LP Pseudocodewords of Cycle Codes are Half-Integral

Nathan Axvig

Abstract

In his Ph.D. dissertation, Feldman and his collaborators define the linear programming decoder for binary linear codes, which is a linear programming relaxation of the maximum-likelihood decoding problem. This decoder does not, in general, attain maximum-likelihood performance; however, the source of this discrepancy is known to be the presence of non-integral extreme points (vertices) within the fundamental polytope, vectors which are also called nontrivial linear programming pseudocodewords. Restricting to the class of cycle codes, we provide necessary conditions for a vector to be a linear programming pseudocodeword. In particular, the components of any such pseudocodeword can only assume values of zero, one-half, or one.

I. INTRODUCTION

Let $C$ be a binary linear code of length $n$, and consider transmission over a binary-input, memoryless channel. Under the additional assumptions that the channel is symmetric and that each codeword is equally likely to be transmitted, a maximum-likelihood decoder returns a codeword that minimizes the probability of word-error. It is well-known (see, e.g., [1], [7], [8]) that the set of maximum-likelihood (ML) codewords coincides with the solution set of the following integer program:

\[
\begin{align*}
\text{minimize} & \quad \lambda^T c \\
\text{subject to} & \quad c \in C
\end{align*}
\]  

where $\lambda$ is the vector of log-likelihood ratios based on the channel output. While solutions to integer linear programs such as Problem (1) are often of great interest, solving integer linear programs in general is quite difficult. On the other hand, solutions to linear programs (problems in which the output vector is not constrained to be an integer vector) can often be found quickly (using, e.g., the simplex algorithm), and they can even be found in polynomial time (using, e.g., the ellipsoid method) [5]. A common approach to solving integer linear programs therefore is to solve a related linear program (or a series of linear programs) and then interpret the solution in light of the original problem.

Letting $\text{conv}(C)$ be the convex hull of $C$ in $\mathbb{R}^n$ (where we regard codewords as vectors in $\{0, 1\}^n \subseteq \mathbb{R}^n$), Problem (1) is equivalent to

\[
\begin{align*}
\text{minimize} & \quad \lambda^T f \\
\text{subject to} & \quad f \in \text{conv}(C).
\end{align*}
\]  

Problem (2) is a linear program; however, the constraints needed to define the feasible region $\text{conv}(C)$ explicitly are likely to be either (a) difficult to compute or (b) very large in number. Elaborating on this, suppose that there was a polynomial time algorithm that was capable of taking a general binary linear code $C$ and producing a set of constraints describing $\text{conv}(C)$. This algorithm, along with the ellipsoid method, can then be used to solve instances of the COSET WEIGHTS decision problem (see [4] for a definition) in polynomial time. Since COSET WEIGHTS is NP-complete [4], this would imply $P = NP$.

This work was supported in part by NSF grants DMS-0602332 and DMS-0838463.

N. Axvig is with the Department of Applied Mathematics, Virginia Military Institute, Lexington, VA 24450, USA.

N. Axvig’s email: axvignd10@vmi.edu
In an attempt to find solutions to Problem (2) efficiently, Feldman [7] defines the linear programming (LP) decoder. The linear programming decoder returns as its output the solution to a linear programming relaxation of Problem (2), namely

$$\begin{align*}
\text{minimize} & \quad \lambda^T f \\
\text{subject to} & \quad f \in \mathcal{P}.
\end{align*}$$

(3)

The feasible set $\mathcal{P}$ of this linear program is known as the fundamental polytope. The fundamental polytope has several properties that make it a reasonable choice for a relaxation of Problem (2), not the least of which is that, from a computational complexity standpoint, it is easier to describe than $\text{conv}(\mathcal{C})$. More specifically, if we assume that the row-weight of the parity-check matrix defining the code is bounded by a constant, then the number of constraints that describe $\mathcal{P}$ is bounded by a polynomial in $n$, where $n$ is the length of the code (see Section II-C).

The fundamental polytope is a subset of the unit hypercube $[0,1]^n$ and thus is bounded. Because $\mathcal{P}$ is bounded, the general theory of linear programming states that (a) a solution to Problem (3) always exists and that (b) we may assume that at least one solution is an extreme point of the underlying polytope [5]. We therefore make the convention that the LP decoder always returns an extreme point of the fundamental polytope. As such, we say that any extreme point of $\mathcal{P}$ is a linear programming (LP) pseudocodeword. Linear programming pseudocodewords are the principal objects of investigation in this paper.

In [7], it shown that the set of integer extreme points of $\mathcal{P}$ is precisely the set of codewords $\mathcal{C}$. Such integer vectors are therefore called trivial linear programming (LP) pseudocodewords. Feldman [7] uses this relationship between $\mathcal{C}$ and $\mathcal{P}$ to show that the LP decoder has the ML-certificate property: if an optimal solution $f^*$ to Problem (3) is an integer vector, then $f^*$ must be an ML codeword. It is often the case, however, that the fundamental polytope contains non-integer extreme points, known as nontrivial linear programming (LP) pseudocodewords, in addition to the integer-valued codewords. By the ML-certificate property, it is precisely the presence of nontrivial LP pseudocodewords that prevents the LP decoder from attaining the performance (with respect to word-error rate) of the ML decoder.

In this paper, we provide new necessary conditions for a vector to be a linear programming pseudocodeword of a cycle code. In particular, we show that any LP pseudocodeword of a cycle code must be half-integral; that is, any LP pseudocodeword must be an element of $\{0, \frac{1}{2}, 1\}^n$. Moreover, in proving this half-integrality we also discover that LP pseudocodewords of cycle codes display additional structure that can be stated succinctly in terms of the code’s Tanner graph. The results of this paper are applied in [2], where the author and his collaborator provide complete graphical characterizations of both LP pseudocodewords and minimal linear programming pseudocodewords (see [9]) for the class of cycle codes.

The remainder of this paper is organized as follows. Section II the background section, is broken into three parts. In Section II-A, we introduce relevant coding theory terminology and the family of cycle codes. Section II-B reviews some basic definitions and facts about extreme points, and Section II-C gives an explicit description of the fundamental polytope. Section III makes up the mathematical bulk of this paper, and it is divided into two parts. Section III-A is dedicated to proving a technical result that is then used in Section III-B to prove Theorem III.11, which is the main result of this work. We make concluding remarks in Section IV.

II. BACKGROUND

A. Background on Coding Theory

The results herein require knowledge not only of a code itself but also of the specific parity-check presenting the code. By “code,” we therefore mean a binary linear code equipped with a fixed parity-
Definition II.1. A Tanner graph is a bipartite graph $G$ with vertex set $X \cup U$ and edge set $E$. The elements of $X$ are called variable nodes, and the elements of $U$ are called check nodes.

