Minimum Pseudoweight Analysis of 3-Dimensional Turbo Codes

Minimum Pseudoweight Analysis of 3-Dimensional Turbo Codes

Eirik Rosnes, *Senior Member, IEEE*, Michael Helmling, and Alexandre Graell i Amat, *Senior Member, IEEE*

***Abstract*—In this work, we consider pseudocodewords of (relaxed) linear programming (LP) decoding of 3-dimensional turbo codes (3D-TCs). We present a relaxed LP decoder for 3D-TCs, adapting the relaxed LP decoder for conventional turbo codes proposed by Feldman in his thesis. We show that the 3D-TC polytope is *proper* and C*-symmetric*, and make a connection to finite graph covers of the 3D-TC factor graph. This connection is used to show that the support set of any pseudocodeword is a stopping set of iterative decoding of 3D-TCs using maximum *a posteriori* constituent decoders on the binary erasure channel. Furthermore, we compute ensemble-average pseudoweight enumerators of 3D-TCs and perform a finite-length minimum pseudoweight analysis for small cover degrees. Also, an explicit description of the fundamental cone of the 3D-TC polytope is given. Finally, we present an extensive numerical study of small-to-medium block length 3D-TCs, which shows that 1) typically (i.e., in most cases) when the minimum distance $d_{\min}$ and/or the stopping distance $h_{\min}$ is high, the minimum pseudoweight (on the additive white Gaussian noise channel) is strictly smaller than both the $d_{\min}$ and the $h_{\min}$, and 2) the minimum pseudoweight grows with the block length, at least for small-to-medium block lengths.**

***Index Terms*—3-dimensional (3D) turbo codes, graph cover, hybrid concatenated codes, linear programming (LP) decoding, QPP interleavers, pseudocodewords, pseudoweight, turbo codes, uniform interleaver.**

I. Introduction

TURBO codes (TCs) have gained considerable attention since their introduction by Berrou *et al.* in 1993 [1] due to their near-capacity performance and low decoding complexity. The conventional TC is a parallel concatenation of two identical recursive systematic convolutional encoders separated by a pseudorandom interleaver. To improve the performance of TCs in the error floor region, hybrid concatenated codes (HCCs) can be used. In [2], a powerful HCC nicknamed 3-dimensional turbo code (3D-TC) was introduced. The coding scheme consists of a conventional turbo encoder and a *patch*, where a fraction λ of the parity bits from the turbo encoder are post-encoded by a third rate-1 encoder. The value of λ can be used to trade off performance in the waterfall region with performance in the error floor region. As shown in [2], this coding scheme is able to provide very low error rates for a wide range of block lengths and code rates at the expense of a small increase in decoding complexity with respect to conventional TCs. In a recent work [3], an in-depth performance analysis of 3D-TCs was conducted. Stopping sets for 3D-TCs were treated in [4]. Finally, we remark that tuned TCs [5] are another family of HCC ensembles where performance in the waterfall and error floor regions can be traded off using a tuning parameter.

The use of linear programming (LP) to decode turbo-like codes was introduced by Feldman *et al.* [6], [7]. They proposed an LP formulation that resembles a shortest path problem, based on the trellis graph representation of the constituent convolutional codes. The natural polytope for LP decoding would be the convex hull of all codewords, in which case LP decoding is equivalent to maximum-likelihood (ML) decoding. However, an efficient description of that polytope is not known in general, i.e., its description length most likely grows exponentially with the block length. The formulation proposed by Feldman *et al.* [6], [7], which grows only linearly with the block length, is a relaxation in the sense that it describes a superset of the ML decoding polytope, introducing additional, fractional vertices. The vertices of the relaxed polytope (both integral and fractional) are what the authors called *pseudocodewords* in [8].

The same authors also introduced a different LP formulation to decode low-density parity-check (LDPC) codes [7], [8] that has been extensively studied since then by various researchers. For that LP decoder, Vontobel and Koetter [9] showed that the set of the polytope's points with rational coordinates (which in particular includes all of its vertices) is equal to the set of all pseudocodewords derived from all finite graph covers of the Tanner graph. In [10], a similar result was established (but with no proof included) for the case of conventional TCs. Here, we prove that statement for the case of 3D-TCs.

In this work, we study (relaxed) LP decoding of 3D-TCs, explore the connection to finite graph covers of the 3D-TC factor graph [11], and adapt the concept of a pseudocodeword. Furthermore, we compute ensemble-average pseudoweight enumerators and perform a finite-length minimum additive white Gaussian noise (AWGN) pseudoweight analysis which shows that the minimum AWGN pseudoweight grows with the block length, at least for small-to-medium block lengths.

The work of E. Rosnes and M. Helmling was partially funded by the DFG (grant RU-1524/2-1) and by DAAD / RCN (grant 54565400 within the German-Norwegian Collaborative Research Support Scheme). The work of A. Graell i Amat was supported by the Swedish Research Council under grant #2011-596. The material in this paper was presented in part at the 2011 IEEE International Symposium on Information Theory, St. Petersburg, Russia, Jul./Aug. 2011.

E. Rosnes was with Ceragon Networks AS, Kokstadveien 23, N-5257 Kokstad, Norway. He is now with the Selmer Center, Department of Informatics, University of Bergen, N-5020 Bergen, Norway, and the Simula Research Lab. E-mail: eirik@ii.uib.no.

M. Helmling was with the Department of Mathematics, University of Kaiserslautern, 67663 Kaiserslautern, Germany. He is now with the Mathematical Institute, University of Koblenz-Landau, 56070 Koblenz, Germany. Email: helmling@uni-koblenz.de.

A. Graell i Amat is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden. Email: alexandre.graell@chalmers.se.

Furthermore, we show by several examples and by computer search that typically (i.e., in most cases) when the minimum distance $d_{\min}$ and/or the stopping distance $h_{\min}$ is high, the minimum AWGN pseudoweight, denoted by $w_{\min}^{\mathrm{AWGN}}$, is strictly smaller than both the $d_{\min}$ and the $h_{\min}$ for these codes. In [12], Chertkov and Stepanov presented an updated, more efficient (compared to the algorithm from [13]) minimum pseudoweight search algorithm based entirely on the concept of the *fundamental cone* [9] of the LDPC code LP decoder. We will show that such a fundamental cone can be described also for the 3D-TC LP decoder.

Some other work related to the content of this paper is the work on pseudocodeword analysis of binary and nonbinary (protograph-based) LDPC and generalized LDPC codes. See, for instance, [14], [15], [16], and references therein. For such codes, the component codes are the trivial repetition code and single parity-check codes, or in the case of generalized LDPC codes, more advanced classical linear block codes. However, not much work has been done on pseudocodeword analysis for turbo-like codes with trellis-based constituent codes. In contrast to these previous works, the trellis-based approach in this paper is different and provides a pseudocodeword analysis of 3D-TCs that can be adapted also to other trellis-based turbo-like codes or concatenated codes based on two or more convolutional codes. We remark that some results on enumerating pseudocodewords for convolutional codes have already been provided by Boston and Conti in [17], [18]. Finally, it is worth mentioning [19] which presents, among several results, a combinatorial characterization of the *Bethe entropy function* in terms of finite graph covers of the factor graph under consideration. In particular, a characterization in terms of the average number of preimages of a *pseudomarginal* vector of rational entries. In contrast to the general framework in [19], this paper discusses techniques to numerically deal with large-degree function nodes representing the indicator function of (long) convolutional codes.

The remainder of the paper is organized as follows. In Section II, we describe the system model and introduce some notation. In Section III, we describe (relaxed) LP decoding of 3D-TCs. The connection to finite graph covers of the 3D-TC factor graph is explored in Section IV. Ensemble-average pseudoweight enumerators of 3D-TCs are computed in Section V. These enumerators are subsequently used to perform a probabilistic finite-length minimum AWGN pseudoweight analysis of 3D-TCs. In Section VI, an efficient heuristic for searching for low AWGN pseudoweight pseudocodewords is discussed. Finally, in Section VII, an extensive numerical study is presented, and some conclusions are drawn in Section VIII.

II. Coding Scheme

A block diagram of the 3D-TC is depicted in Fig. 1. The information data sequence $\mathbf{u}$ of length K bits is encoded by a binary conventional turbo encoder. By a conventional turbo encoder we mean the parallel concatenation of two identical rate-1 recursive convolutional encoders, denoted by C_{a} and C_{b}, respectively. Here, C_{a} and C_{b} are 8-state recursive convolutional encoders with generator polynomial $g(D) = (1 + D + D^3)/(1 + D^2 + D^3)$, i.e., the 8-state constituent encoder specified in the 3GPP LTE standard [20]. The code sequences of C_{a} and C_{b} are denoted by $\mathbf{x}_{\mathrm{a}}$ and $\mathbf{x}_{\mathrm{b}}$, respectively. We also denote by $\mathbf{x}^{\mathrm{TC}}$ the codeword obtained by alternating bits from $\mathbf{x}_{\mathrm{a}}$ and $\mathbf{x}_{\mathrm{b}}$. A fraction λ ($0 \leq \lambda \leq 1$), called the *permeability rate*, of the parity bits from $\mathbf{x}^{\mathrm{TC}}$ are permuted by the interleaver Π_{c} (of length $N_{\mathrm{c}} = 2\lambda K$), and encoded by an encoder of unity rate C_{c} with generator polynomial $g(D) = 1/(1 + D^2)$, called the *patch* or the *post-encoder* [2]. This can be properly represented by a puncturing pattern $\mathbf{p}$ applied to $\mathbf{x}^{\mathrm{TC}}$ (see Fig. 1) of period N_{p} containing λN_{p} ones (where a one means that the bit is not punctured). Note that the encoder of the patch is like two accumulators, one operating on the even bits and one operating on the odd bits. The fraction $1 - \lambda$ of parity bits which are not encoded by C_{c} is sent directly to the channel. Equivalently, this can be represented by a puncturing pattern $\bar{\mathbf{p}}$, the complement of $\mathbf{p}$. We denote by $\mathbf{x}_{\mathrm{c}}$ the code sequence produced by C_{c}. Also, we denote by $\mathbf{x}_{\mathrm{a}}^{\mathrm{ch}}$ and $\mathbf{x}_{\mathrm{b}}^{\mathrm{ch}}$ the *sub-codewords* of $\mathbf{x}_{\mathrm{a}}$ and $\mathbf{x}_{\mathrm{b}}$, respectively, sent directly to the channel, and by $\mathbf{x}^{\mathrm{ch}}$ the codeword obtained by multiplexing (in some order) the bits from $\mathbf{x}_{\mathrm{a}}^{\mathrm{ch}}$ and $\mathbf{x}_{\mathrm{b}}^{\mathrm{ch}}$. Likewise, we denote by $\mathbf{x}_{\mathrm{a}}^{\mathrm{p}}$ and $\mathbf{x}_{\mathrm{b}}^{\mathrm{p}}$ the *sub-codewords* of $\mathbf{x}_{\mathrm{a}}$ and $\mathbf{x}_{\mathrm{b}}$, respectively, encoded by C_{c}, and by $\mathbf{x}^{\mathrm{p}}$ the codeword obtained by multiplexing (in some order) the bits from $\mathbf{x}_{\mathrm{a}}^{\mathrm{p}}$ and $\mathbf{x}_{\mathrm{b}}^{\mathrm{p}}$. Finally, the information sequence and the code sequences $\mathbf{x}^{\mathrm{ch}}$ and $\mathbf{x}_{\mathrm{c}}$ are multiplexed to form the code sequence $\mathbf{x}$, of length N bits, transmitted to the channel. Note that the overall nominal code rate of the 3D-TC is $R = K/N = 1/3$, the same as for the conventional TC without the patch. Higher code rates can be obtained either by puncturing $\mathbf{x}^{\mathrm{ch}}$ or by puncturing the output of the patch, $\mathbf{x}_{\mathrm{c}}$.

