}^{k}, \\
e_{2}^{k+1} &= e_{2}^{k} - \frac{1}{\mu + 1} e_{2}^{k}. \end{align*}
\]

Thus, we have the following derivations

\[
\begin{align*}
\left. \begin{array}{c}
v\left. \right)v - v, \right) \\
||e_1 - e_1||_2
\end{array} \right) &= \left. \begin{array}{c}
\frac{1}{\mu + 1} e_1 \\
\frac{1}{\mu + 1} e_2
\end{array} \right) + \left. \begin{array}{c}
\frac{1}{\mu + 1} e_1 \\
\frac{1}{\mu + 1} e_2
\end{array} \right) + \left. \begin{array}{c}
\frac{1}{\mu + 1} e_1 \\
\frac{1}{\mu + 1} e_2
\end{array} \right)
\end{align*}
\]

where the inequality comes from the triangle inequality. Following the non-expansiveness property of projection operations, (144) can be deduced to

\[
\begin{align*}
||e_1 - e_1||_2 &\leq ||e_1 - e_1||_2 + ||e_1 - e_1||_2 \\
&= \left( \rho + \mu + 1 \right)||e_1 + \mu - e_1||_2 + ||e_1 - e_1||_2.
\end{align*}
\]

Moreover, (145) can be further derived as

\[
\begin{align*}
\left. \begin{array}{c}
v\left. \right)v - v, \right) \\
||e_1 - e_1||_2
\end{array} \right) &= \left. \begin{array}{c}
\frac{1}{\mu + 1} e_1 \\
\frac{1}{\mu + 1} e_2
\end{array} \right) + \left. \begin{array}{c}
\frac{1}{\mu + 1} e_1 \\
\frac{1}{\mu + 1} e_2
\end{array} \right) + \left. \begin{array}{c}
\frac{1}{\mu + 1} e_1 \\
\frac{1}{\mu + 1} e_2
\end{array} \right)
\end{align*}
\]
where “$\geq$” holds since $\mathcal{F}(v, e_1, e_2, e_k)$ is a strongly convex function with modulus $(\rho - \alpha) > 0$ [42] and “$\geq$” follows from (41). Then, we can obtain (73) as follows

$$\|v^k - v^{k+1}\|_2^2 \geq \epsilon_1 \left[ v^k - v \left( p^k, z_1^k, z_2^k, y_1^k, y_2^k \right) \right]_2, $$

$$\|e_1^k - e_1 \|_2^2 + \|e_2^k - e_2 \|_2^2 \geq \epsilon_2 \left[ v^k - v \left( p^k, z_1^k, z_2^k, y_1^k, y_2^k \right) \right]_2,$n

where $\epsilon_1 = \frac{(\rho - \alpha)^2}{16L}$. Next, we prove that (72) holds. Based on the triangle inequality and (72), we have

$$\|v^k - v^{k+1}\|_2^2 \geq \epsilon_2 \left[ v^k - v \left( p^k, z_1^k, z_2^k, y_1^k, y_2^k \right) \right]_2, $$

$$\|e_1^k - e_1 \|_2^2 + \|e_2^k - e_2 \|_2^2 \geq \epsilon_2 \left[ v^k - v \left( p^k, z_1^k, z_2^k, y_1^k, y_2^k \right) \right]_2,$n

where $\epsilon_2 = \frac{(\rho - \alpha)^2}{2(\rho + L + 2 + \epsilon_1) \epsilon_1}$. Moreover, through similar proofs for (3.6)-(3.8) in [41], we can verify that inequalities (75)-(77) hold. In addition, since $g(v, e_1, e_2) = g(v) = \lambda^2 v - \frac{\alpha}{2} ||v - 0.5||_2^2$ is Lipschitz differentiable corresponding to variables $v, e_1, e_2$ with constant $L_\alpha > \alpha$, based on Proposition 2.3 in [41], we can see that problem (35) satisfies the strict complementary condition. Then, we can prove that inequality (79) holds through similar derivations to (3.9) in [41]. This ends the proof.

