Polytope Representations for Linear-Programming Decoding of Non-Binary Linear Codes

Vitaly Skachek
Claude Shannon Institute
University College Dublin
Belfield, Dublin 4, Ireland
vitaly.skachek@ucd.ie

Mark F. Flanagan
DEIS
University of Bologna
47023 Cesena, Italy
mark.flanagan@ieee.org

Eimear Byrne
School of Mathematical Sciences
University College Dublin
Belfield, Dublin 4, Ireland
ebyrne@ucd.ie

Marcus Greferath
School of Mathematical Sciences
University College Dublin
Belfield, Dublin 4, Ireland
marcus.greferath@ucd.ie

Abstract—In previous work, we demonstrated how decoding of a non-binary linear code could be formulated as a linear-programming problem. In this paper, we study different polytopes for use with linear-programming decoding, and show that for many classes of codes these polytopes yield a complexity advantage for decoding. These representations lead to polynomial-time decoders for a wide variety of classical non-binary linear codes.

I. INTRODUCTION

In [1] and [2], the decoding of binary LDPC codes using linear-programming decoding was proposed, and the connections between linear-programming decoding and classical belief propagation decoding were established. In [3], the approach of [2] was extended to coded modulation, in particular to codes over rings mapped to non-binary modulation signals. In both cases, the principal advantage of the linear-programming framework is its mathematical tractability [2], [3].

For the binary coding framework, alternative polytope representations were studied which gave a complexity advantage for decoding. These representations lead to polynomial-time decoders for a wide variety of classical non-binary linear codes.

II. LINEAR-PROGRAMMING DECODING

Consider codes over finite quasi-Frobenius rings (this includes codes over finite fields, but may be more general). Denote by $\mathcal{R}$ such a ring with $q$ elements, by $0$ its additive identity, and let $\mathcal{R}^- = \mathcal{R} \setminus \{0\}$. Let $C$ be a linear code of length $n$ over $\mathcal{R}$ with $m \times n$ parity-check matrix $H$.

Denote the set of column indices and the set of row indices of $H$ by $I = \{1, 2, \ldots, n\}$ and $J = \{1, 2, \ldots, m\}$, respectively. The notation $H_{j,i}$ will be used for the $j$-th row of $H$. Denote by $\text{supp}(c)$ the support of a vector $c$. For each $j \in J$, let $I_j = \text{supp}(H_{j,:})$ and $d_j = |I_j|$, and let $d = \max_{j \in J} \{d_j\}$.

Given any $c \in \mathcal{R}^n$, parity check $j \in J$ is satisfied by $c$ if and only if the following equality holds over $\mathcal{R}$:

$$\sum_{i \in I_j} c_i \cdot H_{j,i} = 0. \quad (1)$$

For $j \in J$, define the single parity check code $C_j$ by

$$C_j = \{ (b_i)_{i \in I_j} : \sum_{i \in I_j} b_i \cdot H_{j,i} = 0 \}$$

Note that while the symbols of the codewords in $C$ are indexed by $I$, the symbols of the codewords in $C_j$ are indexed by $I_j$. Observe that $c \in C$ if and only if all parity checks $j \in J$ are satisfied by $c$.

Assume that the codeword $\bar{c} = (c_1, c_2, \ldots, c_n) \in C$ has been transmitted over a $q$-ary input memoryless channel, and a corrupted word $y = (y_1, y_2, \ldots, y_n) \in \Sigma^n$ has been received. Here $\Sigma$ denotes the set of channel output symbols. In addition, assume that all codewords are transmitted with equal probability.

For vectors $f \in \mathbb{R}^{q-1}n$, the notation

$$f = (f_1 | f_2 | \cdots | f_n),$$

will be used, where

$$\forall i \in I, f_i = (f_i^{(\alpha)})_{\alpha \in \mathcal{R}^-}.$$ We also define a function $\lambda : \Sigma \longrightarrow (\mathbb{R} \cup \{-\infty\})^{q-1}$ by

$$\lambda^{(\alpha)}(y) = \log \left( \frac{p(y | c)}{\sum_{\alpha \in \mathcal{R}^-} p(y | \alpha) \cdot \lambda^{(\alpha)}(y)} \right),$$

and $p(y | c)$ denotes the channel output probability (density) conditioned on the channel input. Extend $\lambda$ to a map on $\Sigma^n$ by $\lambda(y) = (\lambda(y_1) \; | \; \lambda(y_2) \; | \; \ldots \; | \; \lambda(y_n))$.