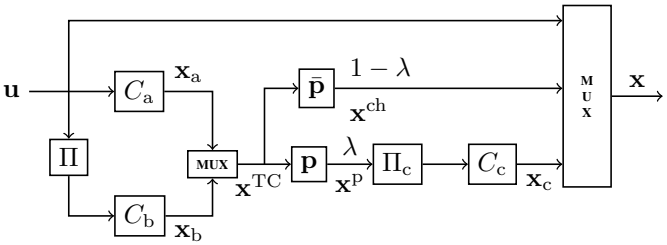

Fig. 1. 3D turbo encoder. A fraction λ of the parity bits from both constituent encoders C_{a} and C_{b} are grouped by a parallel/serial multiplexer, permuted by the interleaver Π_{c}, and encoded by the rate-1 post-encoder C_{c}.

In [2], regular puncturing patterns of period $2/\lambda$ were considered for $\mathbf{p}$. For instance, if $\lambda = 1/4$, every fourth bit from each of the encoders of the outer TC are encoded by encoder C_{c}. The remaining bits are sent directly to the channel, and it follows that $\mathbf{p} = [11000000]$ and $\bar{\mathbf{p}} = [00111111]$. Note that with this particular puncturing pattern and even with the generator polynomial $g(D) = 1/(1 + D^2)$ for C_{c} (which is like two separate accumulators operating on even and odd bits, respectively), the bit streams $\mathbf{x}_{\mathrm{a}}$ and $\mathbf{x}_{\mathrm{b}}$ are in general intermingled because of the interleaver Π_{c}.

III. LP Decoding of 3D-TCs

In this section, we consider relaxed LP decoding of 3D-TCs, adapting the relaxation proposed in [7] for conventional

TCs to 3D-TCs.

Let $T_{\rm x} = T_{\rm x}(V_{\rm x}, E_{\rm x})$ denote the *information bit-oriented trellis* of $C_{\rm x}$, ${\rm x} = {\rm a}, {\rm b}, {\rm c}$, where the vertex set $V_{\rm x}$ partitions as $V_{\rm x} = \cup_{t=0}^{I_{\rm x}} V_{{\rm x},t}$, which also induces the partition $E_{\rm x} = \cup_{t=0}^{I_{\rm x}-1} E_{{\rm x},t}$ of the edge set $E_{\rm x}$, where $I_{\rm x}$ is the trellis length of $T_{\rm x}$. In the following, the encoders $C_{\rm x}$ (and their corresponding trellises $T_{\rm x}$) are assumed (with some abuse of notation) to be systematic, in the sense that the output bits are prefixed with the input bits. Thus, $C_{\rm x}$ is regarded as a rate-$1/2$ encoder, and the trellis $T_{\rm x}$ has an output label containing two bits, for ${\rm x} = {\rm a}, {\rm b}, {\rm c}$. Now, let $e \in E_{{\rm x},t}$ be an arbitrary edge from the tth trellis section. The ith bit in the output label of e is denoted by $c_i(e)$, $i = 0, \ldots, n_{{\rm x},t} - 1$, the starting state of e as $s^S(e)$, and the ending state of e as $s^E(e)$, where $n_{{\rm x},t}$ is the number of bits in the output label of an edge $e \in E_{{\rm x},t}$.

For ${\rm x} = {\rm a}, {\rm b}, {\rm c}$, we define the path polytope $\mathcal{Q}_{\rm x}$ to be the set of all $\mathbf{f}^{\rm x} \in [0,1]^{|E_{\rm x}|}$ satisfying

$$\sum_{e \in E_{{\rm x},0}} f_e^{\rm x} = 1 \tag{1}$$

$$\sum_{\substack{e \in E_{\rm x}: \\ s^S(e) = v}} f_e^{\rm x} = \sum_{\substack{e \in E_{\rm x}: \\ s^E(e) = v}} f_e^{\rm x} \quad \begin{array}{l} \text{for all } v \in V_{{\rm x},t} \\ \text{and } t = 1, \ldots, I_{\rm x} - 1 \end{array} \tag{2}$$

and let $\mathcal{Q} = \mathcal{Q}_{\rm a} \times \mathcal{Q}_{\rm b} \times \mathcal{Q}_{\rm c}$. Note that $\mathcal{Q}$ is the set of all feasible network flows through the three trellis graphs $T_{\rm x}$, ${\rm x} = {\rm a}, {\rm b}, {\rm c}$.

Next, we define the polytope $\mathcal{Q}_{\Pi,\Pi_{\rm c}}$ as the pairs $(\tilde{\mathbf{y}}, \mathbf{f})$, where $\tilde{\mathbf{y}} \in [0,1]^{N+2\lambda K}$ and $\mathbf{f} = (\mathbf{f}^{\rm a}, \mathbf{f}^{\rm b}, \mathbf{f}^{\rm c}) \in [0,1]^{|E_{\rm a} \cup E_{\rm b} \cup E_{\rm c}|}$, meeting the constraints

$$\begin{aligned} &(\mathbf{f}^{\rm a}, \mathbf{f}^{\rm b}, \mathbf{f}^{\rm c}) \in \mathcal{Q} \\ &\sum_{\substack{e \in E_{{\rm x},t}: \\ c_i(e) = 1}} f_e^{\rm x} = \tilde{y}_{\rho_{\rm x}(\phi_{\rm x}(t,i))} \quad \begin{array}{l} \text{for } t = 0, \ldots, I_{\rm x} - 1, \\ i = 0, \ldots, n_{{\rm x},t} - 1, \text{ and} \\ {\rm x} = {\rm a}, {\rm b}, {\rm c}. \end{array} \end{aligned} \tag{3}$$

Here, $\phi_{\rm x}(t,i) = \sum_{j=0}^{t-1} n_{{\rm x},j} + i$, and $\rho_{\rm x}(\cdot)$ denotes the mapping of codeword indices of the constituent encoder $C_{\rm x}$ to codeword indices of the overall codeword of the 3D-TC appended with the $2\lambda K$ parity bits from encoders $C_{\rm a}$ and $C_{\rm b}$ which are sent to the patch.

Finally, let

$$\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}} = \Big\{ \mathbf{y} \in [0,1]^N : \exists \hat{\mathbf{y}} \in [0,1]^{2\lambda K}, \mathbf{f} \in \mathcal{Q} \text{ with } ((\mathbf{y}, \hat{\mathbf{y}}), \mathbf{f}) \in \mathcal{Q}_{\Pi,\Pi_{\rm c}} \Big\}$$

be the projection of $\mathcal{Q}_{\Pi,\Pi_{\rm c}}$ onto the first N variables.

Relaxed LP decoding (on a binary-input memoryless channel) of 3D-TCs can be described by the linear program

$$\text{minimize } \sum_{l=0}^{N-1} \lambda_l y_l \text{ subject to } \mathbf{y} \in \dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}} \tag{4}$$

where

$$\lambda_l = \log\left(\frac{\Pr\{r_l | c_l = 0\}}{\Pr\{r_l | c_l = 1\}} \right), \; l = 0, \ldots, N-1,$$

c_l is the lth codeword bit, and r_l is the lth component of the received vector. If instead of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$ we use the convex hull of the codewords of the 3D-TC, then solving the linear program in (4) is equivalent to ML decoding.

The notion of a *proper* and C*-symmetric* polytope was introduced in [7, Ch. 4] where the author proved that the probability of error of LP decoding is independent of the transmitted codeword on a binary-input output-symmetric memoryless channel when the underlying code is linear and the polytope is proper and C-symmetric.

Proposition 1: Let C denote a given 3D-TC with interleavers Π and $\Pi_{\rm c}$. The polytope $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$ is proper, i.e., $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}} \cap \{0,1\}^N = C$ and C-symmetric, i.e., for any $\mathbf{y} \in \dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$ and $\mathbf{c} \in C$ it holds that $|\mathbf{y} - \mathbf{c}| \in \dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$.

Feldman proved a similar statement in the context of LDPC codes (Lemma 5.2 and Theorem 5.4 in [7]). However, note that his proof is based explicitly on the inequalities of the LP formulation for LDPC codes, and therefore does not generalize to the polytope $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$.

In Appendix A, we give a formal proof of both statements in a much more general setting.

IV. Finite Graph Covers

Let C denote a given 3D-TC with interleavers Π and $\Pi_{\rm c}$, and constituent codes $C_{\rm x}$, ${\rm x} = {\rm a}, {\rm b}, {\rm c}$. The factor graph [11] of $C_{\rm x}$, denoted by $\Gamma(C_{\rm x})$, is composed of *state*, *input*, *parity*, and *check* vertices. The state vertices $s_{{\rm x},0}, \ldots, s_{{\rm x},I_{\rm x}}$ in $\Gamma(C_{\rm x})$ represent state spaces of the length-$I_{\rm x}$ information bit-oriented trellis $T_{\rm x}$ of $C_{\rm x}$. The lth check vertex represents the lth trellis section, i.e., it is an indicator function for the set of allowed combinations of *left* state, input symbol, parity symbol, and *right* state. A factor graph $\Gamma(C)$ of C is constructed as follows.

1) Remove all the input vertices of $\Gamma(C_{\rm b})$ by connecting the lth input vertex of $\Gamma(C_{\rm a})$ to the $\Pi(l)$th check vertex of $\Gamma(C_{\rm b})$.
2) Remove all the input vertices of $\Gamma(C_{\rm c})$ by connecting the parity vertex (from either $\Gamma(C_{\rm a})$ or $\Gamma(C_{\rm b})$) corresponding to the lth bit in $\mathbf{x}^{\rm p}$ to the $\Pi_{\rm c}(l)$th check vertex of $\Gamma(C_{\rm c})$.

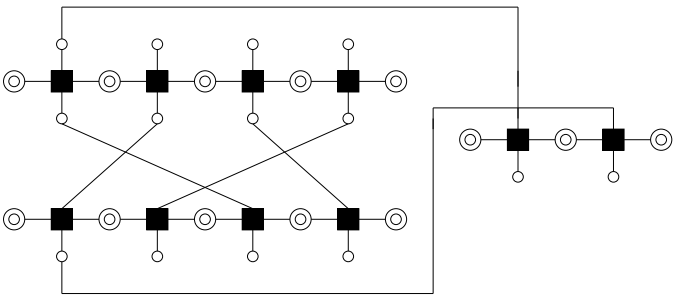

Fig. 2. Factor graph of a nominal rate-$1/3$ 3D-TC with $\lambda = 1/4$, using the regular puncturing pattern $\mathbf{p} = [11000000]$.

To construct a degree-m cover of $\Gamma(C)$, denoted by $\Gamma^{(m)}(C)$, we first make m identical copies of $\Gamma(C)$. Now, any permutation of the edges, denoted by $E = E(\Gamma^{(m)}(C))$, connecting the copies of the constituent factor graphs $\Gamma(C_{\rm x})$ such that the following conditions are satisfied, will give a valid cover of $\Gamma(C)$.

1) The m copies of the lth input vertex of $\Gamma(C_{\rm a})$ should be connected by a one-to-one mapping (or permutation) to the m copies of the $\Pi(l)$th check vertex of $\Gamma(C_{\rm b})$.