**APPENDIX G**

**PROOF OF Lemma 5**

**Proof** First, we define the following quantities

$$\mathcal{F}_v^k = \mathcal{F}(v, e_1, e_2, p^k, z_1^k, z_2^k, y_1^k, y_2^k),$$

$$\mathcal{F}_e_1^k = \mathcal{F}(v^{k+1}, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k),$$

$$\mathcal{F}_e_2^k = \mathcal{F}(v^{k+1}, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k),$$

$$\mathcal{F}_p^k = \mathcal{F}(v^{k+1}, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k),$$

It is easy to see

$$\mathcal{F}_v^k - \mathcal{F}_v^{k+1} = \mathcal{F}_v^k + \mathcal{F}_e_1^k + \mathcal{F}_e_2^k + \mathcal{F}_p^k + \mathcal{F}_y^k.$$

From (37a) and (72), we can find

$$v^{k+1} = \text{argmin}_{v} \mathcal{F}(v, e_1, e_2, p^k, z_1^k, z_2^k, y_1^k, y_2^k).$$

Since $\mathcal{F}(v, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k)$ is a strongly convex function, we have

$$\mathcal{F}_v^k = \frac{\rho + \mu - \alpha}{2} ||v^k - v^{k+1}||_2^2 + \frac{\mu}{2} \mathcal{A}(v^k - v^{k+1})_2 \geq \frac{\rho + \mu - \alpha + \mu \lambda_\text{min}(A^TA)}{2} ||v^k - v^{k+1}||_2^2.$$n

Under the assumption of $\mu \lambda_\text{min}(A^TA) \geq \alpha$, we can obtain

$$\mathcal{F}_v^k \geq \frac{\rho + \mu}{2} ||v^k - v^{k+1}||_2^2.$$n

Similarly, we can also obtain

$$\mathcal{F}_e_1^k \geq \frac{\rho + \mu}{2} ||e_1^k - e_1^{k+1}||_2^2.$$n

Plugging (37d) into (148d), we can derive it as (153), where the last inequality holds since $0 < \beta \leq 1$. Plugging (37c) into (148d), we can obtain

$$\mathcal{F}_y^k = \mathcal{F}(v^{k+1}, e_1^k, e_2^k, p^k, z_1^k, z_2^k, y_1^k, y_2^k),$$

and

$$\mathcal{F}_e_1^k \geq \frac{\rho + \mu}{2} ||e_1^k - e_1^{k+1}||_2^2.$$n

Then, plugging (150) into (149), we have

$$\mathcal{F}_y^k \geq \mathcal{F}_v^k + \mathcal{F}_e_1^k + \mathcal{F}_e_2^k + \mathcal{F}_p^k + \mathcal{F}_y^k.$$

That completes the proof of (80).

Next, we consider to prove inequality (81). To facilitate discussions later, we define

$$D_{p^k} = D_{p^k} - D(p^k, z_1^k, z_2^k, y_1^k, y_2^k),$$

$$D_{y^k} = D(p^k, z_1^k, z_2^k, y_1^k, y_2^k).$$

Then, we have

$$D_{p^k} - D_{y^k} = D_{p^k} + D_{y^k}.$$n

According to (68), we can write $D_{p^k}$ as

$$D_{p^k} = F(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$e_2(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}.$$n

$$\mathcal{F}(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$e_2(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}.$$n

$$\mathcal{F}(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$e_2(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}.$$n

$$\mathcal{F}(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$e_2(p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}),$$

$$p^{k+1}, z_1^{k+1}, z_2^{k+1}, y_1^{k+1}, y_2^{k+1}.$$n

\[ F_{k+1} = \frac{\alpha}{2} \left( (p^{k+1} - p^k)^T (2v^{k+1} - p^{k+1} - p^k) + (z_1^{k+1} - z_1^k)^T (2e_1^{k+1} - z_1^k) + (z_2^{k+1} - z_2^k)^T (2e_2^{k+1} - z_2^k) \right) \]

According to (69), (158) can be rewritten as

\[ D_{pz} k \geq F(p^{k+1}; z_1^k, z_2^k, y_1^k; y_2^k), e_1(p^{k+1}; z_1^k, z_2^k, y_1^k; y_2^k), e_2(p^{k+1}; z_1^k, z_2^k, y_1^k; y_2^k), p^{k+1}; z_1^k, z_2^k, y_1^k; y_2^k) \]