The LP decoder in [3] performs the following cost function minimization:

$$(\hat{f}, \hat{w}) = \arg \min_{(f,w) \in Q} \lambda(y) f^T, \quad (2)$$

where the polytope $Q$ is a relaxation of the convex hull of all points $f \in \mathbb{R}^{(q-1)n}$, which correspond to codewords; this
polytope is defined as the set of \( f \in \mathbb{R}^{(q-1)n} \), together with the auxiliary variables
\[
w_{j,b} \quad \text{for } j \in J, b \in C_j,
\]
which satisfy the following constraints:
\[
\forall j \in J, \forall b \in C_j, \quad w_{j,b} \geq 0,
\]
\[
\forall j \in J, \sum_{b \in C_j} w_{j,b} = 1,
\]
and
\[
\forall j \in J, \forall i \in I_j, \forall \alpha \in \mathbb{R}^-,
\]
\[
f_i^{(\alpha)} = \sum_{b \in C_j, b_i = \alpha} w_{j,b}.
\]

The minimization of the objective function (2) over \( \mathcal{Q} \) forms the relaxed LP decoding problem. The number of variables and constraints for this LP are upper-bounded by \( n(q-1)+mqd^{-1} \) and \( ml(q^d-1) + d(q-1) + 1 \) respectively.

It is shown in [3] that if \( f \) is integral, the decoder output corresponds to the maximum-likelihood (ML) codeword. Otherwise, the decoder outputs an ‘error’.

### III. NEW LP DESCRIPTION

The results in this section are a generalization of the high-density polytope representation [2, Appendix II]. Recall that the ring \( \mathcal{R} \) contains \( q-1 \) non-zero elements. Correspondingly, for vectors \( k \in \mathbb{N}^{q-1} \), we adopt the notation
\[
k = (k_\alpha)_{\alpha \in \mathcal{R}^-}
\]
Now, for any \( j \in J \), we define the mapping
\[
\kappa_j : C_j \longrightarrow \mathbb{N}^{q-1},
\]
\[
b \mapsto \kappa_j(b)
\]
defined by
\[
(\kappa_j(b))_\alpha = |\{ i \in I_j : b_i \cdot H_{j,i} = \alpha \}|
\]
for all \( \alpha \in \mathcal{R}^- \). We may then characterize the image of \( \kappa_j \), which we denote by \( T_j \), as
\[
T_j = \left\{ k \in \mathbb{N}^{q-1} : \sum_{\alpha \in \mathcal{R}^-} \alpha \cdot k_\alpha = 0 \quad \text{and} \quad \sum_{\alpha \in \mathcal{R}^-} k_\alpha \leq d_j \right\},
\]
for each \( j \in J \), where, for any \( k \in \mathbb{N}, \alpha \in \mathcal{R}, \)
\[
\alpha \cdot k = \begin{cases} 0 & \text{if } k = 0 \\ \alpha + \cdots + \alpha & \text{if } k > 0 \ (k \text{ terms in sum}) \end{cases}
\]
The set \( T_j \) is equal to the set of all possible vectors \( \kappa_j(b) \) for \( b \in C_j \).

Note that \( \kappa_j \) is not a bijection, in general. We say that a local codeword \( b \in C_j \) is \( k \)-constrained over \( C_j \) if \( \kappa_j(b) = k \).

Next, for any index set \( \Gamma \subseteq \mathcal{I} \), we introduce the following definitions. Let \( N = |\Gamma| \). We define the single-parity-check-code, over vectors indexed by \( \Gamma \), by
\[
C_\Gamma = \left\{ \alpha = (a_i)_{i \in \Gamma} \in \mathbb{R}^N : \sum_{i \in \Gamma} a_i = 0 \right\}.
\]

Also define a mapping \( \kappa_\Gamma : C_\Gamma \longrightarrow \mathbb{N}^{q-1} \) by
\[
(\kappa_\Gamma(\alpha))_\alpha = |\{ i \in \Gamma : a_i = \alpha \}|
\]
and define, for \( k \in T_j \),
\[
C_\Gamma(k) = \{ \alpha \in C_\Gamma : \kappa_\Gamma(\alpha) = k \}.
\]

Below, we define a new polytope for decoding. Recall that \( y = (y_1, y_2, \ldots, y_n) \in \Sigma^n \) stands for the received (corrupted) word. In the sequel, we make use of the following variables:

- For all \( i \in \mathcal{T} \) and all \( \alpha \in \mathcal{R}^- \), we have a variable \( f_i^{(\alpha)} \). This variable is an indicator of the event \( y_i = \alpha \).
- For all \( j \in \mathcal{J} \) and \( k \in T_j \), we have a variable \( \sigma_{j,k} \). Similarly to its counterpart in [2], this variable indicates the contribution to parity check \( j \) of \( k \)-constrained local codewords over \( C_j \).