Given an $r \times n$ parity-check matrix $H = (h_{j,i})$, one can construct the Tanner graph $G = G(H)$ of $H$ as follows: set $X = \{x_1, x_2, \ldots, x_n\}$ and $U = \{u_1, u_2, \ldots, u_r\}$. Define $G = G(H)$ to be the graph whose vertex set is $X \cup U$ with $x_i$ adjacent to $u_j$ if and only if $h_{j,i} = 1$. In other words, $G(H)$ is the bipartite graph whose bipartite adjacency matrix is $H$. Conversely, one can derive a parity-check matrix from a Tanner graph: index rows by vertices in $U$ and columns by vertices in $X$. Set $h_{u,x} = 1$ precisely when $x$ and $u$ are adjacent in $G$. In this manner one can see that there is a bijective correspondence between Tanner graphs and parity-check matrices.

The Tanner graph is the structure on which the class of message-passing decoding algorithms, e.g., the min-sum algorithm or the sum-product algorithm, operate. Loosely speaking, message-passing decoding algorithms work by relaying bit and/or reliability information across edges of the Tanner graph in an iterative fashion. For the purposes of this paper, however, the Tanner graph is introduced to provide a graphical environment in which to view codewords.

Definition II.2. Let $G = (X \cup U, E)$ be a Tanner graph, and let $f$ be an assignment of binary values to the variable nodes of $G$. The assignment $f$ is called a valid configuration on $G$ provided that for each $u \in U$ the quantity $\sum_{x \in N(u)} x_i$ is even, where, as is standard, $N(u)$ denotes the neighborhood of $u$ in $G$, i.e., the set of all $x \in X$ that are adjacent to $u$.

By identifying valid configurations on a Tanner graph with vectors in $\{0, 1\}^n$, we obtain a natural correspondence between valid configurations and codewords:

Proposition II.3 (see, e.g., [3]). Let $C$ be a code with parity-check matrix $H$, and let $G = G(H) = (X \cup U, E)$ be the corresponding Tanner graph. The set of valid configurations on $G$ corresponds precisely to the set $C$ of codewords.

By Proposition II.3 codewords of $C$ may be viewed as valid binary configurations on a Tanner graph. This graphical realization of codewords is further displayed in the family of cycle codes.

Definition II.4. A cycle code is a code $C$ equipped with a parity-check matrix $H$ that has uniform column weight 2.

We focus on cycle codes because they provide for a more fruitful analysis than general codes. This is due to their special structure: let $C$ be a cycle code with Tanner graph $G$. If $c \in C$, Proposition II.3 states $c$ must be a valid configuration on $G$. Since every variable node of $G$ has degree two, the subgraph of $G$ induced by $\text{supp}(c) \cup N(\text{supp}(c))$ is even, i.e., all nodes in the subgraph have even degree (herein, $\text{supp}(v)$ is used to denote the support of $v$: the set of positions where $v$ is not zero). Thus, the subgraph induced by $\text{supp}(c) \cup N(\text{supp}(c))$ must be a union of edge-disjoint simple cycles. The converse is also true: the indicator vector for a set of variable nodes within a union of edge-disjoint simple cycles is a codeword.

Example II.5. The parity-check matrix

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

This is a departure in terminology from many standard texts and publications in coding theory where a code is described intrinsically as a subspace, not extrinsically as the kernel of a specific matrix.
has uniform column weight two, and hence defines a cycle code. One can check that the vector
\[ \mathbf{c} = (1, 1, 0, 0, 1, 1, 0, 1, 0)^T \]
is in the null-space of \( H \). It also gives rise to a simple cycle in the Tanner graph of the code; see Figure [7] for an illustration. We note that an isomorphic Tanner graph appears in Figure 6.1 of [10].

B. Background on Linear Programming and Extreme Points

Most of the material in this section can be found in [5]. For our purposes, we define a linear program to be an optimization problem of the form

\[ \text{minimize } \mathbf{q}^T \mathbf{f} \]
\[ \text{subject to } \mathbf{a}_j^T \mathbf{f} \geq b_j, \quad j \in M_1 \]
\[ \mathbf{a}_j^T \mathbf{f} \leq b_j, \quad j \in M_2 \]
\[ \mathbf{a}_j^T \mathbf{f} = b_j, \quad j \in M_3 \]
\[ f_i \geq 0, \quad i \in N_1 \]
\[ f_i \leq 0, \quad i \in N_2, \]

where \( \mathbf{q}, \mathbf{f} \), and the \( \mathbf{a}_j \)’s are \( n \)-dimensional real vectors and the index sets \( M_1, M_2, M_3, N_1, \) and \( N_2 \) are finite.\(^3\) The vector \( \mathbf{q} \) is called the cost function of the linear program. The linear functional \( \mathbf{q}^T \mathbf{f} \) is the objective function or the cost function. The vectors \( \mathbf{a}_j \) are called constraint vectors, or simply constraints. Also called constraints are the (in)equalities of the form \( \mathbf{a}_j^T \mathbf{f} \geq b_j, \mathbf{a}_j^T \mathbf{f} \leq b_j, \) or \( \mathbf{a}_j^T \mathbf{f} = b_j \).

Any vector \( \mathbf{f} \) that satisfies all of the linear program’s constraints is called a feasible solution, and the set of all feasible solutions is the feasible set. A feasible solution that minimizes the objective function is an optimal feasible solution, or simply an optimal solution. Note that while the optimality of a vector \( \mathbf{f} \) depends on both the constraints and the cost function, the feasibility of a solution \( \mathbf{f} \) depends only on the constraints. Since the feasible set for a linear program is an intersection of closed half-spaces in \( \mathbb{R}^n \), it is a polyhedron. A polyhedron that is also bounded is called a polytope. When dealing with polyhedra in the context of linear programming, particular attention is paid to the extreme points.

\(^3\)In general, a linear program can be either a maximization or a minimization problem. There is, however, no essential loss of generality in using this convention: maximizing \( \mathbf{q}^T \mathbf{f} \) is equivalent to minimizing \( (-\mathbf{q})^T \mathbf{f} \).
Definition II.6 ([5], Definition 2.6). Let \( \mathcal{M} \) be a nonempty polyhedron in \( \mathbb{R}^n \). A point \( \omega \in \mathcal{M} \) is an extreme point provided that it is not in the convex hull of \( \mathcal{M} \setminus \{ \omega \} \). In other words, \( \omega \) is an extreme point of \( \mathcal{M} \) if it cannot be written as a convex sum \( \alpha y + (1 - \alpha)z \) for \( y, z \in \mathcal{M} \setminus \{ \omega \} \) and \( \alpha \in (0, 1) \).

The importance of extreme points is summarized in the next two theorems.