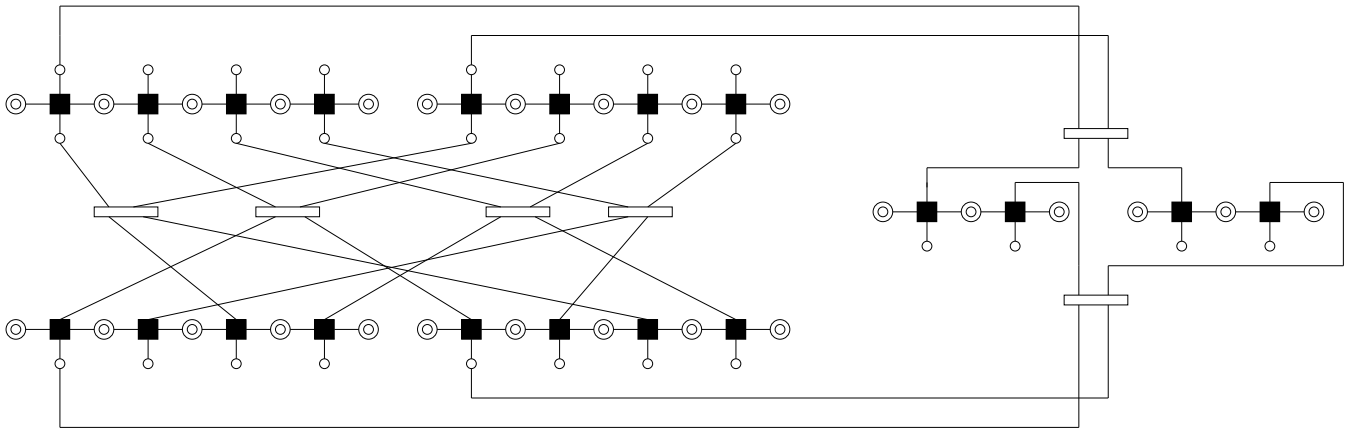

Fig. 3. Degree-2 cover of the factor graph in Fig. 2. The six small rectangular boxes are permutations of size 2 that can be chosen arbitrarily.

2) The m copies of the parity vertex (from either $\Gamma(C_{\rm a})$ or $\Gamma(C_{\rm b})$) corresponding to the lth bit in $\mathbf{x}^{\rm p}$ should be connected by a permutation to the m copies of the $\Pi_{\rm c}(l)$th check vertex of $\Gamma(C_{\rm c})$.

The corresponding code is denoted by $C^{(m)}$. Let $\mathbf{x}^{(m)} = (x_0^{(0)}, \ldots, x_{N-1}^{(0)}, \ldots, x_0^{(m-1)}, \ldots, x_{N-1}^{(m-1)})$ denote a codeword in $C^{(m)}$, define

$$\omega_l(\mathbf{x}^{(m)}) = \frac{|\{i:\, x_l^{(i)} = 1\}|}{m}$$

and let $\boldsymbol{\omega} = \boldsymbol{\omega}(\mathbf{x}^{(m)}) = (\omega_0(\mathbf{x}^{(m)}), \ldots, \omega_{N-1}(\mathbf{x}^{(m)}))$. Now, $\boldsymbol{\omega}$ as defined above is said to be a *graph-cover pseudocodeword* of degree m.

Example 1: Fig. 2 depicts the factor graph of a nominal rate-$1/3$ 3D-TC with $\lambda = 1/4$ and input length $K = 4$, using the regular puncturing pattern $\mathbf{p} = [11000000]$. The upper part to the left is the factor graph of $C_{\rm a}$, the lower part to the left is the factor graph of $C_{\rm b}$, and the right part is the factor graph of $C_{\rm c}$. The black squares in the graph are check vertices corresponding to trellis sections. The single circles are input and parity vertices, and the double circles are state vertices. Fig. 3 depicts a degree-2 cover of the factor graph from Fig. 2. The six small rectangular boxes are permutations of size 2 that can be chosen arbitrarily.

Proposition 2: The following statements are true:

1) The points in $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}} \cap \mathbb{Q}^N$ are in one-to-one correspondence with $\mathcal{P}_{\Pi,\Pi_{\rm c}}$, where $\mathbb{Q}$ is the set of rational numbers and $\mathcal{P}_{\Pi,\Pi_{\rm c}}$ is the set of all graph-cover pseudocodewords from all finite graph covers of the 3D-TC factor graph.
2) All vertices of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$ have rational entries.

A similar result was proved in [9] for LDPC codes. We give a formal proof for the case of 3D-TCs in Appendix B. It is inspired by the one in [9], but is a bit more involved because pseudocodewords are defined only indirectly by a linear image of the polytope $\mathcal{Q}$. Note that our proof does not depend on the detailed set-up of 3D-TCs, so it can be extended to all sorts of turbo-like coding schemes.

When decoding 3D-TCs by solving the linear program in (4), there is always a vertex of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\rm c}}$ at which the optimum value is attained. We can therefore assume that the LP decoder always returns a vertex and hence, by Proposition 2, a (graph-cover) pseudocodeword. Furthermore, the pseudoweight on the AWGN channel of a nonzero pseudocodeword $\boldsymbol{\omega}$ is defined as [9], [21]

$$w^{\rm AWGN}(\boldsymbol{\omega}) = \frac{\|\boldsymbol{\omega}\|_1^2}{\|\boldsymbol{\omega}\|_2^2} = \frac{\left(\sum_{l=0}^{N-1} \omega_l\right)^2}{\sum_{l=0}^{N-1} \omega_l^2} \tag{5}$$

where $\|\cdot\|_q$ is the ℓ_q-norm of a vector.

Proposition 3: Let C denote a given 3D-TC with interleavers Π and $\Pi_{\rm c}$. For any pseudocodeword $\boldsymbol{\omega}$, the support set $\chi(\boldsymbol{\omega})$ of $\boldsymbol{\omega}$, i.e., the index set of the nonzero coordinates, is a stopping set according to [4, Def. 1]. Conversely, for any stopping set $\mathcal{S} = \mathcal{S}(\Pi, \Pi_{\rm c})$ of the 3D-TC there exists a pseudocodeword $\boldsymbol{\omega}$ with support set $\chi(\boldsymbol{\omega}) = \mathcal{S}$.

Proof: This result can be proved in the same manner as the corresponding result for conventional TCs [10, Lem. 2]. The proof given in [10] is based on the linearity of the subcodes $\bar{C}_{\rm a}$ and $\bar{C}_{\rm b}$ (from the stopping set definition in [10, Def. 1]). For 3D-TCs the same proof applies using the linearity of all the three subcodes $\hat{C}_{\rm a}$, $\hat{C}_{\rm b}$, and $\hat{C}_{\rm c}$ from [4, Def. 1]. ■

As a consequence of Proposition 3, it follows that $w_{\min}^{\rm AWGN}$ of C is upper-bounded by the $h_{\min}$ of C.

We remark that the $d_{\min}$ can be computed exactly by solving the *integer* program in (4) with $\lambda_l = 1$ for all l, with integer constraints on all the flow variables $\mathbf{f}$ in (1)–(3), i.e., $f_l \in \{0, 1\}$ for all l, and with the constraint $\sum_{l=0}^{N-1} y_l \geq 1$ to avoid obtaining the all-zero codeword (see [22, Prop. 3.6]). The exact $d_{\min}$ of 3D-TCs has not been computed before in the literature, but will be computed later in this paper for several codes. For instance, in [3], only estimates of $d_{\min}$ were provided. Finally, note that the exact $h_{\min}$ can be computed in a similar manner using *extended trellis modules* in $T_{\rm x}$ (see [23] for details).

V. Ensemble-Average Pseudoweight Enumerators

In this section, we describe how to compute the *ensemble-average pseudoweight enumerator* of 3D-TCs for a given graph cover degree m.

Here, we first introduce the concept of a *pseudocodeword vector-weight enumerator (PCVWE)* $\mathcal{P}_{\mathbf{w},\mathbf{h}}^{C_{\rm x}}$ of the constituent

code C_{x}. In particular, $\mathcal{P}_{\mathbf{w},\mathbf{h}}^{C_{\mathrm{x}}}$ is the number of *pseudocodewords* in the constituent code C_{x}, $\mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}$, of Hamming *vector-weight* $\mathbf{h} = (h_1, \ldots, h_m)$ corresponding to input sequences of Hamming *vector-weight* $\mathbf{w} = (w_1, \ldots, w_m)$. The pseudocodewords of a constituent code C_{x} are obtained as follows. Let $C_{\mathrm{x}}^{(m)}$ denote the degree-m cover of constituent code C_{x}, which is obtained by concatenating C_{x} by itself m times, i.e.,

$$C_{\mathrm{x}}^{(m)} = \big\{(\mathbf{x}_0, \ldots, \mathbf{x}_{m-1}) : \mathbf{x}_i \in C_{\mathrm{x}}, \forall i \in \{0, \ldots, m-1\}\big\}.$$

Now, let

$$\mathbf{x}^{(m)} = \big(x_0^{(0)}, \ldots, x_{N_{\mathrm{x}}-1}^{(0)}, \ldots, x_0^{(m-1)}, \ldots, x_{N_{\mathrm{x}}-1}^{(m-1)}\big)$$

denote a codeword in $C_{\mathrm{x}}^{(m)}$, where $(x_0^{(i)}, \ldots, x_{N_{\mathrm{x}}-1}^{(i)})$ is a codeword in C_{x} for all i, $0 \leq i \leq m-1$, and N_{x} is the block length of C_{x}. The corresponding *unnormalized* pseudocodeword is

$$\boldsymbol{\omega}(\mathbf{x}^{(m)}) = \left(\sum_{i=0}^{m-1} x_0^{(i)}, \ldots, \sum_{i=0}^{m-1} x_{N_{\mathrm{x}}-1}^{(i)}\right) \tag{6}$$

where addition is integer addition, which means that each component of a pseudocodeword is an integer between 0 and m. The jth component h_j of the vector-weight $\mathbf{h} = (h_1, \ldots, h_m)$ of the pseudocodeword in (6) is the number of components in the pseudocodeword with value j, i.e.,

$$h_j = \left|\left\{l : \sum_{i=0}^{m-1} x_l^{(i)} = j \text{ and } l \in \{0, \ldots, N_{\mathrm{x}}-1\}\right\}\right|.$$

The PCVWE of constituent code C_{x} can be computed using a nonbinary trellis constructed from the ordinary (information bit-oriented) trellis T_{x}. This trellis will be called the *pseudocodeword trellis* and is denoted by $T_{\mathrm{x},m}^{\mathrm{PC}}$ for constituent code C_{x}. The procedure to construct $T_{\mathrm{x},m}^{\mathrm{PC}}$ from T_{x} is described below.