Plugging (72) into (159), we can obtain

\[ D_{pz} \geq \frac{\alpha}{2} (p^{k+1} - p^k)^T (p^{k+1} - p^k) - 2v(p^{k+1}; z_1^k, z_2^k, y_1^k; y_2^k) \]

which can be further written as

\[ D_{pz} \geq \frac{\alpha}{2} \mu \left[ A v^k + e_1^k - b^k \right]^T \phi^k . \]

Through similar derivations, we can also get

\[ D_{z} \geq \mu \left[ A v^k + e_1^k - b^k \right]^T \phi^k . \]

Thus, combining (161) and (162), we obtain (81).

Finally, we consider to prove (82). According to Danskin's theorem [43], we have

\[ \nabla P(p^k, z_1^k, z_2^k) = \frac{\partial}{\partial p^k} p^k - v(p^k, z_1^k, z_2^k) \]

Applying the triangle inequality property and (76) to (163), we have the following derivations

\[ \| \nabla P(p^k, z_1^k, z_2^k) - \nabla P(p^{k+1}, z_1^{k+1}, z_2^{k+1}) \|_2 \]

which means the gradient of function $P$ is Lipschitz continuous with respect to variables $p$, $z_1$, and $z_2$. The corresponding Lipschitz constant is $\rho \left( 1 + \frac{1}{\sqrt{\epsilon_4}} \right)$. Then, according to property of the Lipschitz continuous function [44], we have

\[ p^{k+1} - p^k \leq \left( \left( \left( p^{k+1} - z_1^{k+1} \right) - \left( p^k - z_1^k \right) \right)^T \left( p^{k+1} - z_1^{k+1} \right) - e_1(p^k, z_1^k, z_2^k) \right) \]

Letting $\eta = \left( 1 + \frac{1}{\sqrt{\epsilon_4}} \right)$, we can reach (82). This ends the proof.

**APPENDIX H**

**Proof of Fact H**

**Proof** To be clear, we rewrite $A$ in (144) as follows

\[ A = \left[ \hat{W}_1(Q_1 \otimes I) ; \cdots ; \hat{W}_n(Q_n \otimes I) ; \cdots ; \hat{W}_r(Q_r \otimes I) ; S \right] . \]

Moreover, $\hat{W}_r$ in (57) can be written equivalently as

\[ \hat{W}_r = \left[ P \left( \sum_{i \in K_r} T_i \oplus 2 \right) D_r \oplus 2 ; \cdots ; P \left( \sum_{i \in K_r} T_i \oplus 2 \right) D_r \right] . \]

Based on the definitions of $T_i$ and $D_r$ (see (19A) and (19B)), we can derive each term in matrix $\hat{W}_r$ as

\[ P \left( \sum_{i \in K_r} T_i \right) D_r = P \text{diag} \left( \sum_{i \in K_r} b_i \oplus 2 \right) D(2^q \cdot h_{r_1}) . \]

\[ \left( \sum_{i \in K_r} b_i \oplus 2 \right) D(2^q \cdot h_{r_2}) . \]
Since there is only one “1” in either column/row of matrix $D(2^n, h_{ij})$ and nonzero elements in $\sum_{i \in K_i} b_i^T D_{ij} + 2$ are “1”, row vector $(\sum_{i \in K_i} b_i^T D_{ij} + 2) D(2^n, h_{ij})$ only includes one nonzero element “1”, where $j = 1, 2, 3$. Moreover, since elements in matrix $P$ are 1 or -1, elements in $P(\sum_{i \in K_i} T_i) D_{ij}$ are 0, 1, or -1. Therefore, we can conclude that element $W_{\tau}$ is also either 0, 1, or -1.

Moreover, since variable-selecting matrix $Q_{\tau}$ has only one nonzero element “1” in its each row/column, $Q_{\tau} \otimes I$ should have the same property. Therefore, it is obvious that elements in $W_{\tau}(Q_{\tau} \otimes I)$ are 1, -1, or 0. Besides, since matrix $S$ only includes one nonzero element “1”, we can conclude that matrix $A$ consists of elements 1, -1, and 0. This completes the proof.