Theorem II.7 ([5], Theorem 2.7). Suppose that a linear program has feasible set \( \mathcal{M} \). If \( \mathcal{M} \) has an extreme point and an optimal solution to the linear program exists, then there exists an optimal solution that is an extreme point of \( \mathcal{M} \).

Theorem II.7 implies that we may assume that an optimal solution to a linear program occurs at an extreme point. In particular, when solving a linear program whose underlying polyhedron is bounded, an optimal solution is guaranteed to exist and it therefore suffices to consider only extreme points as candidate solutions. On the other hand, given an extreme point \( e \) of a polyhedron \( \mathcal{M} \) there is always some linear program with \( \mathcal{M} \) as its feasible set whose unique solution is \( e \):

Theorem II.8 ([5], Theorem 2.3). Let \( \mathcal{M} \) be a polyhedron in \( \mathbb{R}^n \). If \( \omega \) is an extreme point of \( \mathcal{M} \), then there exists a cost function \( q \in \mathbb{R}^n \) such that \( \omega \) is the unique optimal solution to the following linear program:

\[
\begin{align*}
\text{minimize} & \quad q^T f \\
\text{subject to} & \quad f \in \mathcal{M}.
\end{align*}
\]

In other words, \( q^T \omega < q^T f \) for all \( f \in \mathcal{M} \setminus \{ \omega \} \).

It is often convenient to represent extreme points algebraically instead of geometrically. We say that a constraint of the form \( a^T f \geq b, a^T f \leq b \), or \( a^T f = b \) is active at \( \omega \) if \( a^T \omega = b \), i.e., if the constraint is met with equality. If the polyhedron \( \mathcal{M} \) is defined by linear equality and inequality constraints and \( \omega \) is an element of \( \mathbb{R}^n \), then we say that \( \omega \) is a basic solution if all equality constraints of \( \mathcal{M} \) are satisfied and the set of active constraint vectors spans all of \( \mathbb{R}^n \). If \( \omega \) is a basic solution that satisfies all of the constraints, it is called a basic feasible solution. Using these algebraic notions, Theorem II.9 gives an alternate characterization of extreme points.

Theorem II.9 ([5], Theorem 2.3). Let \( \mathcal{M} \) be a nonempty polyhedron in \( \mathbb{R}^n \) defined by a given set of linear equality and inequality constraints, and let \( \omega \in \mathcal{M} \). The following are equivalent:

(a) \( \omega \) is an extreme point of \( \mathcal{M} \),
(b) \( \omega \) is a basic feasible solution of \( \mathcal{M} \).

Theorem II.9 relates a geometric invariant to the specific algebraic representation of a polyhedron. It also points to a method of finding extreme points: suppose that \( a_1, \ldots, a_n \) is a set of \( n \) linearly independent constraint vectors, and let \( b_1, \ldots, b_n \) be the associated constraint values. Let \( A \) be the matrix whose rows are \( a_1, \ldots, a_n \), and let \( b \) be a column vector whose entries are \( b_1, \ldots, b_n \). Since the rows of \( A \) are linearly independent, there is a solution to \( Af = b \), namely \( f = A^{-1}b \). By the construction of \( f \), it satisfies the constraints \( a_1, \ldots, a_n \) with equality. If \( f \) turns out to satisfy the rest of the constraints of the polyhedron, then \( f \) is a basic feasible solution and hence, by Theorem II.9, an extreme point.

Conversely, if \( f \) is an extreme point of a polyhedron, then Theorem II.9 implies that there exists a set of \( n \) linearly independent constraint vectors that are active at \( f \); call them \( a_1, \ldots, a_n \). With \( A \) and \( b \) as in the preceding paragraph, the extreme point \( f \) must be the unique solution to \( Af = b \). This interpretation of an extreme point as a solution to a linear system of equations gives the proof of the following well-known theorem.

Theorem II.10. Let \( \mathcal{M} \) be a non-empty polyhedron in \( \mathbb{R}^n \). If every constraint vector defining \( \mathcal{M} \) has rational coefficients, then every extreme point of \( \mathcal{M} \) is a rational vector.
C. The Fundamental Polytope

We now explicitly describe the fundamental polytope, which is the feasible region over which Feldman’s linear programming decoder operates (see Problem (3)).

Definition II.11 ([7]). Let C be a code with parity-check matrix $H$ and Tanner graph $G = (X \cup U, E)$. The fundamental polytope $P = P(H)$ is the set of all vectors $f \in \mathbb{R}^{|X|}$ satisfying the following constraints:

(a) $0 \leq f_x \leq 1$ for all $x \in X$, and

(b) $\sum_{x \in S} f_x + \sum_{x \in N(u) \setminus S} (1 - f_x) \leq |N(u)| - 1$ for all pairs $(u, S)$, where $u \in U$ and $S$ is a subset of $N(u)$ with odd cardinality.

Feldman [7] offers a heuristic justification for the constraints defining $P$, which we reproduce here for the sake of clarity. Since all codewords lie in $\{0, 1\}^n$, the components of a vector in $P$ should not be able to assume values outside of $[0, 1]$, but since linear programs require convexity we permit these variables to assume intermediate values. The other nontrivial family of constraints is best explained in the integer case, for then these constraints can be viewed as cutting planes which forbid “bad” configurations. To be explicit, suppose that $f \in \{0, 1\}^n$ is not a codeword. This means that there exists a row $r_j$ of the parity-check matrix $H$ such that $r_j \cdot f = 1$, which in turn implies that there is a check node $j \in \mathcal{J}$ of the Tanner graph $T$ such that $f$ assigns an odd number of ones to the variable nodes of $N(j)$ in $T$. Letting $S$ denote this subset, we see that $f$ is such that

$$\sum_{i \in S} f_i + \sum_{i \in N(j) \setminus S} (1 - f_i) = \sum_{i \in S} 1 + \sum_{i \in N(j) \setminus S} 1 = |S| + |N(j) \setminus S| = |N(j)| > |N(j)| - 1.$$ 

Hence, the non-codeword integer vector $f$ is excluded from $P$. Conversely, if $f \in \{0, 1\}^n$ is a codeword, then one can check that it satisfies every constraint of the fundamental polytope. These two results are summarized in Theorem II.12 below.

Theorem II.12 ([7]). Let C be a code with parity-check matrix $H$ and fundamental polytope $P = P(H)$. An integer vector $f$ is an element of $P$ if and only if $f \in C$.

Since all the constraints of the fundamental polytope are integer vectors, Theorem II.10 states that all extreme points of the fundamental polytope must be rational vectors. If an extreme point $f$ is an integer vector, Theorem II.12 states that $f$ must be a codeword. As mentioned in the introduction, there are often fractional extreme points of $P$ as well – see Example II.13. In Section III, we investigate the structure of the non-integral extreme points.