A. Constructing $T_{\mathrm{x},m}^{\mathrm{PC}}$ From T_{x}

The pseudocodeword trellis $T_{\mathrm{x},m}^{\mathrm{PC}} = T_{\mathrm{x},m}^{\mathrm{PC}}(V_{\mathrm{x},m}^{\mathrm{PC}}, E_{\mathrm{x},m}^{\mathrm{PC}})$, where $V_{\mathrm{x},m}^{\mathrm{PC}}$ is the vertex set and $E_{\mathrm{x},m}^{\mathrm{PC}}$ is the edge set, can be constructed from the trellis T_{x} in the following way. First, define the sets

$$\begin{aligned}\tilde{V}_{\mathrm{x},m,t}^{\mathrm{PC}} &= \overbrace{V_{\mathrm{x},t} \times V_{\mathrm{x},t} \times \cdots \times V_{\mathrm{x},t}}^{m} \\ \tilde{E}_{\mathrm{x},m,t}^{\mathrm{PC}} &= \big\{((v_{\mathrm{l}}^{(0)}, \ldots, v_{\mathrm{l}}^{(m-1)}), (v_{\mathrm{r}}^{(0)}, \ldots, v_{\mathrm{r}}^{(m-1)})) : \\ &\qquad (v_{\mathrm{l}}^{(i)}, v_{\mathrm{r}}^{(i)}) \in E_{\mathrm{x},t}, \forall i \in \{0, \ldots, m-1\}\big\}\end{aligned}$$

where the time index t runs from 0 to I_{x} (resp. $I_{\mathrm{x}} - 1$) for the vertices (resp. edges). The label of an edge $((v_{\mathrm{l}}^{(0)}, \ldots, v_{\mathrm{l}}^{(m-1)}), (v_{\mathrm{r}}^{(0)}, \ldots, v_{\mathrm{r}}^{(m-1)})) \in \tilde{E}_{\mathrm{x},m,t}^{\mathrm{PC}}$ is the integer sum of the labels of its constituent edges $(v_{\mathrm{l}}^{(i)}, v_{\mathrm{r}}^{(i)}) \in E_{\mathrm{x},t}$ for all i, $0 \leq i \leq m-1$, which makes the trellis (to be constructed below) nonbinary in general.

Let $\Psi(\cdot)$ denote a permutation that reorders the components of a vertex $(v^{(0)}, \ldots, v^{(m-1)}) \in \tilde{V}_{\mathrm{x},m,t}^{\mathrm{PC}}$ according to their labels in a nondecreasing order. As an example, for $m = 3$, $\Psi(v_1, v_0, v_2) = (v_0, v_1, v_2)$ and $\Psi(v_2, v_1, v_0) = (v_0, v_1, v_2)$, assuming that vertex v_i has label i. Now, define the vertex set $V_{\mathrm{x},m,t}^{\mathrm{PC}}$ by expurgating vertices from the vertex set $\tilde{V}_{\mathrm{x},m,t}^{\mathrm{PC}}$ as follows:

$$V_{\mathrm{x},m,t}^{\mathrm{PC}} = \Big\{\Psi(v^{(0)}, \ldots, v^{(m-1)}) \colon (v^{(0)}, \ldots, v^{(m-1)}) \in \tilde{V}_{\mathrm{x},m,t}^{\mathrm{PC}}\Big\}.$$

The edge set $E_{\mathrm{x},m,t}^{\mathrm{PC}}$ is defined by expurgating edges from the edge set $\tilde{E}_{\mathrm{x},m,t}^{\mathrm{PC}}$ as follows:

$$\begin{aligned}E_{\mathrm{x},m,t}^{\mathrm{PC}} = \big\{&(\Psi(v_{\mathrm{l}}^{(0)}, \ldots, v_{\mathrm{l}}^{(m-1)}), \Psi(v_{\mathrm{r}}^{(0)}, \ldots, v_{\mathrm{r}}^{(m-1)})), \\ &\forall\, ((v_{\mathrm{l}}^{(0)}, \ldots, v_{\mathrm{l}}^{(m-1)}), (v_{\mathrm{r}}^{(0)}, \ldots, v_{\mathrm{r}}^{(m-1)})) \in \tilde{E}_{\mathrm{x},m,t}^{\mathrm{PC}}\big\}\end{aligned}$$

where all duplicated edges (edges with the same left and right vertex and edge label) are expurgated. The final pseudocodeword trellis is constructed by concatenating the trellis sections $T_{\mathrm{x},m,t}^{\mathrm{PC}} = T_{\mathrm{x},m,t}^{\mathrm{PC}}(V_{\mathrm{x},m,t}^{\mathrm{PC}}, E_{\mathrm{x},m,t}^{\mathrm{PC}})$, $t = 0, \ldots, I_{\mathrm{x}} - 1$.

As an example, in Fig. 4, we show both the standard trellis section $T_{\mathrm{x},t}$ (on the left) and the pseudocodeword trellis section $T_{\mathrm{x},m,t}^{\mathrm{PC}}$ for $m = 2$ (on the right), both being invariant of the time index t, of the accumulator code with generator polynomial $1/(1 + D)$. Note that for the pseudocodeword trellis section there are two edges with labels $2/1$ and $0/1$, respectively, from the middle vertex to the middle vertex.

Lemma 1: For $m = 2$, $|V_{\mathrm{x},m,t}^{\mathrm{PC}}| = |V_{\mathrm{x},t}| + \binom{|V_{\mathrm{x},t}|}{2}$ and $|E_{\mathrm{x},m,t}^{\mathrm{PC}}| = 2|V_{\mathrm{x},t}|^2 + |V_{\mathrm{x},t}|$.

Proof: See Appendix C. ∎

For a 4-state encoder this means 10 states, and for an 8-state encoder this means 36 states. For an accumulator we only have 3 states as can be seen in Fig. 4 (the right trellis section). Similar formulas for the number of vertices and edges can be derived for $m > 2$, but are omitted for brevity here.

Now, we define an *equivalence relation* on the set of pseudocodewords as follows. The pseudocodewords $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ are said to be *equivalent* if and only if there exists a positive real number Δ such that $\boldsymbol{\omega}_1 = \Delta \cdot \boldsymbol{\omega}_2$, i.e., they are scaled versions of each other. As a consequence only a single pseudocodeword from an equivalence class can be a vertex of the decoding polytope, which justifies counting equivalence classes only. Running a Viterbi-like algorithm (see Section V-B below for details) on the pseudocodeword trellis $T_{\mathrm{x},m}^{\mathrm{PC}}$ constructed above, will, in general, count pseudocodewords from the same equivalence class. However, counting pseudocodewords instead of their equivalence classes does not violate the bounding argument of Section V-E below, but may lead to a loose bound. For $m = 2$, for instance, these *duplicates* can be removed by a simple procedure which removes all terms of $\mathcal{P}_{\mathbf{w},\mathbf{h}}^{C_{\mathrm{x}}}$ with vector-weights $(\mathbf{w}, \mathbf{h}) = ((0, w), (0, h))$.

As a final remark, the issue of counting pseudocodewords from the same equivalence class is not considered in [14], [24] in the context of LDPC code ensembles.

B. Computing the PCVWE $\mathcal{P}_{\mathbf{w},\mathbf{h}}^{C_{\mathrm{x}}}$

The computation of the PCVWE $\mathcal{P}_{\mathbf{w},\mathbf{h}}^{C_{\mathrm{x}}}$ for a constituent code C_{x} can be performed (for a given cover degree m) on the corresponding pseudocodeword trellis $T_{\mathrm{x},m}^{\mathrm{PC}}$, similarly to the computation (on the trellis T_{x}) of the input-output weight enumerator. The algorithm performs I_{x} steps in the trellis.

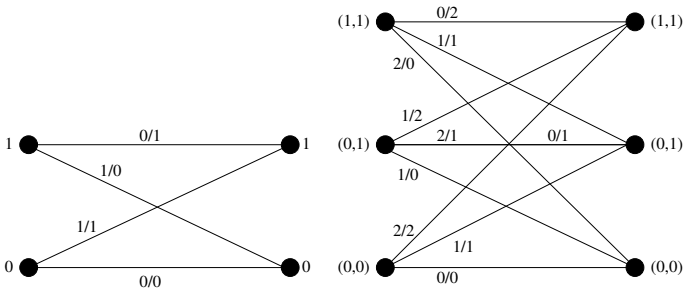

Fig. 4. The standard trellis section $T_{\mathrm{x},t}$ (on the left) and the pseudocodeword trellis section $T^{\mathrm{PC}}_{\mathrm{x},m,t}$ (on the right), both being invariant of the time index t, of the accumulator code for $m = 2$. Note that for the pseudocodeword trellis section there are two edges with labels $2/1$ and $0/1$, respectively, from the middle vertex to the middle vertex. The vertices of the pseudocodeword trellis section are labeled according to the vertex labeling of the standard trellis section on the left.

At trellis depth t, and for each state s, it computes the partial enumerator $\mathcal{P}^{C_{\mathrm{x}}}_{\mathbf{w},\mathbf{h}}(t,s)$ giving the number of paths in the trellis merging to state s at trellis depth t with input vector-weight $\mathbf{w}$ and output vector-weight $\mathbf{h}$. In particular, $\mathcal{P}^{C_{\mathrm{x}}}_{\mathbf{w},\mathbf{h}}(t,s)$ is computed from $\mathcal{P}^{C_{\mathrm{x}}}_{\mathbf{w},\mathbf{h}}(t-1, s^S(e))$ by considering all edges $e = (s^S(e), s^E(e))$ from starting state $s^S(e)$ (at time $t-1$) to ending state $s^E(e) = s$ (at time t) according to the dynamic programming principle. Finally, $\mathcal{P}^{C_{\mathrm{x}}}_{\mathbf{w},\mathbf{h}} = \mathcal{P}^{C_{\mathrm{x}}}_{\mathbf{w},\mathbf{h}}(I_{\mathrm{x}}, s_0)$, where s_0 denotes the all-zero state.

The number of required computations per trellis section is $|E^{\mathrm{PC}}_{\mathrm{x},m,t}| \cdot w^m_{\max} \cdot h^m_{\max}$, where $t = 0, \ldots, I_{\mathrm{x}} - 1$ and $w_{\max}$ (resp. $h_{\max}$) is the maximum entry of $\mathbf{w}$ (resp. $\mathbf{h}$) that we consider. Note that the computational complexity and the memory requirements scale exponentially with the cover degree m ($m = 1$ corresponds to the codeword input-output weight enumerator).

C. Average Pseudoweight Enumerator With Random Puncturing Pattern $\mathbf{p}$

We assume a random puncturing pattern for $\mathbf{p}$. In particular, the puncturing patterns are sampled uniformly at random from the ensemble of puncturing patterns $\mathbf{p}$ with a fraction of λ ones. Now, using the concept of *uniform interleaver*, the ensemble-average pseudocodeword input-output vector-weight enumerator is

$$\bar{\mathcal{P}}_{\mathbf{w},\mathbf{h}} = \sum_{\mathbf{q},\mathbf{q}_{\mathrm{a}},\mathbf{n}} \frac{\mathcal{P}^{C_{\mathrm{a}}}_{\mathbf{w},\mathbf{q}_{\mathrm{a}}} \mathcal{P}^{C_{\mathrm{b}}}_{\mathbf{w},\mathbf{q}-\mathbf{q}_{\mathrm{a}}}}{\binom{K}{w_1,w_2,\ldots,w_m}} \times \frac{\left[\prod_{i=1}^m \binom{q_i}{n_i}\right] \binom{2K-\sum_{i=1}^m q_i}{2\lambda K-\sum_{i=1}^m n_i}}{\binom{2K}{2\lambda K}} \frac{\mathcal{P}^{C_{\mathrm{c}}}_{\mathbf{n},\mathbf{h}-\mathbf{w}-\mathbf{q}+\mathbf{n}}}{\binom{2\lambda K}{n_1,n_2,\ldots,n_m}} \tag{7}$$

where $\bar{\mathcal{P}}_{\mathbf{w},\mathbf{h}}$ gives the average number (over all interleavers) of unnormalized pseudocodewords of input vector-weight $\mathbf{w}$ and output vector-weight $\mathbf{h}$. In (7),

$$\binom{K}{w_1, w_2, \ldots, w_m} = \frac{K!}{w_1! \cdots w_m! (K - \sum_{i=1}^m w_i)!},$$

$\mathbf{q}_{\mathrm{a}}$ is the output vector-weight from the constituent code C_{a}, $\mathbf{q}$ is the total output vector-weight from the outer turbo code, and $\mathbf{n}$ is input vector-weight for the inner constituent code C_{c}.