Motivated by these variable definitions, for all \( j \in \mathcal{J} \) we impose the following set of constraints:
\[
\forall i \in I_j, \forall \alpha \in \mathcal{R}^-,
\]
\[
f_i^{(\alpha)} = \sum_{k \in T_j} z_{i,j,k}^{(\alpha)} = 1.
\]
\[
\forall k \in T_j, \forall \alpha \in \mathcal{R}^-,
\]
\[
\sum_{i \in I_j, \beta \in \mathcal{R}^-, \beta H_{i,j} = \alpha} z_{i,j,k}^{(\beta)} = k_\alpha \cdot \sigma_{j,k}.
\]
\[
\forall i \in I_j, \forall k \in T_j, \forall \alpha \in \mathcal{R}^-,
\]
\[
z_{i,j,k}^{(\alpha)} \geq 0.
\]
\[
\forall i \in I_j, \forall k \in T_j, \forall \alpha \in \mathcal{R}^-,
\]
\[
\sum_{\alpha \in \mathcal{R}^-} \sum_{\beta \in \mathcal{R}^-} z_{i,j,k}^{(\beta)} \leq \sigma_{j,k}.
\]

We note that the further constraints
\[
\forall i \in \mathcal{T}, \forall \alpha \in \mathcal{R}^-,
\]
\[
0 \leq f_i^{(\alpha)} \leq 1,
\]
\[
\forall j \in \mathcal{J}, \forall k \in T_j,
\]
\[
0 \leq \sigma_{j,k} \leq 1,
\]
and
\[
\forall j \in \mathcal{J}, \forall i \in I_j, \forall k \in T_j, \forall \alpha \in \mathcal{R}^-,
\]
\[
z_{i,j,k}^{(\alpha)} \leq \sigma_{j,k},
\]
follow from constraints \( \text{(7)-(11)} \). We denote by \( \mathcal{U} \) the polytope formed by constraints \( \text{(7)-(11)} \).

Let \( T = \max_{j \in J} |T_j| \). Then, upper bounds on the number of variables and constraints in this LP are given by \( n(q-1) + mn(q-1)+T \) and \( m(d(q-1)+1)+m((d+1)(q-1)+d)T \), respectively. Since \( T \leq (\frac{q}{d}+q-1) \) the number of variables and constraints are \( O(mq^{d-1}) \), which, for many families of codes, is significantly lower than the corresponding complexity for polytope \( \mathcal{Q} \).
For notational simplicity in proofs in this paper, it is convenient to define a new set of variables as follows:

\[ \forall j \in \mathcal{J}, \forall i \in \mathcal{I}_j, \forall k \in \mathcal{T}_j, k \in \mathcal{T}_j, \forall \alpha \in \mathcal{R}^- \cdot \]

\[ x_{i,j,k}^{(\alpha)} = \sum_{\beta \in \mathcal{R}^-, \beta \mathcal{H}_{i,j} = \alpha} z_{i,j,k}^{(\beta)} . \quad (15) \]

Then constraints (9) and (11) may be rewritten as

\[ \forall j \in \mathcal{J}, k \in \mathcal{T}_j, \forall \alpha \in \mathcal{R}^-, \quad \sum_{i \in \mathcal{I}_j} x_{i,j,k}^{(\alpha)} = k_\alpha \cdot \sigma_{j,k} , \quad (16) \]

\[ \forall j \in \mathcal{J}, \forall i \in \mathcal{I}_j, \forall k \in \mathcal{T}_j, \quad 0 \leq \sum_{\alpha \in \mathcal{R}^-} x_{i,j,k}^{(\alpha)} \leq \sigma_{j,k} . \quad (17) \]

Note that the variables \( \tau \) do not form part of the LP description, and therefore do not contribute to its complexity. However these variables will provide a convenient notational shorthand for proving results in this paper.

We will prove that optimizing the cost function (2) over this new polytope is equivalent to optimizing over \( \mathcal{Q} \). First, we state the following proposition, which will be necessary to prove this result.