Example II.13. Let $C$ be the cycle code whose parity-check matrix is given in Example II.5 and whose Tanner graph is depicted in Figure 1 and let $P$ be the fundamental polytope of $C$. Theorem II.12 states that the set of integer extreme points of $P$ is precisely $C$. Since the dimension of $C$ is $k = 4$, it follows that $P$ contains $2^4 = 16$ integer extreme points. As is shown in [2] (and also in [7]), there are exactly four non-integral extreme points of $P$:

$$\left\{ \begin{array}{c} \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 0, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \\ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, 1, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \\ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, 0, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \\ \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \end{array} \right\}.$$
III. LP PSEUDOCODEWORDS OF CYCLE CODES ARE HALF-INTEGRAL

This section is dedicated to proving that linear programming pseudocodewords are half-integral. Along the way, we will see that LP pseudocodewords also have a close relationship with unions of vertex-disjoint simple cycles in the Tanner graph. This additional structure turns out to be inseparable from half-integrality. We make this latter fact explicit in Theorem III.11 which is the main result of this paper.

In order to prove Theorem III.11 we adopt the viewpoint suggested by Theorem II.9 – namely, we view an extreme point of a polytope as a solution to a linear system of equations. In our case, the equations forming this system come from the constraint vectors of the fundamental polytope that are satisfied with equality. We proceed as follows. We begin by prove a technical lemma pertaining to active constraints in Section III-A. This lemma is then applied to cycle codes\(^4\) in Section III-B giving a proof of Theorem III.11.

A. A Technical Lemma on Active Constraints

Let \( C \) be a code defined by parity-check matrix \( H \), and let \( G = (X \cup U, E) \) be its Tanner graph. We may rearrange the constraints defining the fundamental polytope to put them into the standard form presented in Section II-B

\[
\begin{align*}
  f_x & \leq 1 & \forall x & \in X \\
  f_x & \geq 0 & \forall x & \in X \\
  \sum_{x \in S} f_x + \sum_{x \in N(u) \setminus S} -f_x & \leq |S| - 1 & \forall u & \in U, S \subseteq N(u) & \text{such that } |S| \text{ is odd}
\end{align*}
\]

For convenience, we use the notation \((u, S)\) to represent the constraint vector of \( P \) corresponding to the check \( u \) and the odd-sized subset \( S \subseteq N(u) \).

**Definition III.1.** Let \( C \) be a code presented by a parity-check matrix \( H \). Let \( G = (X \cup U, E) \) be its Tanner graph, and let \( \mathcal{P} \) be its fundamental polytope. For a check node \( u \in U \), a subset \( S \subseteq N(u) \), and a vector \( f \in [0, 1]^n \), the cost of \( S \) at \( u \) relative to \( f \) is given by

\[
\kappa_{u,f}(S) = \sum_{x \in S} f_x + \sum_{x \in N(u) \setminus S} (1 - f_x).
\]

If there exists an odd-sized subset \( S \subseteq N(u) \) such that \( \kappa_{u,f}(S) = |N(u)| - 1 \), then we say that the check \( u \) is active at \( f \) and that the set \( S \) is active for \( f \) at \( u \).

Note that a vector \( f \in [0, 1]^n \) is in the fundamental polytope if and only if \( \kappa_{u,f}(S) \leq |N(u)| - 1 \) for all pairs \((u, S)\) such that \( S \) is an odd-sized subset of \( N(u) \). The next lemma follows directly from Definition III.1 in conjunction with the definition of the fundamental polytope.

**Lemma III.2.** Let \( C \) be a code presented by a parity-check matrix \( H \). Let \( G = (X \cup U, E) \) be its Tanner graph, and let \( \mathcal{P} \) be its fundamental polytope. If \( u \in U \) is active at some \( f \in \mathcal{P} \) and \( \alpha \in (0, 1) \), then \( f \) can assume the value of \( \alpha \) in at most \( \max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\} \) positions of \( N(u) \). In particular, \( f \) can assume the value of \( \frac{1}{2} \) at most twice in \( N(u) \).

**Proof:** Let \( D \subseteq X \) be the set of variable nodes in \( N(u) \) that are assigned a value of \( \alpha \) by \( f \), and let \( d := |D| \). Since \( u \) is active at \( f \), there exists an odd-sized subset \( S \subseteq N(u) \) such that \( S \) is active for \( f \) at \( u \). We consider two cases.

If \( \alpha \geq \frac{1}{2} \), then \( \alpha \geq 1 - \alpha \). We therefore have

\[
|N(u)| - 1 = \kappa_{u,f}(S) \leq |N(u)| - d(1 - \alpha),
\]

\(^4\)While this paper’s main result concerns linear programming pseudocodewords of cycle codes, Lemma III.3 can be applied to any binary linear code.
which implies that \( d \leq \frac{1}{1-\alpha} = \max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\} \).

On the other hand, if \( \alpha < \frac{1}{2} \), then \( \alpha < 1 - \alpha \). We therefore have

\[ |N(u)| - 1 = \kappa_{u,f}(S) \leq |N(u)| - d\alpha, \]

which implies that \( d \leq \frac{1}{\alpha} = \max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\} \).

Recalling that the symmetric difference between two sets \( S_1 \) and \( S_2 \) is \( S_1 \triangle S_2 := (S_1 \cup S_2) \setminus (S_1 \cap S_2) \), we now prove Lemma III.3, which is crucial in establishing Theorem III.1.

Lemma III.3. Let \( C \) be a code presented by a parity-check matrix \( H \). Let \( G = (X \cup U, E) \) be its Tanner graph, and let \( P \) be its fundamental polytope. Fix \( f \in P \) and \( u \in U \). If \( u \) is active at \( f \) and \( f \) is not integral on \( N(u) \) (i.e., there is some \( x \in N(u) \) such that \( 0 < f_x < 1 \)), then there are at most two active sets for \( f \) at \( u \).

Moreover, in the case that the number of active sets for \( f \) at \( u \) is exactly two, the symmetric difference of these two sets consists of exactly two variable nodes, and these two variable nodes are exactly the neighbors of \( u \) at which \( f \) is not integral.

Proof: Let \( f \in P \) and \( u \in U \) be given such that \( f^{N(u)} \) is not integral and \( u \) is active at \( f \). For each \( x \in N(u) \), the value \( f_x \) appears in the expression for \( \kappa_{u,f}(S) \) as \( f_x \) if \( x \in S \) or as \( 1 - f_x \) if \( x \in N(u) \setminus S \). The basic idea of this proof is to address the following question: how can \( f_x \) make the largest contribution to \( \kappa_{u,f}(S) \)? It is clear that \( f_x > 1 - f_x \) if and only if \( f_x > \frac{1}{2} \) and that \( f_x < 1 - f_x \) if and only if \( f_x < \frac{1}{2} \).