We remark that (7) can be seen as a nonbinary version of [3, Eq. (2)].

Now, the ensemble-average pseudoweight enumerator on channel $\mathcal{H}$ is

$$\bar{\mathcal{P}}^{\mathcal{H}}_w = \sum_{\mathbf{w}} \sum_{\substack{\mathbf{h}: \\ w^{\mathcal{H}}(\mathbf{h}) = w}} \bar{\mathcal{P}}_{\mathbf{w},\mathbf{h}}$$

where $w^{\mathcal{H}}(\mathbf{h})$ is the weight metric on $\mathcal{H}$. For instance, if $\mathcal{H}$ is the AWGN channel, then

$$w^{\mathcal{H}}(\mathbf{h}) = \left(\sum_{j=1}^m j \cdot h_j\right)^2 \Big/ \sum_{j=1}^m j^2 \cdot h_j$$

and if $\mathcal{H}$ is the binary erasure channel, then $w^{\mathcal{H}}(\mathbf{h}) = \sum_{j=1}^m h_j$.

D. Average Pseudoweight Enumerator With Regular Puncturing Pattern $\mathbf{p}$

In a similar fashion as for the case with a random puncturing pattern $\mathbf{p}$, we can modify [3, Eq. (3)] to arrive at a similar expression (to (7)) for the ensemble-average pseudocodeword input-output vector-weight enumerator. Details are omitted for brevity.

E. Finite-Length Minimum Pseudoweight Analysis

The ensemble-average pseudoweight enumerator $\bar{\mathcal{P}}^{\mathcal{H}}_w$ can be used to bound the minimum pseudoweight on $\mathcal{H}$, denoted by $w^{\mathcal{H}}_{\min}$, of the 3D-TC ensemble in the finite-length regime. In particular, the probability that a code randomly chosen from the ensemble has minimum pseudoweight $w^{\mathcal{H}}_{\min} < \bar{w}$ on $\mathcal{H}$ is upper-bounded by [25]

$$\Pr(w^{\mathcal{H}}_{\min} < \bar{w}) \leq \sum_{w>0}^{<\bar{w}} \bar{\mathcal{P}}^{\mathcal{H}}_w. \tag{8}$$

The upper bound in (8) can be used to obtain a probabilistic lower bound on the minimum pseudoweight of a code ensemble. For a fixed value of ϵ, where ϵ is any positive value between 0 and 1, we define the probabilistic lower bound with probability ϵ, denoted by $w^{\mathcal{H}}_{\min,\mathrm{LB},\epsilon}$, to be the largest real number $\bar{w}$ such that the right-hand side of (8) is at most ϵ. This guarantees that $\Pr(w^{\mathcal{H}}_{\min} \geq \bar{w}) \geq 1 - \epsilon$.

VI. Searching for the Minimum Pseudoweight

In this section, we present an efficient heuristic to search for low-weight pseudocodewords of 3D-TCs. We use the recently published improved minimum pseudoweight estimation algorithm by Chertkov and Stepanov [12]. In the following, we review that algorithm, restated for 3D-TCs and in a more convenient language.

Recall that the determination of $w^{\mathrm{AWGN}}_{\min}$ amounts to minimizing (5) over all nonzero vertices of $\mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$. Some important observations allow us to state an equivalent but simpler problem.

Koetter and Vontobel [26] already noted that

$$w^{\mathrm{AWGN}}_{\min} = \min_{\boldsymbol{\omega} \in \mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}) \setminus \{\mathbf{0}\}} w^{\mathrm{AWGN}}(\boldsymbol{\omega}) \tag{9}$$

where $\mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}})$ is the conic hull of $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ (also termed the fundamental cone). The statement follows immediately from the fact that $w^{\mathrm{AWGN}}(\boldsymbol{\omega}) = w^{\mathrm{AWGN}}(\tau\boldsymbol{\omega})$ for any pseudocodeword $\boldsymbol{\omega}$ and for all $\tau > 0$. The same property allows us to further restrict the search region to the conic section

$$\mathcal{F}_{\mathrm{sec}} = \mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}) \cap \{\boldsymbol{\omega} : \|\boldsymbol{\omega}\|_1 = 1\}$$

because every nonzero pseudocodeword may be scaled to satisfy the normalizing condition without changing the pseudoweight. The benefit of this step is twofold: First, in contrast to (9) the domain of optimization now is a polytope that can be stated explicitly by means of (in)equalities. Secondly, minimizing the pseudoweight $w^{\mathrm{AWGN}}(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_1^2/\|\boldsymbol{\omega}\|_2^2$ now is equivalent to maximizing $\|\boldsymbol{\omega}\|_2^2$, since the numerator is constant on $\mathcal{F}_{\mathrm{sec}}$.

We are thus in the situation of maximizing a convex function ($\|\cdot\|_2^2$) on a convex polytope. While this is an NP-hard problem in general, the following heuristic proposed in [12] gives very good results in practice.

For $\boldsymbol{\omega} \in \mathcal{F}_{\mathrm{sec}}$, $\|\boldsymbol{\omega}\|_2^2 = \|\boldsymbol{\omega} - \mathbf{1}/N\|_2^2 + \frac{1}{N}$, where $\mathbf{1}/N = (1,\ldots,1)/N$, i.e., our goal is to maximize, within $\mathcal{F}_{\mathrm{sec}}$, the distance to the central point $\mathbf{1}/N$ (the constant $\frac{1}{N}$ does not affect maximization). Chertkov and Stepanov proposed to first generate a random point $\boldsymbol{\omega}^{(0)} \neq \mathbf{1}/N$ on $\mathcal{F}_{\mathrm{sec}}$, serving as the initial search direction. Then, the linear program

$$\boldsymbol{\omega}^{(i+1)} = \arg\max (\boldsymbol{\omega}^{(i)} - \mathbf{1}/N)\boldsymbol{\omega}^T \text{ subject to } \boldsymbol{\omega} \in \mathcal{F}_{\mathrm{sec}}$$

where $(\cdot)^T$ denotes the transpose of its argument, is solved iteratively until the stopping criterion $\boldsymbol{\omega}^{(i+1)} = \boldsymbol{\omega}^{(i)}$ is reached. In each iteration, $\left\|\boldsymbol{\omega}^{(i)} - \mathbf{1}/N\right\|_1$ increases and therefore $\left\|\boldsymbol{\omega}^{(i)} - \mathbf{1}/N\right\|_2$ increases as well, and the result is a local maximum. The search is repeated for an arbitrary number of times in different random directions.

In the case of LDPC codes which are covered in [12], an explicit description of the polytope in question by means of inequalities is available, thus the fundamental cone can be described explicitly as well by omitting those inequalities which are not tight at $\boldsymbol{\omega} = \mathbf{0}$ [26]. This is however not the case for the polytope $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ which is only implicitly given as the projection of $\mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$ onto $\mathbf{y}$. Instead, as we will now show, the cone can be obtained by dropping upper bound constraints on all variables while ensuring that the total flow is equal on all three trellis graphs.

For $\mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}$, let $\mathcal{Q}_\mathrm{x}^\tau$ be defined as the set of all $\mathbf{f}^\mathrm{x} \in \mathbb{R}_{\geq 0}^{|E_\mathrm{x}|}$, where $\mathbb{R}$ is the real numbers, satisfying (2) and the following modified version of (1):

$$\sum_{e \in E_{\mathrm{x},0}} f_e^\mathrm{x} = \tau$$

and let

$$\mathcal{F} = \left\{\mathbf{f} = (\mathbf{f}^\mathrm{a}, \mathbf{f}^\mathrm{b}, \mathbf{f}^\mathrm{c}) : \exists \tau > 0 : \mathbf{f}^\mathrm{x} \in \mathcal{Q}_\mathrm{x}^\tau \text{ for } \mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}\right\}$$

which is, like $\mathcal{Q}$, the set of all network flows in the trellis graphs, but now with an arbitrary positive total flow τ instead of 1. Analogously to $\mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$, we define $\mathcal{F}_{\Pi,\Pi_\mathrm{c}}$ as the set of $(\tilde{\mathbf{y}}, \mathbf{f})$ where $\tilde{\mathbf{y}} \in \mathbb{R}_{\geq 0}^{N+2\lambda K}$ and $\mathbf{f} = (\mathbf{f}^\mathrm{a}, \mathbf{f}^\mathrm{b}, \mathbf{f}^\mathrm{c}) \in \mathcal{F}$ and additionally (3) is satisfied. The following lemma shows that the projection of $\mathcal{F}_{\Pi,\Pi_\mathrm{c}}$ onto $\mathbf{y}$ indeed yields the fundamental cone of 3D-TC LP decoding.

Lemma 2: Let $\dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}}$ be the projection of $\mathcal{F}_{\Pi,\Pi_\mathrm{c}}$ onto the first N variables. Then, $\dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}} = \mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}})$.

Proof: See Appendix D. ∎

VII. Numerical Results

In this section, we present some numerical results when the interleaver pair (Π, Π_c) is taken from the set of all possible interleaver pairs, and when it is taken from the set of pairs of quadratic permutation polynomials (QPPs) over integer rings. Permutation polynomial based interleavers over integer rings for conventional TCs were first proposed in [27]. These interleavers are fully algebraic and *maximum contention-free* [28], which makes them very suitable for parallel implementation in the turbo decoder. QPP-based interleavers for conventional TCs were also recently adopted for the 3GPP LTE standard [20]. We remark that for the results below, $\lambda = 1/4$ and the regular puncturing pattern $\mathbf{p} = [11000000]$ are assumed. As shown in [3], $\lambda = 1/4$ gives a suitable trade-off between performance in the waterfall and error floor regions. Finally, we emphasize that all the numerical estimates of the $d_{\min}$ and $w_{\min}^{\mathrm{AWGN}}$ given below are actually also upper bounds on the exact values.

A. Ensemble-Average Results for $K = 128$ and $R = 1/3$

In Fig. 5, we present the exact $d_{\min}$ and an estimate of $w_{\min}^{\mathrm{AWGN}}$ (which is also an upper bound), denoted by $\hat{w}_{\min}^{\mathrm{AWGN}}$, of unpunctured 3D-TCs with $K = 128$ and with 100 randomly selected pairs of interleavers (Π, Π_c) (blue plus signs). The corresponding results with QPP-based interleaver pairs (and with no constraints on the inverse polynomials) are also displayed (green x-marks). For all codes, except 11, the estimated $w_{\min}^{\mathrm{AWGN}}$ is at most equal to the $d_{\min}$. The values of $w_{\min}^{\mathrm{AWGN}}$ were estimated using the algorithm from [13] (which is straightforward to apply to 3D-TCs) with a signal-to-noise ratio (SNR) of 2.0 dB and 500 evaluations of the algorithm, while the $d_{\min}$ was computed exactly as described in the second paragraph following the proof of Proposition 3. Note that when the $d_{\min}$ is strictly smaller than the estimated $w_{\min}^{\mathrm{AWGN}}$ (points above the diagonal line), the estimation algorithm from [13] was unable to provide an estimate that beats the trivial upper bound provided by Proposition 3. From the figure, it follows that QPPs give *better codes* (can provide a higher $d_{\min}$ and a higher $\hat{w}_{\min}^{\mathrm{AWGN}}$), and that $w_{\min}^{\mathrm{AWGN}}$ is strictly lower than $d_{\min}$ for most codes when the $d_{\min}$ is large. As a side remark, the algorithm from Section VI gives slightly worse results (the average $\hat{w}_{\min}^{\mathrm{AWGN}}$ increases by approximately 0.05) than with the algorithm from [13] with the same number of runs (500) per instance. However, the algorithm from Section VI is significantly faster.