**Proposition 3.1:** Let \( M \in \mathbb{N} \) and \( k \in \mathbb{N}^{\mathcal{I}^-} \). Also let \( \Gamma \subseteq \mathcal{I} \). Assume that for each \( \alpha \in \mathcal{R}^- \), we have a set of nonnegative integers \( \chi^{(\alpha)} = \{ x_{i}^{(\alpha)} : i \in \Gamma \} \) and that together these satisfy the constraints

\[ \sum_{i \in \Gamma} x_{i}^{(\alpha)} = k_\alpha M \quad (18) \]

for all \( \alpha \in \mathcal{R}^- \), and

\[ \sum_{\alpha \in \mathcal{R}^-} x_{i}^{(\alpha)} \leq M \quad (19) \]

for all \( i \in \Gamma \).

Then, there exist nonnegative integers \( \{ w_{\alpha} : \alpha \in \mathcal{C}_{\Gamma}^{(k)} \} \) such that

\[ \sum_{\alpha \in \mathcal{C}_{\Gamma}^{(k)}} w_{\alpha} = M . \quad (20) \]

1) \( \sum_{a \in \mathcal{C}_{\Gamma}^{(k)}} w_{a} = M . \quad (20) \)

2) For all \( \alpha \in \mathcal{R}^- \), \( i \in \Gamma \),

\[ x_{i}^{(\alpha)} = \sum_{a \in \mathcal{C}_{\Gamma}^{(k)}, a \alpha = \alpha} w_{a} . \quad (21) \]

A sketch of the proof of this proposition will follow at the end of this section. We now prove the main result.

**Theorem 3.2:** The set \( \mathcal{U} = \{ f : \exists \sigma, z \text{ s.t. } (f, \sigma, z) \in \mathcal{U} \} \) is equal to the set \( \mathcal{Q} = \{ f : \exists w \text{ s.t. } (f, w) \in \mathcal{Q} \} \). Therefore, optimizing the linear cost function (2) over \( \mathcal{U} \) is equivalent to optimizing over \( \mathcal{Q} \).

**Proof:**

1) Suppose, \( (f, w) \in \mathcal{Q} \). For all \( j \in \mathcal{J}, k \in \mathcal{T}_j \), we define

\[ \sigma_{j,k} = \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k} w_{j,b} , \]

and for all \( j \in \mathcal{J}, i \in \mathcal{I}_j, k \in \mathcal{T}_j, \alpha \in \mathcal{R}^- \), we define

\[ z_{i,j,k}^{(\alpha)} = \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k, b_i = \alpha} w_{j,b} , \]

It is straightforward to check that constraints (10) and (11) are satisfied by these definitions. For every \( j \in \mathcal{J}, i \in \mathcal{I}_j, \alpha \in \mathcal{R}^- \), we have by (5)

\[ f_i^{(\alpha)} = \sum_{b \in \mathcal{C}_j, b_i = \alpha} w_{j,b} \]

\[ = \sum_{k \in \mathcal{T}_j} \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k, b_i = \alpha} w_{j,b} = \sum_{k \in \mathcal{T}_j} z_{i,j,k}^{(\alpha)} , \]

and thus constraint (7) is satisfied. Next, for every \( j \in \mathcal{J} \), we have by (4)

\[ 1 = \sum_{b \in \mathcal{C}_j} w_{j,b} = \sum_{k \in \mathcal{T}_j} \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k} w_{j,b} \]

\[ = \sum_{k \in \mathcal{T}_j} \sigma_{j,k} , \]

and thus constraint (8) is satisfied. Finally, for every \( j \in \mathcal{J}, k \in \mathcal{T}_j, \alpha \in \mathcal{R}^- \),

\[ \sum_{i \in \mathcal{I}_j, \beta \in \mathcal{R}^-, \beta \mathcal{H}_{i,j} = \alpha} z_{i,j,k}^{(\beta)} \]

\[ = \sum_{i \in \mathcal{I}_j, \beta \in \mathcal{R}^-, \beta \mathcal{H}_{i,j} = \alpha} \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k, b_i = \beta} w_{j,b} \]

\[ = \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k} \sum_{i \in \mathcal{I}_j, b_i \mathcal{H}_{i,j} = \alpha} w_{j,b} \]

\[ = \sum_{b \in \mathcal{C}_j, \kappa_j(b) = k} k_\alpha \cdot w_{j,b} = k_\alpha \cdot \sigma_{j,k} . \]

Thus, constraint (9) is also satisfied. This completes the proof of the first part of the theorem.

2) Now assume \( (f, \sigma, z) \) is a vertex of the polytope \( \mathcal{U} \), and so all variables \( \tau \) are rational, as are the variables \( \sigma \).