From this, we see that \( \kappa_{u,f}(S) \) is maximized exactly when \( S \) consists of all nodes \( x \) such that \( f_x > \frac{1}{2} \), and possibly some nodes \( x \) with \( f_x = \frac{1}{2} \). In a search for sets \( S \) that are active for \( f \) at \( u \), however, we can only consider those sets \( S \) whose cardinality is odd. This parity-based issue is highly dependent on \( f \) itself, about which we know little. We therefore consider several cases, each of which takes the greedy solutions that ignore parity and makes them into solutions that respect this parity condition.

Define the following sets:

- \( L := \{x \in N(u) : f_x < \frac{1}{2}\} \)
- \( \mathcal{E} := \{x \in N(u) : f_x = \frac{1}{2}\} \)
- \( \mathcal{G} := \{x \in N(u) : f_x > \frac{1}{2}\} \)
- \( \mathcal{Q} := \{x \in N(u) : |\frac{1}{2} - f_x| \leq |\frac{1}{2} - f_{x'}| \text{ for all } x' \in N(u)\} \).

The set \( \mathcal{Q} \) contains all nodes \( x \) such that \( f_x \) is closest to \( \frac{1}{2} \) among all \( x' \in N(u) \); note that \( \mathcal{Q} = \mathcal{E} \) if \( \mathcal{E} \neq \emptyset \).

Since \( u \) is active at \( f \), by Lemma III.2 we know that \( 0 \leq |\mathcal{E}| \leq 2 \). Suppose first that \( |\mathcal{E}| = 2 \), and write \( \mathcal{Q} = \mathcal{E} = \{q_1, q_2\} \). If \( \mathcal{G} \) is odd, then the sets \( S_1 := \mathcal{G} \) and \( S_2 := \mathcal{G} \cup \mathcal{Q} \) are the only two odd-sized subsets of \( N(u) \) that maximize \( \kappa_{u,f} \) over all odd-sized subsets of \( N(u) \). Moreover, since \( u \) is active at \( f \), the maximum value achieved by \( \kappa_{u,f} \) over all odd-sized subsets of \( N(u) \) is precisely \( |N(u)| - 1 \). Having \( |\mathcal{E}| = 2 \) therefore forces all values on \( N(u) \setminus \mathcal{Q} \) to be integral. Since \( S_1 \) and \( S_2 \) satisfy \( S_1 \triangle S_2 = \mathcal{Q} \) and \( |S_1 \triangle S_2| = 2 \), the lemma is proved in this case. If \( \mathcal{G} \) is even, then the sets \( S_1 := \mathcal{G} \cup \{q_1\} \) and \( S_2 := \mathcal{G} \cup \{q_2\} \) are the only two odd-sized subsets that maximize \( \kappa_{u,f} \). As in the previous case, the maximum value achieved by \( \kappa_{u,f} \) over all odd-sized subsets of \( N(u) \) is \( |N(u)| - 1 \), so \( |\mathcal{E}| = 2 \) again forces all values on \( N(u) \setminus \mathcal{Q} \) to be integral. Since \( S_1 \triangle S_2 = \mathcal{Q} \), we are done in this case as well.

Now suppose that \( |\mathcal{E}| = 1 \). If \( \mathcal{G} \) is odd, then \( S = \mathcal{G} \) is the unique maximizer of \( \kappa_{u,f} \) over all odd-sized subsets of \( N(u) \). If \( \mathcal{G} \) is even, then \( S = \mathcal{G} \cup \mathcal{E} \) is the unique maximizer of \( \kappa_{u,f} \) over all odd-sized subsets of \( N(u) \). In either situation there is at most one active set at \( u \), and so the lemma is proved in this case.

Finally, assume that \( |\mathcal{E}| = 0 \). If \( \mathcal{G} \) is odd, then \( S = \mathcal{G} \) is the unique maximizer of \( \kappa_{u,f} \) over all odd-sized subsets of \( N(u) \) and the lemma is proved. If \( \mathcal{G} \) is even, then the collection of all active sets for \( f \) at \( u \) is given by \( \{S_q | q \in \mathcal{Q}\} \), where \( S_q \) is defined as follows:

\[
S_q := \begin{cases} 
\mathcal{G} \cup \{q\} & \text{if } q \notin \mathcal{G} \\
\mathcal{G} \setminus \{q\} & \text{if } q \in \mathcal{G}.
\end{cases}
\]
If \(|Q| = 1\), there is a unique set \(S_q\) that maximizes \(\kappa_{u,f}(S)\) and we are done. So suppose that \(|Q| > 1\), and let \(q_1\) and \(q_2\) be distinct elements of \(Q\). By the definition of \(Q\), we have \(f_{q_1} = f_{q_2}\) if \(q_1, q_2 \in \mathcal{L}\) or \(q_1, q_2 \in \mathcal{G}\), and \(f_{q_1} = 1 - f_{q_2}\) if \(q_1 \in \mathcal{L}\) and \(q_2 \in \mathcal{G}\) or vice-versa. If \(q_1, q_2 \in \mathcal{L}\), then

\[
|N(u)| - 1 = \kappa_{u,f}(S_{q_1}) = \sum_{x \in S_q} f_x + \sum_{x \in N(u) \setminus S_q} (1 - f_x) = \sum_{x \in S_q \setminus \{q_1\}} f_x + f_{q_1} + (1 - f_{q_2}) + \sum_{x \in N(u) \setminus (S_q \cup \{q_2\})} (1 - f_x) = \sum_{x \in S_q \setminus \{q_1\}} f_x + 1 + \sum_{x \in N(u) \setminus (S_q \cup \{q_2\})} (1 - f_x).
\]

We conclude that \(f\) must be integral on \(N(u) \setminus \{q_1, q_2\}\); otherwise, it would not be possible for \(\kappa_{u,f}(S_{q_1})\) to attain the value \(|N(u)| - 1\). A similar argument yields the same conclusion in each of the other three cases.

By assumption we have that \(f\) must be non-integral in at least one position of \(N(u)\), so either \(0 < f_{q_1} < 1\) or \(0 < f_{q_2} < 1\). It follows from this and from the definition of \(Q\) that \(Q = \{q_1, q_2\}\) and that \(0 < f_{q_1}, f_{q_2} < 1\). Thus, there are exactly two active sets \(S_{q_1}\) and \(S_{q_2}\), these sets satisfy \(|S_{q_1} \triangle S_{q_2}| = 2\), and \(S_{q_1} \triangle S_{q_2}\) is exactly the set of indices where \(f\) is not integral.