B. Exhaustive/Random Search Optimizing $w_{\min}^{\mathrm{AWGN}}$

In this subsection, we present the results of a computer search for pairs of QPPs with a quadratic inverse for $K = 128$,

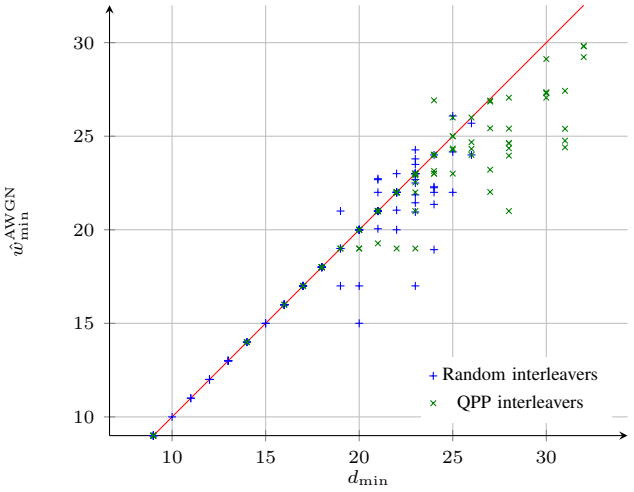

Fig. 5. *Estimated $w_{\min}^{\mathrm{AWGN}}$, using the algorithm from [13] (which is straightforward to apply to 3D-TCs), and exact $d_{\min}$ for 3D-TCs with* 100 *randomly selected pairs of interleavers (blue plus signs) and with* 100 *randomly selected pairs of QPP-based interleavers (green x-marks). The diagonal line gives the trivial upper bound of $d_{\min}$ on $w_{\min}^{\mathrm{AWGN}}$ provided by Proposition 3. $K = 128$ and $R = 1/3$.*

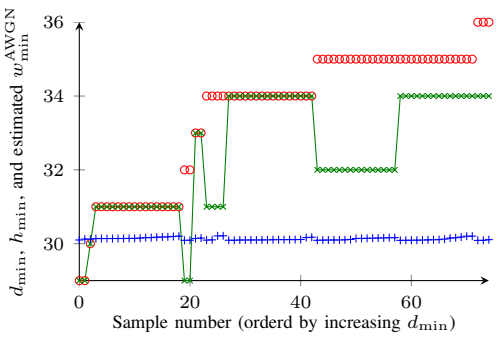

Fig. 6. *Exact $d_{\min}$ (red circles), exact $h_{\min}$ (green x-marks), and $\hat{w}_{\min}^{\mathrm{AWGN}}$ (blue plus signs) of the* 75 *best (in terms of $\hat{w}_{\min}^{\mathrm{AWGN}}$) QPP-based interleaver pairs for the 3D-TC with input block length $K = 128$ and code rate $R = 1/3$.*

256, and 320 for unpunctured $R = 1/3$ 3D-TCs. The objective of the search was to find pairs of QPPs giving a large estimated $w_{\min}^{\mathrm{AWGN}}$. To speed up the search, an adaptive threshold on the minimum AWGN pseudoweight $w_{\min}^{\mathrm{AWGN}}$ was set in the search, in the sense that if a pseudocodeword of AWGN pseudoweight smaller than the threshold was found, then this particular candidate pair of QPPs was rejected.

For $K = 128$, we performed an exhaustive search over all 2^{17} pairs of QPPs (with a quadratic inverse). The minimum AWGN pseudoweight was estimated using the algorithm from [13] (which is straightforward to apply to 3D-TCs) with an SNR of 1.7 dB and 500 evaluations of the algorithm. In Fig. 6, we plot the exact $d_{\min}$ (red circles), the exact $h_{\min}$ (green x-marks), and $\hat{w}_{\min}^{\mathrm{AWGN}}$ (blue plus signs) of the 75 3D-TCs with the best $\hat{w}_{\min}^{\mathrm{AWGN}}$. For each point in the figure, the x-coordinate is the sample index (the results are ordered by increasing $d_{\min}$), while the y-coordinate is either the exact $d_{\min}$, the exact $h_{\min}$, or $\hat{w}_{\min}^{\mathrm{AWGN}}$. From the figure, we observe that the best $w_{\min}^{\mathrm{AWGN}}$ (which is at most 30.2139) is strictly smaller than the best possible $d_{\min}$ or $h_{\min}$. The best possible $d_{\min}$ was established to be 38 (exhaustive search), and for this particular code $h_{\min} = 36$, but the estimate of $w_{\min}^{\mathrm{AWGN}}$ is not among the 75 best; it is only 29.6042 (see Table I which shows the results of an exhaustive/random search optimizing the $d_{\min}$ for pairs of QPPs with a quadratic inverse).

For $K = 256$, only a partial search has been conducted. The largest found value for $\hat{w}_{\min}^{\mathrm{AWGN}}$, after taking about 180000 samples, which is close to 17% of the whole space, is 43.0335. As for $K = 128$, the minimum AWGN pseudoweight was estimated using the algorithm from [13] (which is straightforward to apply to 3D-TCs) with an SNR of 1.7 dB and 500 evaluations of the algorithm.

For $K = 320$, we again performed an exhaustive search over the 2^{18} pairs of QPPs with a quadratic inverse. This time we used the algorithm presented in Section VI with 500 iterations per code. The largest estimated minimum pseudoweight $\hat{w}_{\min}^{\mathrm{AWGN}}$ that we found was 46.0612, which is considerably larger than that for the code in Table I (which shows the results of an exhaustive/random search optimizing the $d_{\min}$ for pairs of QPPs with a quadratic inverse).

C. Exhaustive/Random Search Optimizing $d_{\min}$

We also performed an exhaustive/random search optimizing the $d_{\min}$ for pairs of QPPs with a quadratic inverse for selected values of K for unpunctured $R = 1/3$ 3D-TCs. For $K = 128$, 160, 192, and 208, the search was exhaustive, in the sense that each pair of interleavers was looked at. In the search, the $d_{\min}$ was estimated using the triple impulse method [29]. The results are given in Table I for selected values of K, where $f(x) = f_1 x + f_2 x^2 \pmod K$ generates the TC interleaver, $\tilde{f}(x) = \tilde{f}_1 x + \tilde{f}_2 x^2 \pmod{N_c}$ generates the permutation in the patch, and $\hat{d}_{\min}$ and $\hat{w}_{\min}^{\mathrm{AWGN}}$ denote the estimated $d_{\min}$ and the estimated $w_{\min}^{\mathrm{AWGN}}$, respectively. The estimates of $w_{\min}^{\mathrm{AWGN}}$ were obtained by using the algorithm from [13] (which is straightforward to apply to 3D-TCs) with SNR and number of evaluations of the algorithm given in the ninth and tenth column of the table, respectively. Finally, we remark that the codes in the first and second rows, for $K = 128$ and 160, are $d_{\min}$-optimal, in the sense that there does not exist any pair of QPPs (with a quadratic inverse) giving a $d_{\min}$ strictly larger than 38 and 42, respectively, for the unpunctured 3D-TC.

D. Ensemble-Average Results for Various K and $R = 1/3$

In Fig. 7, we present the average estimated (now using the algorithm from Section VI) minimum AWGN pseudoweight of 3D-TCs for $K = 128$, 160, 192, 208, 256, 320, 512, 640, 768, 1024, and 1504. Both random interleaver pairs and QPP-based (with a quadratic inverse) interleaver pairs have been considered. In both cases, we generated 40 interleaver pairs of each size. For each code we ran $K/10$ trials of the estimation algorithm described in Section VI. From Fig. 7, we observe that the average $\hat{w}_{\min}^{\mathrm{AWGN}}$ grows with K for both random interleaver pairs and QPP-based interleaver pairs. For all values of K, as expected, the average $\hat{w}_{\min}^{\mathrm{AWGN}}$ is higher for QPP-based interleaver pairs than for random interleaver pairs. As a comparison, we have also plotted the corresponding

TABLE I
RESULTS FROM AN EXHAUSTIVE/RANDOM SEARCH FOR PAIRS OF QPPS WITH $\lambda = 1/4$, BOTH WITH A QUADRATIC INVERSE, IN WHICH THE FIRST QPP $f(x) = f_1 x + f_2 x^2 \pmod K$ GENERATES THE TC INTERLEAVER AND THE SECOND QPP $\tilde{f}(x) = \tilde{f}_1 x + \tilde{f}_2 x^2 \pmod{N_c}$ GENERATES THE PERMUTATION IN THE PATCH. MOREOVER, TERMS LIKE "SNR" ARE EXPLAINED IN THE TEXT

K	f_1	f_2	N_c	$\tilde{f}_1$	$\tilde{f}_2$	$\hat{d}_{\min}$	$\hat{w}_{\min}^{\text{AWGN}}$	SNR	Evaluations
128 [a]	55	96	64	9	16	38 [b]	29.6042	2.0 dB	2000
160 [a]	131	60	80	9	20	42 [b]	30.0000	1.7 dB	500
192 [a]	35	24	96	11	12	46	32.9046	1.7 dB	500
208 [a]	165	182	104	37	26	49	36.3370	1.7 dB	500
256	239	192	128	37	32	52	42.7816	1.7 dB	500
320	183	280	160	57	20	58	41.3818	1.7 dB	500
512 [c]	175	192	256	15	192	67	45.5872	2.0 dB	500

[a] Exhaustive search, which implies that the corresponding $\hat{d}_{\min}$ is an upper bound on the optimum $d_{\min}$ (the true optimum $d_{\min}$ when the estimate $\hat{d}_{\min}$ is exact) for this input block length. [b] This is the exact $d_{\min}$, and we can observe a large gap between $d_{\min}$ and $w_{\min}^{\text{AWGN}}$. [c] The QPPs are taken from [3].

theoretical values $w_{\min,\text{LB},0.5}^{\text{AWGN}}$ from Section V (using (8)) for graph cover degree 2. Also, for comparison, we have plotted the corresponding lower bounds on the $d_{\min}$ and the $h_{\min}$ using a similar ensemble analysis as the one from Section V. For details, we refer the interested reader to [3], [4]. Note that the curves coincide for small values of K. The reason that the curve for the probabilistic lower bound on $w_{\min}^{\text{AWGN}}$ of the 3D-TC ensemble is higher than the corresponding curve for $h_{\min}$ is that the cover degree is limited to $m = 2$. In general, the pseudocodewords with support set equal to a small-size stopping set which is not a codeword have a cover degree which is quite large. We would expect that the curve for the probabilistic lower bound on $w_{\min}^{\text{AWGN}}$ of the 3D-TC ensemble go further down for m larger than 2. However, it is currently unfeasible to do the actual computations for larger values of m (both the computational complexity and the memory requirements scale exponentially with m). For ease of computation we have used random puncturing patterns $\mathbf{p}$ to compute the curves, while the estimated average values are for regular patterns which in general give better results.