Next, fix some \( j \in \mathcal{J}, k \in \mathcal{T}_j \), and consider the sets

\[ \chi_0^{(\alpha)} = \left\{ \frac{x_{i,j,k}^{(\alpha)}}{\sigma_{j,k}} : i \in \mathcal{I}_j \right\} . \]

for all \( \alpha \in \mathcal{R}^- \). By constraint (17), for each \( \alpha \in \mathcal{R}^- \), all the values in the set \( \chi_0^{(\alpha)} \) are rational numbers between 0 and 1. Let \( \mu \) be the lowest common denominator of all the numbers in all the sets \( \chi_0^{(\alpha)} \), \( \alpha \in \mathcal{R}^- \). Let

\[ \chi^{(\alpha)} = \left\{ \frac{\mu \cdot x_{i,j,k}^{(\alpha)}}{\sigma_{j,k}} : i \in \mathcal{I}_j \right\} , \]

for each \( \alpha \in \mathcal{R}^- \). The sets \( \chi^{(\alpha)} \) consist of integers between 0 and \( \mu \). By constraint (16), we must have that for every \( \alpha \in \mathcal{R}^- \), the sum of the elements in \( \chi^{(\alpha)} \) is equal to \( k_\alpha \mu \). By constraint (17), we have

\[ \sum_{\alpha \in \mathcal{R}^-} \mu \cdot \frac{x_{i,j,k}^{(\alpha)}}{\sigma_{j,k}} \leq \mu \]

for all \( i \in \mathcal{I}_j \).

We now apply the result of Proposition 3.1 with \( \Gamma = \mathcal{I}_j \), \( M = \mu \) and with the sets \( \chi^{(\alpha)} \) defined as above (here
\[ N = d_j \). Set the variables \( \{ w_\alpha : \alpha \in C_\Gamma^{(k)} \} \) according to Proposition \[3.1\].

Next, for \( k \in T_j \), we show how to define the variables \( \{ w'_b : \ b \in C_j, \kappa_j(b) = k \} \). Initially, we set \( w'_b = 0 \) for all \( b \in C_j, \kappa_j(b) = k \). Observe that the values \( \mu \cdot \frac{z_{i,j,k}}{\sigma_{j,k}} \) are non-negative integers for every \( i \in I, j \in J, k \in T_j, \beta \in \mathbb{R}^- \).

For every \( \alpha \in C_\Gamma^{(k)} \), we define \( w_\alpha \) words \( b^{(1)}, b^{(2)}, \ldots, b^{(w_\alpha)} \in C_j \). Assume some ordering on the elements \( \beta \in \mathbb{R}^- \) satisfying \( \beta \in \mathbb{T}_{i,j,i} = a_i \), namely \( \beta_1, \beta_2, \ldots, \beta_{w_\alpha} \) for some positive integer \( t_\alpha \). For \( i \in T_j, b^{(i)}_j (\ell = 1, 2, \ldots, w_\alpha) \) is defined as follows: \( b^{(i)}_j \) is equal to \( \beta_1 \) for the first \( i \cdot z_{i,j,k} / \sigma_{j,k} \) \( b^{(1)}, b^{(2)}, \ldots, b^{(w_\alpha)} \); \( b^{(i)}_j \) is equal to \( \beta_2 \) for the next \( i \cdot z_{i,j,k} / \sigma_{j,k} \), and so on. For every \( b \in C_j \), we define

\[
\sigma_{j,k} = \left\{ i \in \{1, 2, \ldots, w_\alpha \} : b^{(i)}_j = b \right\}.
\]

Finally, for every \( b \in C_j, \kappa_j(b) = k \), we define

\[
w_{j,b} = \frac{\sigma_{j,k}}{\mu} \cdot w'_b.
\]

Using Proposition \[3.1\]

\[
\sum_{\alpha \in C_\Gamma^{(k)}, a_\alpha = \alpha} w_\alpha = \mu \cdot \frac{T_{i,j,k}}{\sigma_{j,k}} = \sum_{\beta : \beta \in \mathbb{T}_{i,j,i} = \alpha} \mu \cdot \frac{z_{i,j,k}}{\sigma_{j,k}},
\]

and so all \( b^{(1)}, b^{(2)}, \ldots, b^{(w_\alpha)} \) (for all \( \alpha \in C_\Gamma^{(k)} \)) are well-defined. It is also straightforward to see that \( b^{(\ell)}_j \in C_j \) for \( \ell = 1, 2, \ldots, w_\alpha \). Next, we check that the newly-defined \( w_{j,b} \) satisfy \( \text{(3)-(5)} \) for every \( j \in J, b \in C_j \).