**B. A Proof of Theorem III.11**

Let \(C\) be a code with parity-check matrix \(H\), Tanner graph \(G = (X \cup U, E)\), and fundamental polytope \(\mathcal{P}\). Let \(\omega \in \mathcal{P}\) be an LP pseudocodeword, i.e., an extreme point of \(\mathcal{P}\). By Theorem II.9, \(\omega\) is a basic feasible solution. Thus, the span of the constraint vectors of \(\mathcal{P}\) that are active at \(\omega\) has dimension \(n := |X|\). For each node to which \(\omega\) assigns either a 0 or a 1 there is an active constraint of the form (4) or (5). This set of constraint vectors is linearly independent, so we may extend it by other active constraint vectors, which will necessarily be of the form \((u, S)\), to obtain a set of \(n\) linearly independent constraint vectors that are active at \(\omega\). Up to a permutation of rows and columns, we may write these \(n\) constraint vectors in matrix form as

\[
L_{\omega} := \begin{bmatrix} I_{n-m} & 0 \\ Q_{\omega} & R_{\omega} \end{bmatrix}
\]

where the last \(m\) columns represent the variable nodes in \(\mathcal{F}_{\omega} := \{x \mid 0 < \omega_x < 1\}\). Note that since the last \(m\) rows of \(L_{\omega}\) come from constraints of the form \((u, S)\), each entry of \(L_{\omega}\) is either \(-1, 0,\) or \(+1\). Note also that since \(\det(L_{\omega}) = \det(R_{\omega})\), the fact that \(L_{\omega}\) is invertible implies that the square submatrix \(R_{\omega}\) is also invertible.

Since each row of \(L_{\omega}\) is a constraint vector that is active at \(\omega\), we have that \(L_{\omega}\omega = z_{\omega}\), where \(z_{\omega}\) is an integer vector determined from the right-hand sides of constraints (4) – (6). If we knew that \(L_{\omega}^{-1}\) takes entries only from \(\frac{1}{2} \mathbb{Z}\), we could write \(\omega = L_{\omega}^{-1}z_{\omega}\) and conclude that \(\omega \in \{0, \frac{1}{2}, 1\}^n\). Our next goal is to show that, in the case of cycle codes, \(L_{\omega}^{-1}\) has entries only in \(\frac{1}{2} \mathbb{Z}\). Because of the block structure of \(L_{\omega}\), this amounts to showing that \(R_{\omega}^{-1}\) has entries only in \(\frac{1}{2} \mathbb{Z}\). To show this algebraic fact about \(R_{\omega}\), we turn to graphical methods.

**Definition III.4.** Let \(C\) be a code with parity-check matrix \(H\), Tanner graph \(G = (X \cup U, E)\) and fundamental polytope \(\mathcal{P}\). Let \(\omega\) be a nontrivial linear programming pseudocodeword, define \(\mathcal{F}_{\omega} := \{x \in X \mid 0 < \omega_x < 1\}\), and set \(m = |\mathcal{F}_{\omega}|\). Let \(R_{\omega}\) be an \(m \times m\) matrix formed as above, and define \(U_{\omega}\) to be the set of all check nodes \(u \in U\) such that a constraint of the form \((u, S)\) is represented in the rows of
Let at least two distinct elements of \( x \) and the set of erased bits contains a stopping set \([6]\).

**Lemma III.5** (see also [2], Theorem 7.1). Let \( C \) be a code with parity-check matrix \( H \), Tanner graph \( G = (X \cup U, E) \) and fundamental polytope \( \mathcal{P} \). Fix \( f \in \mathcal{P} \) and \( u \in U \). If \( u \) is incident to a variable node \( x \in \mathcal{F}_f := \{ x \in X \mid 0 < f_x < \} \), then it is incident to at least two such nodes in \( \mathcal{F}_f \).

**Proof:** Assume that \( u \) is incident to one and only one variable node \( x_0 \in \mathcal{F}_f \). Then every variable node in \( N(u) \setminus \{x_0\} \) is assigned a value of 0 or 1 by \( f \). Let \( \mathcal{O}_u := \{ x \in N(u) \mid f_x = 1 \} \). If \( |\mathcal{O}_u| \) is even, then \( S := \mathcal{O}_u \cup \{x_0\} \) has odd cardinality. Notice that \( \kappa_{u,f}(S) > |S| + |N(u) \setminus S| - 1 = |N(u)| - 1 \), which means that \( f \) does not satisfy constraint \((u, S)\) of \( \mathcal{P} \) given by

\[
\kappa_{u,f}(S) = \sum_{x \in S} f_x + \sum_{x \in N(u) \setminus S} (1 - f_x) \leq |N(u)| - 1.
\]

Hence, \( f \) is not an element of \( \mathcal{P} \).

If \( |\mathcal{O}_u| \) is odd, set \( S = \mathcal{O}_u \) and observe that \( \kappa_{u,f}(S) > |S| + |N(u) \setminus S| - 1 = |N(u)| - 1 \). Again, \( f \) fails to satisfy constraint \((u, S)\) of \( \mathcal{P} \), so \( f \) is not an element of \( \mathcal{P} \).

**Remark III.6.** Lemma III.3 implies that for any \( f \in \mathcal{P}, \mathcal{F}_f \) is a stopping set. By definition, a stopping set is a set of variable nodes \( V \) such that if a check \( u \) is adjacent to some \( v \in V \), then it is adjacent to at least two distinct elements of \( V \). Stopping sets are significant in the study of iterative message-passing decoding on the binary erasure channel: the belief propagation algorithm fails to decode if and only if the set of erased bits contains a stopping set \([6]\).

We now restrict to the case where \( C \) is a cycle code.

**Lemma III.7.** Let \( C \) be a cycle code with Tanner graph \( G = (X \cup U, E) \) and fundamental polytope \( \mathcal{P} \). Let \( L_\omega \) and \( R_\omega \) be matrices formed as above, and let \( \mathcal{F}_\omega \) and \( U_\omega \) be the subsets of \( X \) and \( U \), respectively, as given in Definition III.4. For any nontrivial linear programming pseudocodeword \( \omega \in \mathcal{P} \) we have \( U_\omega \subseteq N(\mathcal{F}_\omega), |\mathcal{F}_\omega| = |U_\omega| \), and \( G_{R_\omega} \) is 2-regular.

**Proof:** Let \( Q_\omega \) be given as above, so that

\[
L_\omega = \begin{bmatrix} I_{n-m} & 0 \\ Q_\omega & R_\omega \end{bmatrix}.
\]

We first show that \( U_\omega \subseteq N(\mathcal{F}_\omega) \). Let \( u \in U_\omega \) be given. There must be a corresponding \( S \subseteq N(u) \) such that \((u, S)\) is a row of \( L_\omega \). Since \( R_\omega \) is non-singular, this row must involve some variable nodes in \( \mathcal{F}_\omega \).