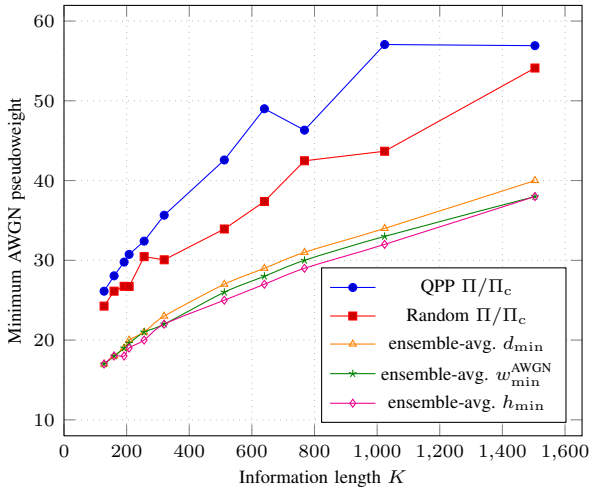

Fig. 7. The average estimated minimum AWGN pseudoweight for 3D-TCs for different information block lengths K, both for QPP-based (with a QPP inverse) and random interleaver pairs. The lower curves show the probabilistic lower bounds on $d_{\min}$, $w_{\min}^{\text{AWGN}}$, and $h_{\min}$ of the 3D-TC ensemble (for cover degrees of at most $m = 2$).

VIII. CONCLUSION

In this work, we performed a minimum pseudoweight analysis of pseudocodewords of (relaxed) LP decoding of 3D-TCs, adapting the LP relaxation proposed by Feldman in his thesis for conventional TCs. We proved that the 3D-TC polytope is proper and C-symmetric, and made a connection to finite graph covers of the 3D-TC factor graph. This connection was used to show that the support set of any pseudocodeword is a stopping set (as defined in [4, Def. 1]), and enabled a finite-length minimum pseudoweight analysis. Furthermore, an explicit description of the fundamental cone of the 3D-TC polytope was given. Finally, both a theoretical and an extensive numerical study of the minimum AWGN pseudoweight of small-to-medium block length 3D-TCs was presented, which showed that 1) typically (i.e., in most cases) when the $d_{\min}$ and/or the $h_{\min}$ is high, $w_{\min}^{\text{AWGN}}$ is strictly smaller than both the $d_{\min}$ and the $h_{\min}$ for these codes, and 2) that $w_{\min}^{\text{AWGN}}$ grows with the block length, at least for small-to-medium block lengths. For instance, the exhaustive search for $K = 128$ over the entire class of QPP-based interleaver pairs (with a quadratic inverse) revealed that the best minimum AWGN pseudoweight is strictly smaller than the best minimum/stopping distance. It is expected that the $w_{\min}^{\text{AWGN}}$ will dominate the decoding performance for high SNRs.

APPENDIX A
PROOF OF PROPOSITION 1

We first prove a more general result and then show how it applies to our case.

Lemma 3: Let C_δ, $\delta = 1, \ldots, \Delta$, be linear block codes of the same length N and let $C = \cap_{\delta=1}^{\Delta} C_\delta$. Then,

1) $\text{conv}(C_\delta)$ is proper and C_δ-symmetric for all δ, and
2) $\bigcap_{\delta=1}^{\Delta} \text{conv}(C_\delta)$ is proper and C-symmetric.

Proof: 1) The convex hull $\text{conv}(C_\delta)$ is proper because all codewords are by definition vertices of the polytope. Moreover, because no vertex of the unit hypercube is the convex

combination of others, $\mathrm{conv}(C_\delta)$ cannot contain any other integral points. To show C_δ-symmetry, choose $\mathbf{a} \in \mathrm{conv}(C_\delta)$ and $\mathbf{c} \in C_\delta$ arbitrarily. By construction, $\mathbf{a}$ can be written as a convex combination of codewords of C_δ, i.e.,

$$\mathbf{a} = \sum_{i=1}^{|C_\delta|} \lambda_i \mathbf{c}_i \quad \text{where} \sum_{i=1}^{|C_\delta|} \lambda_i = 1 \text{ and } \lambda_i \geq 0.$$

We claim that

$$|\mathbf{a} - \mathbf{c}| = \sum_{i=1}^{|C_\delta|} \lambda_i (\mathbf{c}_i \oplus \mathbf{c}) = \sum_{i=1}^{|C_\delta|} \lambda_i \tilde{\mathbf{c}}_i \tag{10}$$

where $\oplus$ denotes integer addition modulo 2 and $\tilde{\mathbf{c}}_i$ is the ith codeword of C_δ using a different ordering. This would imply C_δ-symmetry, i.e., if $\mathbf{a} \in \mathrm{conv}(C_\delta)$ and $\mathbf{c} \in C_\delta$, then $|\mathbf{a} - \mathbf{c}| \in \mathrm{conv}(C_\delta)$.

Let a_j, c_j, and $c_{i,j}$ denote the jth coordinate of $\mathbf{a}$, $\mathbf{c}$, and $\mathbf{c}_i$, respectively. The first equality in (10) follows for $c_j = 0$ from

$$|a_j - c_j| = a_j = \sum_{i=1}^{|C_\delta|} \lambda_i c_{i,j} = \sum_{i=1}^{|C_\delta|} \lambda_i (c_{i,j} \oplus c_j)$$

and for $c_j = 1$ because

$$|a_j - c_j| = 1 - a_j = \sum_{i=1}^{|C_\delta|} \lambda_i (1 - c_{i,j}) = \sum_{i=1}^{|C_\delta|} \lambda_i (c_{i,j} \oplus c_j).$$

The second part of (10) holds because $\mathbf{c}_i \oplus C_\delta = C_\delta$ due to the linearity of C_δ.

2) Let $\mathcal{P} = \cap_{\delta=1}^{\Delta} \mathrm{conv}(C_\delta)$. The properness of $\mathcal{P}$ for C follows immediately from the properness of $\mathrm{conv}(C_\delta)$, $\delta = 1, \ldots, \Delta$, and the definition of C. Now, if $\mathbf{a} \in \mathcal{P}$ and $\mathbf{c} \in C$, then for $\delta = 1, \ldots, \Delta$ we have $\mathbf{a} \in \mathrm{conv}(C_\delta)$ and $\mathbf{c} \in C_\delta$, so by 1) $|\mathbf{a} - \mathbf{c}| \in \mathrm{conv}(C_\delta)$ and thus $|\mathbf{a} - \mathbf{c}| \in \mathcal{P}$. ∎

Now, let C be a 3D-TC. By $\tilde{C}$ we denote the code of length $N + 2\lambda K$ obtained by appending the hidden parity bits from C_a and C_b which are sent to the patch. For $\mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}$ we define a supercode $\tilde{C}_\mathrm{x}$ of $\tilde{C}$ by unconstraining all bits that are not connected to the constituent code C_x, i.e., $\tilde{\mathbf{x}} \in \tilde{C}_\mathrm{x}$ if and only if $(\tilde{x}_{\rho_\mathrm{x}(0)}, \ldots, \tilde{x}_{\rho_\mathrm{x}(N_\mathrm{x}-1)}) \in C_\mathrm{x}$, where N_x is the block length of C_x and $\rho_\mathrm{x}(\cdot)$ is defined in (3), and $\tilde{x}_i \in \{0, 1\}$ for all remaining i. Observe that $\tilde{C}_\mathrm{a} \cap \tilde{C}_\mathrm{b} \cap \tilde{C}_\mathrm{c} = \tilde{C}$.

Next, define polytopes $\mathcal{Q}^\mathrm{x}_{\Pi,\Pi_\mathrm{c}}$ that are obtained from $\mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$ by dropping in (3) all constraints not corresponding to C_x, and let $\tilde{\mathcal{Q}}^\mathrm{x}_{\Pi,\Pi_\mathrm{c}}$ be the projection of $\mathcal{Q}^\mathrm{x}_{\Pi,\Pi_\mathrm{c}}$ onto the $\tilde{\mathbf{y}}$ variables. Finally, define $\tilde{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ in analogy to $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ as the projection of $\mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$ onto the first $N + 2\lambda K$ variables.

Due to the trellis structure, it is easily seen that $\tilde{\mathcal{Q}}^\mathrm{x}_{\Pi,\Pi_\mathrm{c}} = \mathrm{conv}(\tilde{C}_\mathrm{x})$ for $\mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}$, and by comparing the polytope definitions we see that $\tilde{\mathcal{Q}}^\mathrm{a}_{\Pi,\Pi_\mathrm{c}} \cap \tilde{\mathcal{Q}}^\mathrm{b}_{\Pi,\Pi_\mathrm{c}} \cap \tilde{\mathcal{Q}}^\mathrm{c}_{\Pi,\Pi_\mathrm{c}} = \tilde{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$. Applying Lemma 3 shows that $\tilde{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ is both proper and $\tilde{C}$-symmetric. Now, C is the projection of $\tilde{C}$ onto the first N variables, and $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ the according projection of $\tilde{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$.

To show that $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ is proper, first observe that the projection of any $\tilde{\mathbf{c}} \in \tilde{C}$ onto the first N variables is obviously contained in $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$. Conversely, let $\mathbf{a} \in \dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}} \cap \{0, 1\}^N$. By definition there exists $\tilde{\mathbf{a}} \in \tilde{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ such that $\tilde{\mathbf{a}} = (\mathbf{a}, \hat{\mathbf{a}})$. In order to show that $\hat{\mathbf{a}}$ is integral, note that the systematic part of $\tilde{\mathbf{a}}$ is contained in $\mathbf{a}$ and thus integral. Again by the trellis structure, this implies a unique and integral configuration of the flow variables in T_a and T_b, and consequently also the variables according to the output of those encoders, including the hidden bits sent to the patch, must be integral. Because $\tilde{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ is proper it follows that $\tilde{\mathbf{a}} \in \tilde{C}$ and thereby also $\mathbf{a} \in C$, which proves properness. Finally, note that C-symmetry is trivially preserved by projections, which concludes the proof.

Appendix B
Proof of Proposition 2

Let $\mathcal{Q}^\mathbf{f}_{\Pi,\Pi_\mathrm{c}} \subset \mathcal{Q}$ be the projection of $\mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$ onto $\mathbf{f}$. We call a flow $\mathbf{f} \in \mathcal{Q}$ *agreeable* if $\mathbf{f} \in \mathcal{Q}^\mathbf{f}_{\Pi,\Pi_\mathrm{c}}$. For an agreeable flow $\mathbf{f}$ let $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}(\mathbf{f})$ be the uniquely determined element of $[0, 1]^{N+2\lambda K}$ such that $(\tilde{\mathbf{y}}, \mathbf{f}) \in \mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$. Analogously, $\mathbf{y}(\mathbf{f})$ is the projection of $\tilde{\mathbf{y}}(\mathbf{f})$ onto the first N variables. Note that $\tilde{\mathbf{y}}(\mathbf{f})$ (and $\mathbf{y}(\mathbf{f})$) can be read off from $\mathbf{f}$ by (3).

For $\mathbf{f} = (\mathbf{f}^\mathrm{a}, \mathbf{f}^\mathrm{b}, \mathbf{f}^\mathrm{c}) \in \mathcal{Q}$, but not necessarily $\mathbf{f} \in \mathcal{Q}^\mathbf{f}_{\Pi,\Pi_\mathrm{c}}$, we can still use (3) to deduce local values of $\tilde{\mathbf{y}}$. More precisely, if we define

$$\Phi(\mathrm{x}) = \{j : j = \rho_\mathrm{x}(\phi_\mathrm{x}(t, i)) \text{ for some } (t, i)\}$$

then for all $j \in \Phi(\mathrm{x})$ we can deduce $\tilde{y}^\mathrm{x}_j(\mathbf{f}) = \sum_{\substack{e \in E_{\mathrm{x},t}: \\ c_i(e)=1}} f^\mathrm{x}_e$ where $t \in \{0, \ldots, I_\mathrm{x} - 1\}$ and $i \in \{0, \ldots, n_{\mathrm{x},t} - 1\}$ are determined by (3). This implies that $\mathbf{f} \in \mathcal{Q}$ is agreeable if and only if

$$\tilde{y}^\mathrm{x}_j(\mathbf{f}) = \tilde{y}^{\mathrm{x}'}_j(\mathbf{f}) \tag{11}$$

for all $(j, \mathrm{x}, \mathrm{x}')$ such that $j \in \Phi(\mathrm{x}) \cap \Phi(\mathrm{x}')$ and $(\mathrm{x}, \mathrm{x}') \in \{(\mathrm{a}, \mathrm{b}), (\mathrm{a}, \mathrm{c}), (\mathrm{b}, \mathrm{c})\}$, where the first case amounts to the outer interleaver Π and the remaining cases are due to the connections to the patch from C_a and C_b, respectively, via Π_c. We denote the set of these triples $(j, \mathrm{x}, \mathrm{x}')$ by $\mathcal{A}$.