It is easy to see that \( w_{j,b} \geq 0 \); therefore \( \text{(4)} \) holds. By Proposition \[3.1\] we obtain

\[
\sigma_{j,k} = \sum_{b \in C_j, \kappa_j(b) = k} w_{j,b},
\]

for all \( j \in J, k \in T_j \), and

\[
\sum_{\beta : \beta \in \mathbb{T}_{i,j,i} = \alpha} \mu \cdot \frac{z_{i,j,k}}{\sigma_{j,k}} = \sum_{b \in C_j, \kappa_j(b) = k, b \in \mathbb{T}_{j,i} = \alpha} w_{j,b},
\]

for all \( j \in J, i \in T_j, k \in T_j, \alpha \in \mathbb{R}^- \). Let \( \mathbb{T}_{j,i} = \alpha \).

By the definition of \( w_{j,b} \) it follows that

\[
\sum_{b \in C_j, \kappa_j(b) = k, b \in \mathbb{T}_{j,i} = \alpha} w_{j,b} = \sum_{\beta : \beta \in \mathbb{T}_{i,j,k} = \alpha} \mu \cdot \frac{z_{i,j,k}}{\sigma_{j,k}},
\]

where the first equality is due to the definition of the words \( b^{(\ell)}_j, \ell = 1, 2, \ldots, w_\alpha \).

By constraint \( \text{(4)} \) we have, for all \( j \in J \),

\[
1 = \sum_{k \in T_j} \sigma_{j,k} = \sum_{k \in T_j} \sum_{b \in C_j, \kappa_j(b) = k} w_{j,b} = \sum_{b \in C_j} \sum_{k \in T_j} w_{j,b},
\]

thus satisfying \( \text{(4)} \).

Finally, by constraint \( \text{(1)} \) we obtain, for all \( j \in J, i \in T_j, \beta \in \mathbb{R}^- \),

\[
f_i^{(\beta)} = \sum_{k \in T_j} \sum_{b \in C_j, \kappa_j(b) = k, b \in \mathbb{T}_{j,i} = \beta} w_{j,b} = \sum_{b \in C_j, b \in \mathbb{T}_{j,i} = \beta} w_{j,b},
\]

thus satisfying \( \text{(5)} \).

**Sketch of the Proof of Proposition \[3.1\]**

In this proof, we use a network flow approach (see \[6\] for background material).

The proof will be by induction on \( M \). We set \( w_\alpha = 0 \) for all \( \alpha \in C_\Gamma^{(k)} \). We show that there exists a vector \( \alpha = (a_\alpha)_{\alpha \in \mathbb{R}^-} \) such that

(i) For every \( i \in I \) and \( \alpha \in \mathbb{R}^- \),

\[
a_i = \alpha \implies x_i^{(\alpha)} > 0.
\]

(ii) If for some \( i \in I, \sum_{\alpha \in \mathbb{R}^-} x_i^{(\alpha)} = M \), then \( a_i = \alpha \) for some \( \alpha \in \mathbb{R}^- \).

Then, we ‘update’ the values of \( x_i^{(\alpha)} \)’s and \( M \) as follows. For every \( i \in I \) and \( \alpha \in \mathbb{R}^- \) with \( a_i = \alpha \) we set \( x_i^{(\alpha)} \leftarrow x_i^{(\alpha)} - 1 \). In addition, we set \( M \leftarrow M - 1 \). We also set \( w_\alpha \leftarrow w_\alpha + 1 \).

It is easy to see that the ‘updated’ values of \( x_i^{(\alpha)} \)’s and \( M \) satisfy

\[
\sum_{\alpha \in \mathbb{R}^-} x_i^{(\alpha)} = k_\alpha M
\]

for all \( \alpha \in \mathbb{R}^- \), and \( \sum_{\alpha \in \mathbb{R}^-} x_i^{(\alpha)} \leq M \) for all \( i \in I \). Therefore, the inductive step can be applied with respect to these new values. The induction ends when the value of \( M \) is equal to zero.

It is straightforward to see that when the induction terminates, \( \text{(20)} \) and \( \text{(21)} \) hold with respect to the original values of the \( x_i^{(\alpha)} \)’s and \( M \).