Thus, \( u \in N(\mathcal{F}_\omega) \), so \( U_\omega \subseteq N(\mathcal{F}_\omega) \).

We now show that \(|U_\omega| = |\mathcal{F}_\omega| \). This amounts to showing that no check is represented in the rows of \([Q_\omega R_\omega]\) more than once. Clearly, a single constraint vector \((u, S)\) associated with \( u \) cannot appear in the rows of \([Q_\omega R_\omega]\) more than once, since otherwise \( R_\omega \) would not be invertible.

Suppose now that for some \( u \in U_\omega \) there are two distinct subsets \( S_1, S_2 \subseteq N(u) \) such that the constraints \((u, S_1)\) and \((u, S_2)\) are both rows of \([Q_\omega R_\omega]\). Then both \( S_1 \) and \( S_2 \) are active for \( \omega \) at \( u \). In general, the vectors \((u, S_1)\) and \((u, S_2)\) are such that \((u, S_1) = -(u, S_2)\) when we restrict to those positions of \( S_1 \triangle S_2 \), and \((u, S_1) = (u, S_2) = 0\) on the positions of \( X \setminus N(u) \). By applying Lemma III.3 to this check node, we see that \( S_1 \triangle S_2 = \{ x \in N(u) \mid 0 < \omega_x < 1 \} \). This means that \((u, S_1) = -(u, S_2)\) on \( S_1 \triangle S_2 = N(u) \cap \mathcal{F}_\omega \) and \((u, S_1) = (u, S_2) = 0\) on \( \mathcal{F}_\omega \setminus N(u) \). It follows that the projections of \((u, S_1)\) and \((u, S_2)\) onto the positions of \( \mathcal{F}_\omega \) are scalar multiples of one another, which contradicts in the fact that \( R_\omega \) is invertible. We conclude that \(|U_\omega| = |\mathcal{F}_\omega| \).

To prove the third and final claim of the lemma, we bound the number \( e \) of edges in \( G_{R_\omega} \) in two different ways. Since \( C \) is a cycle code, each variable node has degree exactly 2 in \( G \), so every variable
node in \(G_{R_\omega}\) has degree at most 2. Thus, \(e \leq 2|F_\omega|\). On the other hand, since \(U_\omega \subseteq N(F_\omega)\), Lemma II.5 yields \(e \geq 2|U_\omega|\). Since \(|F_\omega| = |U_\omega|\), the result follows.

**Proposition III.8.** Let \(C\) be a cycle code with fundamental polytope \(\mathcal{P}\). For any nontrivial linear programming pseudocodeword \(\omega \in \mathcal{P}\) and any corresponding matrix \(R_\omega\), the matrix \(R_\omega^{-1}\) has entries only in \(\{-\frac{1}{2}, 0, +\frac{1}{2}\}\).

**Proof:** By Definition II.4 the non-zero entries of \(R_\omega\) give the incidence structure of \(G_{R_\omega}\). More formally, the matrix \(|R_\omega|\) obtained by taking the coordinate-wise absolute value of each entry in \(R_\omega\) is the bipartite incidence matrix for \(G_{R_\omega}\). By Lemma II.7 \(G_{R_\omega}\) is a 2-regular bipartite graph; therefore, each connected component of \(G_{R_\omega}\) is a cycle of even length. Using this connection between \(R_\omega\) and \(G_{R_\omega}\), we may assume (by permuting rows and columns) that \(R_\omega\) is a block diagonal matrix

\[
R_\omega = \begin{bmatrix}
D_1 & 0 & \cdots & 0 \\
0 & D_2 & \cdots & 0 \\
0 & 0 & \ddots & \vdots \\
0 & \ddots & \ddots & D_b
\end{bmatrix}
\]

where each block is square and has the form

\[
D = \begin{bmatrix}
d_{1,1} & d_{1,2} & 0 & \cdots & 0 \\
0 & d_{2,2} & d_{2,3} & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & d_{\ell-1,\ell-1} & d_{\ell-1,\ell} \\
d_{\ell,1} & 0 & \cdots & 0 & d_{\ell,\ell}
\end{bmatrix}
\]

with \(d_{p,q} \in \{-1, +1\}\) for all \(p, q\). To show that \(R_\omega^{-1}\) has entries only in \(\{-\frac{1}{2}, 0, +\frac{1}{2}\}\), it suffices to show that each block \(D\) of \(R_\omega\) is such that \(D^{-1}\) takes entries only in \(\{-\frac{1}{2}, 0, +\frac{1}{2}\}\). In fact, we will show that \(D^{-1}\) takes entries only in \(\{-\frac{1}{2}, +\frac{1}{2}\}\).

Let \(D\) be a block of \(R_\omega\). Since \(R_\omega\) is invertible, \(D\) is also invertible. This implies the existence of a unique solution \(a\) to the equation \(Da = \epsilon_p\), where \(\epsilon_p\) is the \(p\)th standard basis vector. The equation \(Da = \epsilon_p\) gives rise to a set of relations that must hold between the entries of \(a\).

In the following we take subscripts modulo \(\ell\) to respect the cyclic nature of \(D\). For all \(q \neq p\), we have \(a_q d_{q,q} + a_{q+1} d_{q,q+1} = 0\). Since \(d_{q,q+1}, d_{q,q} \in \{-1, +1\}\) we may rearrange to get \(a_q = -\frac{d_{q,q+1}}{d_{q,q}} a_{q+1}\), and so \(a_q = \pm a_{q+1}\) for all \(q \neq p\). This in turn implies that all of the entries of \(a\) are the same up to sign.

The equation \(Da = \epsilon_p\) also implies that \(a_p d_{p,p} + a_{p+1} d_{p,p+1} = 1\). We know from the previous paragraph that \(a_{p+1}\) is either \(a_p\) or \(-a_p\). Thus, we have to consider two possible equations: \(a_p (d_{p,p} + d_{p,p+1}) = 1\) or \(a_p (d_{p,p} - d_{p,p+1}) = 1\). In either case, since \(d_{p,p+1}, d_{p,p} \in \{-1, +1\}\) we have that \(a_p\) is either \(-\frac{1}{2}\) or \(+\frac{1}{2}\).

Combining this with the previous paragraph, we see that all the entries of \(a\) are in \(\{-\frac{1}{2}, +\frac{1}{2}\}\).

Renaming the unique solution to the equation \(Da = \epsilon_p\) to be \(a_p\), we see that

\[
D^{-1} = \begin{bmatrix}
a_1 & | & \cdots & | \\
| & a_2 & \cdots & | \\
| & \cdots & \cdots & | \\
\end{bmatrix}
\]

It follows that \(D^{-1}\) takes entries only in \(\{-\frac{1}{2}, +\frac{1}{2}\}\). 