REFERENCES

[1] M. C. Davey and D. J. C. MacKay, “Low density parity check codes over GF(q),” IEEE Commun. Lett., vol. 2, no. 6, pp. 165-167, June 1998.

[2] I. B. Djordjevic and B. Vasic, “Nonbinary LDPC codes for optical communication systems,” Photonics Tech. Lett., vol. 17, no. 10, pp. 2224-2226, Oct. 2005.

[3] J. Huang, S. Zhou and P. Willett, “Nonbinary LDPC coding for multichannel underwater acoustic communication,” IEEE J. Select. Areas Commun., vol. 26, no. 9, pp. 1684-1696, Dec. 2008.

[4] Y. Maeda and H. Kano, “Error control coding for multilevel cell flash memories using nonbinary low-density parity-check codes,” 2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Chicago, IL, pp. 367-375, Oct. 2009.

[5] R. Peng and R. Chen, “Design of nonbinary quasi-cyclic LDPC cycle codes,” 2007 IEEE Information Theory Workshop, Tahoe City, CA, pp. 13-18, Sept. 2007.

[6] H. X. Song and J. R. Cruz, “Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording,” IEEE Trans. Magn., vol. 39, no. 2, pp. 1081-1087, Mar. 2003.

[7] B. Rong, T. Jiang, X. Li and M. R. Soleymani, “On the decomposition method for redundant rows for improving fractional distance,” IEEE Trans. Inf. Theory, vol. 59, no. 1, pp. 282-300, Jan. 2013.

[8] J. Feldman, T. Wadayama, and I. Takumi, “A cutting-planemethod based on redundant rows for improving fractional distance,” IEEE J. Sel. Areas Commun., vol. 27, no. 6, pp. 1012-1105, Aug. 2009.

[9] J. Huang, S. Zhou and P. Willett, “Nonbinary LDPC coding for multichannel underwater acoustic communication,” IEEE J. Select. Areas Commun., vol. 26, no. 9, pp. 1684-1696, Dec. 2008.

[10] M. C. Davey and D. J. C. MacKay, “Low density parity check codes over GF(q),” IEEE Commun. Lett., vol. 2, no. 6, pp. 165-167, June 1998.

[11] I. B. Djordjevic and B. Vasic, “Nonbinary LDPC codes for optical communication systems,” Photonics Tech. Lett., vol. 17, no. 10, pp. 2224-2226, Oct. 2005.

[12] J. Huang, S. Zhou and P. Willett, “Nonbinary LDPC coding for multicharier underwater acoustic communication,” IEEE J. Select. Areas Commun., vol. 26, no. 9, pp. 1684-1696, Dec. 2008.

[13] Y. Maeda and H. Kano, “Error control coding for multilevel cell flash memories using nonbinary low-density parity-check codes,” 2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Chicago, IL, pp. 367-375, Oct. 2009.

[14] R. Peng and R. Chen, “Design of nonbinary quasi-cyclic LDPC cycle codes,” 2007 IEEE Information Theory Workshop, Tahoe City, CA, pp. 13-18, Sept. 2007.

[15] H. X. Song and J. R. Cruz, “Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording,” IEEE Trans. Magn., vol. 39, no. 2, pp. 1081-1087, Mar. 2003.

[16] B. Rong, T. Jiang, X. Li and M. R. Soleymani, “On the decomposition method for redundant rows for improving fractional distance,” IEEE Trans. Inf. Theory, vol. 59, no. 1, pp. 282-300, Jan. 2013.

[17] J. Feldman, T. Wadayama, and I. Takumi, “A cutting-planemethod based on redundant rows for improving fractional distance,” IEEE J. Sel. Areas Commun., vol. 27, no. 6, pp. 1012-1105, Aug. 2009.

[18] J. Feldman, T. Wadayama, and I. Takumi, “A cutting-planemethod based on redundant rows for improving fractional distance,” IEEE J. Sel. Areas Commun., vol. 27, no. 6, pp. 1012-1105, Aug. 2009.