**Existence of \( \alpha \) that satisfies (i):** We construct a flow network \( G = (V, E) \) as follows: \( V = \{ s, t \} \cup U_1 \cup U_2 \), where \( U_1 = \mathbb{R}^- \) and \( U_2 = \Gamma \). Also set

\[
E = \{(s, \alpha)\alpha \in \mathbb{R}^- \cup \{(i, t)\}_{i \in \Gamma} \cup \{(\alpha, i)\}_{\alpha \in \mathbb{R}^-}, i \in \Gamma \}
\]

We define an integral capacity function \( c : E \rightarrow \mathbb{N} \cup \{+\infty\} \) as follows:

\[
c(e) = \begin{cases} k_\alpha & \text{if } e = (s, \alpha), \alpha \in \mathbb{R}^- \\ 1 & \text{if } e = (i, t), i \in \Gamma \\ +\infty & \text{if } e = (\alpha, i), \alpha \in \mathbb{R}^-, i \in \Gamma \end{cases}
\]

(22)

Next, apply the Ford-Fulkerson algorithm on the network \( (G(E, V), c) \) to produce a maximal flow \( f_{\text{max}} \). Since all the values of \( c(e) \) are integral for all \( e \in E \), the value of \( f_{\text{max}} \) must be integral for every \( e \in E \) (see \[6\]).

It can be shown that the minimum cut in this graph has capacity \( c_{\text{min}} = \sum_{\alpha \in \mathbb{R}^-} k_\alpha \).
The flow $f_{\text{max}}$ in $G$ has a value of $\sum_{\alpha \in \mathcal{P}} k_{\alpha}$. Observe that $f_{\text{max}}((\alpha, i)) \in \{0, 1\}$ for all $\alpha \in \mathcal{P}^{-}$ and $i \in \Gamma$. Then, for all $i \in \Gamma$, we define
\[
a_i = \begin{cases} 
\alpha & \text{if } f_{\text{max}}((\alpha, i)) = 1 \text{ for some } \alpha \in U_1 \\
0 & \text{otherwise}
\end{cases}.
\]
For this selection of $a = (a_1, a_2, \cdots, a_N)$, we have $a \in C_{\Gamma}^{(k)}$ and $a_i = \alpha$ only if $a_i^{(\alpha)} > 0$.

Existence of $a$ that satisfies (i) and (ii) simultaneously: We start with the following definition.

**Definition 3.1:** The vertex $i \in U_2$ is called a critical vertex, if $\sum_{\alpha \in \mathcal{P}^{+}} x_{i}^{(\alpha)} = M$.

In order to have (19) satisfied after the next inductive step, we have to decrease the value of $\sum_{\alpha \in \mathcal{P}^{+}} x_{i}^{(\alpha)}$ by (exactly) 1 for every critical vertex. This is equivalent to having $f_{\text{max}}((i, t)) = 1$.

We aim to show that there exists a flow $f^*$ of the same value, which has $f^*((i, t)) = 1$ for every critical vertex $i$. Suppose that there is no such flow. Then, consider the maximum flow $f'$, which has $f'((i, t)) = 1$ for the maximum possible number of the critical vertices $i \in U_2$. We assume that there is a critical vertex $i_0 \in U_2$, which has $f'((i_0, t)) = 0$. It is possible to show that the flow $f'$ can be modified towards the flow $f^*$ of the same value, such that for $f''$ the number of critical vertices $i \in U_2$ having $f''((i, t)) = 1$ is strictly larger than for $f'$.

It follows that there exists an integral flow $f^*$ in $(G(V, E), c)$ of value $\sum_{\alpha \in \mathcal{P}^{+}} k_{\alpha}$, such that for every critical vertex $i \in U_2$, $f^*((i, t)) = 1$. We define
\[
a_i = \begin{cases} 
\alpha & \text{if } f^*((\alpha, i)) = 1 \text{ for some } \alpha \in U_1 \\
0 & \text{otherwise}
\end{cases}.
\]
and $a = (a_i)_{i \in \Gamma}$. For this selection of $a$, we have $a \in C_{\Gamma}^{(k)}$ and the properties (i) and (ii) are satisfied.

IV. CASCaded POLYTOPE REPRESENTATION

In this section we show that the “cascaded polytope” representation described in [4] and [5] can be extended to non-binary codes in a straightforward manner. Below, we elaborate on the details.

For $j \in J$, consider the $j$-th row $\mathcal{H}_j$ of the parity-check matrix $\mathcal{H}$ over $\mathcal{R}$, and recall that
\[
\mathcal{C}_j = \{ (b_i)_{i \in \mathcal{I}_j} : \sum_{i \in \mathcal{I}_j} b_i \cdot \mathcal{H}_{j,i} = 0 \}.
\]
Assume that $\mathcal{I}_j = \{ i_1, i_2, \cdots, i_{d_j} \}$ and denote $\mathcal{L}_j = \{ 1, 2, \cdots, d_j - 3 \}$. We introduce new variables $\chi_j^\delta = (\chi_j^\delta)_{\delta \in \mathcal{L}_j}$ and denote $\chi = (\chi_j^\delta)_{j \in J}$.