The next corollary follows immediately from Proposition III.8 and the discussion surrounding the introduction of \(L_\omega\).

**Corollary III.9.** Let \(C\) be a cycle code with fundamental polytope \(\mathcal{P}\). If \(\omega\) is an extreme point of \(\mathcal{P}\) and \(L_\omega\) is formed from \(\omega\) as above, then \(L_\omega^{-1}\) has entries only in \(\frac{1}{2}\mathbb{Z}\).

We now have the tools to prove the following proposition.
Proposition III.10. Let \( C \) be a cycle code of length \( n \) with Tanner graph \( G = (X \cup U, E) \) and fundamental polytope \( \mathcal{P} \). If \( \omega \) is a nontrivial linear programming pseudocodeword of \( \mathcal{P} \), then \( \omega \in \{0, \frac{1}{2}, 1\}^n \) and the subgraph \( G_{\mathcal{F}_\omega} \) of \( G \) induced by the set \( \mathcal{F}_\omega = \{ x \mid \omega_x = \frac{1}{2} \} \) and its neighborhood \( N(\mathcal{F}_\omega) \) is precisely \( G_{R\omega} \), which is 2-regular.

Proof: Let \( L_\omega, R_\omega, \) and \( U_\omega \) be given as above. By Corollary III.9 \( L_\omega^{-1} \) has entries in \( \frac{1}{2}\mathbb{Z} \). By the discussion following the introduction of \( L_\omega \) earlier in this section, we have that \( L_\omega \omega = z_\omega \), where \( z_\omega \) is an integer vector. Thus, \( \omega = L^{-1}z \) must have entries that come only from \( \frac{1}{2}\mathbb{Z} \). Since any point in the fundamental polytope can only assume values between 0 and 1, \( \omega \in \{0, \frac{1}{2}, 1\}^n \).

Lemma III.7 implies that \( \mathcal{U}_\omega \subseteq N(\mathcal{F}_\omega), |\mathcal{F}_\omega| = |U_\omega|, \) and \( G_{R\omega} \) is 2-regular. From \( U_\omega \subseteq N(\mathcal{F}_\omega) \) and Definition III.4 we see that \( G_{R\omega} \) is a subgraph of \( G_{\mathcal{F}_\omega} \). Letting \( e \) denote the number of edges in \( G_{\mathcal{F}_\omega} \), we have that \( e = 2|\mathcal{F}_\omega| \) since \( C \) is a cycle code. Lemma III.5 implies that \( 2|N(\mathcal{F}_\omega)| \leq e \). Using the fact that \( U_\omega \subseteq N(\mathcal{F}_\omega) \), we have

\[
2|U_\omega| \leq 2|N(\mathcal{F}_\omega)| \leq e = 2|\mathcal{F}_\omega|.
\]

Since \( |\mathcal{F}_\omega| = |U_\omega| \), these inequalities must be tight. Thus \( G_{\mathcal{F}_\omega} \) has the same number of edges as \( G_{R\omega} \) and contains \( G_{R\omega} \) as a subgraph. We conclude that \( G_{\mathcal{F}_\omega} = G_{R\omega} \).

We may now prove the main result of this paper.

Theorem III.11. Let \( C \) be a cycle code with Tanner graph \( G = (X \cup U, E) \) and fundamental polytope \( \mathcal{P} \). If \( \omega \in \mathcal{P} \) is a linear programming pseudocodeword of \( \mathcal{P} \), then the following two conditions hold:

(a) \( \omega \in \{0, \frac{1}{2}, 1\}^{|X|} \), and

(b) with \( \mathcal{H} := \{ x \in X \mid \omega_x = \frac{1}{2} \} \), the subgraph \( \Gamma \) of \( G \) induced by \( \mathcal{H} \cup N(\mathcal{H}) \) is 2-regular. Equivalently, \( \Gamma \) is a union of vertex-disjoint simple cycles \( \gamma_1, \gamma_2, \ldots, \gamma_\ell \).

Proof: Let \( \omega \) be a linear programming pseudocodeword of \( \mathcal{P} \). If \( \omega \) is a codeword, the two conditions in the statement of the theorem are satisfied and we are done. If, on the other hand, \( \omega \) is a nontrivial linear programming pseudocodeword, we may apply Proposition III.10 to conclude that \( \omega \) satisfies the two conditions.

IV. Conclusion

We have shown that a vector \( \omega \) can be a linear programming pseudocodeword of a cycle code only if \( \omega \) is half-integral. Moreover, the subgraph of the Tanner graph induced by the variable nodes assigned a value of \( \frac{1}{2} \) by \( \omega \), along with their neighboring check nodes, must be a union of vertex-disjoint simple cycles. These necessary conditions, however, are not sufficient to characterize LP pseudocodewords for cycle codes. The results from this paper are extended in [2], where the author and his collaborator provide complete graphical characterizations of both linear programming pseudocodewords and minimal linear programming pseudocodewords (see [9]) for the class of cycle codes.

Finally, we note that Lemma III.3, which was an essential ingredient in our proof of Theorem III.11, can be applied to any binary linear code. Testing whether this lemma can shed light on the pseudocodeword structure of additional families of codes is an object of future pursuit.

ACKNOWLEDGEMENT

The results from this paper are taken from the author’s Ph.D. dissertation [1]. The author thanks his advisor Judy L. Walker for her support and encouragement.

REFERENCES

[1] N. Axvig. Applications of Linear Programming to Coding Theory. PhD thesis, Department of Mathematics, University of Nebraska-Lincoln, 2010.

[2] N. Axvig and D. Dreher. Graphical characterizations of linear programming pseudocodewords for cycle codes. Submitted to IEEE Transactions on Information Theory.
[3] N. Axvig, D. Dreher, K. Morrison, E. Psota, L.C. Pérez, and J.L. Walker. Analysis of connections between pseudocodewords. *IEEE Transactions on Information Theory*, 55(9):4099–4107, September 2009.

[4] E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg. On the inherent intractibility of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–6, May 1978.

[5] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[6] D. Changyan, D. Proetti, I.E. Telatar, T.J. Richardson, and R.L. Urbanke. Finite length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Transactions on Information Theory*, 48:1570–1579, June 2002.

[7] J. Feldman. *Decoding Error-Correcting Codes via Linear Programming*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[8] R. Koetter, W.-C.W. Li, P.O. Vontobel, and J.L. Walker. Characterizations of pseudo-codewords of LDPC codes. *Advances in Mathematics*, 213:205–229, 2007.

[9] P.O. Vontobel and R. Koetter. Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes. CoRR, [http://arxiv.org/abs/cs/0512078](http://arxiv.org/abs/cs/0512078), Dec. 2005.

[10] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1996.