We define a new linear code $C_{\Gamma}^{(\chi)}$ of length $2d_j - 3$ by
\[
b_{i_{d_j}} \mathcal{H}_{j,i_{d_j}} + b_{i_{d_j-1}} \mathcal{H}_{j,i_{d_j-1}} + \chi_1 = 0 .
\]
We also define a linear code $C_{\Gamma}^{(\chi)}$ of length $n + \sum_{j \in J} (d_j - 3)$ defined by $\sum_{j \in J} (d_j - 2) \times (n + \sum_{j \in J} (d_j - 3))$ parity-check matrix $\mathcal{F}$ associated with all the sets of parity-check equations (23)-(25).}

**Theorem 4.1:** The vector $(b_i)_{i \in \mathcal{I}_j} \in \mathcal{R}^{d_j}$ is a codeword of $\mathcal{C}_j$ if and only if there exists some vector $\chi_j^\delta \in \mathcal{R}^{d_j-3}$ such that $(b_i)_{i \in \mathcal{I}_j}, | \chi_j^\delta | \in C_{\Gamma}^{(\chi)}$.

We denote by $S$ the polytope corresponding to the LP relaxation problem (3)-(5) for the code $C_{\Gamma}^{(\chi)}$ with the parity-check matrix $\mathcal{F}$. Let $(b, \chi)$ be a word in $C_{\Gamma}^{(\chi)}$, where $b \in \mathcal{B}$. It is natural to represent points in $S$ as $((f, \chi), z)$, where $f = (f_i^{(\alpha)})_{i \in \mathcal{I}}, \alpha \in \mathcal{P}^{-}$ and $h = (h_i^{(\alpha)})_{i \in \mathcal{L}, \alpha \in \mathcal{P}^{-}}$ are vectors of indicators corresponding to the entries $b_i (i \in \mathcal{I})$ in $b$ and $\chi_{\delta} (j \in J, \delta \in \mathcal{L})$ in $\chi$, respectively.

**Theorem 4.2:** The set $S = \{ (f, \chi) : \exists w \ s.t. (f, w) \in Q \}$ is equal to the set $Q = \{ (f, \chi) : \exists w \ s.t. (f, w) \in Q \}$, and therefore, optimizing the linear cost function (2) over $S$ is equivalent to optimizing it over $Q$.

It follows from Theorem 4.2 that the polytope $S$ equivalently describes the code $C$. This description has at most $n + m \cdot (d - 3)$ variables and $m \cdot (d - 2)$ parity-check equations. However, the number of variables participating in every parity-check equation is at most 3. Therefore, the total number of variables and of equations in the respective LP problem will be bounded from above by
\[
(n + m(d-3)) (q-1) + m(d-2) \cdot q^2
\]
and
\[
m(d-2)(q^2 + 3q - 2) .
\]

The polytope representation in this section, when used with the LP problem in [3], leads to a polynomial-time decoder for a wide variety of classical non-binary codes. Its performance under LP decoding is yet to be studied.

**Acknowledgements**

The authors wish to thank O. Milenkovic for interesting discussions. This work was supported by the Claude Shannon Institute for Discrete Mathematics, Coding and Cryptography (Science Foundation Ireland Grant 06/MI/006).

**References**

[1] J. Feldman, Decoding Error-Correcting Codes via Linear Programming, Ph.D. Thesis, Massachusetts Institute of Technology, Sep. 2003.

[2] J. Feldman, M.J. Wainwright, D.R. Karger, Using linear programming to decode binary linear codes, IEEE Trans. Inform. Theory, vol. 51, no. 3, pp. 954-972, Mar. 2005.

[3] M.F. Flanagan, V. Skachek, E. Byrne, M. Grefeffarth, Linear-programming Decoding of Non-binary Linear Codes, Proc. 7th Int. ITG Conf. on Source and Channel Coding (SCC), Jan. 2008, Ulm, Germany. Available also at http://arxiv.org/abs/0707.4360

[4] M. Cherkov, M. Stepánov, Pseudo-codeword Landscape, Proc. IEEE Int. Symp. on Inform. Theory (ISIT), June 2007, pp.1546-1550, Nice, France.

[5] K. Yang, X. Wang, J. Feldman, Cascaded Formulation of the Fundamental Polytope of General Linear Block Codes, Proc. IEEE Int. Symp. on Inform. Theory (ISIT), June 2007, p.1361-1365, Nice, France.

[6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second edition, MIT Press and McGraw-Hill, 2001.