Lemma 4: The relation $\mathcal{P}_{\Pi,\Pi_\mathrm{c}} \subseteq \dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ holds.

Proof: Let $\boldsymbol{\omega}(\mathbf{x}^{(m)}) \in \mathcal{P}_{\Pi,\Pi_\mathrm{c}}$ be a graph-cover pseudocodeword of C, i.e., there exists a degree-m cover code $C^{(m)}$ of C such that $\mathbf{x}^{(m)}$ is a codeword of $C^{(m)}$. As before, we can extend $\mathbf{x}^{(m)}$ to

$$\tilde{\mathbf{x}}^{(m)} = (\tilde{x}^{(0)}_0, \ldots, \tilde{x}^{(0)}_{N+2\lambda K-1}, \ldots, \tilde{x}^{(m-1)}_0, \ldots, \tilde{x}^{(m-1)}_{N+2\lambda K-1})$$

by appending the parity bits of the copies of C_a and C_b that are sent to copies of C_c.

For each $l = 0, \ldots, m-1$, $(\tilde{x}^{(l)}_0, \ldots, \tilde{x}^{(l)}_{N+2\lambda K-1})$ induces via trellis encoding a flow $\mathbf{f}^{(l)}_{\boldsymbol{\omega}}$ in $\mathcal{Q}$ with entries only from $\{0, 1\}$. In general, $\mathbf{f}^{(l)}_{\boldsymbol{\omega}}$ is not agreeable because the connections are mixed with different copies in the cover graph. However, from the definition of a graph cover we can conclude that

$$\tilde{y}^\mathrm{x}_j(\mathbf{f}^{(l)}_{\boldsymbol{\omega}}) = \tilde{y}^{\mathrm{x}'}_j(\mathbf{f}^{(\pi_j(l))}_{\boldsymbol{\omega}}) \tag{12}$$

for all $(j, \mathrm{x}, \mathrm{x}') \in \mathcal{A}$ and all $l = 0, \ldots, m-1$, where π_j is the corresponding permutation introduced by the graph cover, either (in the case $\mathrm{x} = \mathrm{a}$ and $\mathrm{x}' = \mathrm{b}$) on connections from an input vertex of $\Gamma(C_\mathrm{a})$ to a check vertex of $\Gamma(C_\mathrm{b})$ or (if $\mathrm{x}' = \mathrm{c}$) on connections from a parity vertex of C_a or C_b to a check vertex of $\Gamma(C_\mathrm{c})$.

We claim that

$$\mathbf{f}_{\boldsymbol{\omega}} = \frac{1}{m}\sum_{l=0}^{m-1} \mathbf{f}_{\boldsymbol{\omega}}^{(l)} \tag{13}$$

is agreeable and that $\mathbf{y}(\mathbf{f}_{\boldsymbol{\omega}}) = \boldsymbol{\omega}(\mathbf{x}^{(m)})$.

First, note that $\mathbf{f}_{\boldsymbol{\omega}}$ is a convex combination of elements from the convex set $\mathcal{Q}$, so $\mathbf{f}_{\boldsymbol{\omega}} \in \mathcal{Q}$ as well. To prove agreeability, we verify (11) for all $(j, \mathrm{x}, \mathrm{x}') \in \mathcal{A}$:

$$\begin{aligned}\tilde{y}_j^{\mathrm{x}}(\mathbf{f}_{\boldsymbol{\omega}}) = &\sum_{\substack{e\in E_{\mathrm{x},t}:\\ c_i(e)=1}} f_{\boldsymbol{\omega},e}^{\mathrm{x}} = \sum_{\substack{e\in E_{\mathrm{x},t}:\\ c_i(e)=1}} \frac{1}{m}\sum_{l=0}^{m-1} f_{\boldsymbol{\omega},e}^{\mathrm{x},(l)} \\ &= \frac{1}{m}\sum_{l=0}^{m-1} \tilde{y}_j^{\mathrm{x}}(\mathbf{f}_{\boldsymbol{\omega}}^{(l)}) = \frac{1}{m}\sum_{l=0}^{m-1} \tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_{\boldsymbol{\omega}}^{(\pi_j(l))}) \\ &= \frac{1}{m}\sum_{l=0}^{m-1} \tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_{\boldsymbol{\omega}}^{(l)}) = \tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_{\boldsymbol{\omega}})\end{aligned}$$

where we have used (3), (13), and (12). The second-to-last equality follows since π_j is a permutation of $\{0,\dots,m-1\}$. This shows that $\mathbf{f}_{\boldsymbol{\omega}}$ is agreeable and thus $\mathbf{y}(\mathbf{f}_{\boldsymbol{\omega}})$ is well-defined.

Now, fix $j \in \{0,\dots,N-1\}$ and pick any x such that $j \in \Phi(\mathrm{x})$. Then

$$\begin{aligned}y_j(\mathbf{f}_{\boldsymbol{\omega}}) = &\frac{1}{m}\sum_{l=0}^{m-1} \tilde{y}_j^{\mathrm{x}}(\mathbf{f}_{\boldsymbol{\omega}}^{(l)}) = \frac{1}{m}\sum_{l=0}^{m-1} x_j^{(l)} \\ &= \frac{1}{m}\left|\{l:\ x_j^{(l)} = 1\}\right| = \omega_j(\mathbf{x}^{(m)})\end{aligned}$$

which concludes the proof. ∎

Before proving the other direction for rational points, we first show part 2 of Proposition 2.

Lemma 5: All vertices of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$ have rational entries.

Proof: Let $\mathbf{y}$ be a vertex of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$. Because $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$ is a projection of the polytope $\mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$, and $\mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$ is the image of $\mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$ under a linear map, there exists some vertex $\mathbf{f}$ of $\mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$ such that $(\tilde{\mathbf{y}}(\mathbf{f}), \mathbf{f})$ is also a vertex of $\mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$ and $\mathbf{y}$ is the projection of $\tilde{\mathbf{y}}$ onto the first N variables.

Now, $\mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$ is a rational polyhedron (i.e., it is defined by (in)equalities with rational entries only), so every vertex of $\mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$ is rational [30, p. 123]. Since by (3) each $\tilde{y}_j$ is just a sum of entries of $\mathbf{f}$ for each $j = 0,\dots,N+2\lambda K-1$, $\tilde{\mathbf{y}}$ and thus $\mathbf{y}$ must be rational as well. ∎

Lemma 6: For every $\mathbf{y} \in \dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}} \cap \mathbb{Q}^N$ there exists a rational point $(\tilde{\mathbf{y}}, \mathbf{f}) \in \mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$ such that $\mathbf{y} = \mathbf{y}(\mathbf{f})$.

Proof: Let $\mathbf{y}$ be a rational point of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$. Because $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$ is a polytope, $\mathbf{y}$ can be written as a convex combination of vertices of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$, i.e., $\mathbf{y} = \sum_{k=0}^{d} \lambda_k \mathbf{y}^k$ where $\lambda_k \geq 0$ for $k = 0,\dots,d$ and $\sum_{k=0}^{d}\lambda_k = 1$. Furthermore, by Carathéodory's theorem (e.g., [31, p. 94]), this is even possible with some $d \leq N$ such that the $\mathbf{y}^k$, $k = 0,\dots,d$, are affinely independent. Consequently, $\boldsymbol{\lambda}$ is the unique solution of the system

$$\begin{pmatrix} \mathbf{y}^0 & \mathbf{y}^1 & \dots & \mathbf{y}^d \\ 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \vdots \\ \lambda_d \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix}$$

and by applying Cramer's rule for solving linear equation systems (and Lemma 5 which guarantees that all $\mathbf{y}^k$ have rational entries) we see that $\lambda_k \in \mathbb{Q}$ for $k = 0,\dots,d$. Furthermore, the proof of Lemma 5 tells us that for each $\mathbf{y}^k$ there is a rational $\mathbf{f}^k \in \mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$ such that $\mathbf{y}^k = \mathbf{y}(\mathbf{f}^k)$. The flow $\mathbf{f} = \sum_{k=0}^{d}\lambda_k \mathbf{f}^k$ satisfies $\mathbf{y} = \mathbf{y}(\mathbf{f})$ (because $\mathbf{y}(\cdot)$ is linear) and $(\tilde{\mathbf{y}}(\mathbf{f}), \mathbf{f}) \in \mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$ is rational, which concludes the proof. ∎

Now, we are able to prove the missing counterpart to Lemma 4.

Lemma 7: It holds that $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}} \cap \mathbb{Q}^N \subseteq \mathcal{P}_{\Pi,\Pi_{\mathrm{c}}}$.

Proof: Let $\mathbf{y}$ be a rational point of $\dot{\mathcal{Q}}_{\Pi,\Pi_{\mathrm{c}}}$. By Lemma 6, there exist rational $\mathbf{f} \in \mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$ and rational $\tilde{\mathbf{y}} = (\mathbf{y}, \hat{\mathbf{y}})$ such that $(\tilde{\mathbf{y}}, \mathbf{f}) \in \mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$. Let m be the least common denominator of the entries of $\mathbf{f}$. Then, $\mathbf{f}_m = m\mathbf{f}$ is a flow with integral values between 0 and m. Applying the flow decomposition theorem [32, p. 80] in this context guarantees that $\mathbf{f}_m$ can be split up into m binary flows, i.e.,

$$\mathbf{f}_m = \sum_{l=0}^{m-1} \mathbf{f}_m^{(l)} \tag{14}$$

where $\mathbf{f}_m^{(l)}$ has entries from $\{0,1\}$ and represents a valid path for each trellis T_{x}, $\mathrm{x} = \mathrm{a},\mathrm{b},\mathrm{c}$.

Because $\mathbf{f} \in \mathcal{Q}^{\mathbf{f}}_{\Pi,\Pi_{\mathrm{c}}}$, we conclude from (11) that $\tilde{y}_j^{\mathrm{x}}(\mathbf{f}) = \tilde{y}_j^{\mathrm{x}'}(\mathbf{f})$ for all $(j,\mathrm{x},\mathrm{x}') \in \mathcal{A}$. This is equivalent (by linearity) to $\tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m) = \tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_m)$ which by (14) means that $\sum_{l=0}^{m-1}\tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m^{(l)}) = \sum_{l=0}^{m-1}\tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_m^{(l)})$. Because all $\mathbf{f}_m^{(l)}$ are $\{0,1\}$-valued, this last equation implies

$$|\{l:\ \tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m^{(l)}) = 1\}| = |\{l:\ \tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_m^{(l)}) = 1\}|$$

and consequently for each $(j,\mathrm{x},\mathrm{x}') \in \mathcal{A}$ a permutation π_j on $\{0,\dots,m-1\}$ can be chosen such that $\tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m^{(l)}) = \tilde{y}_j^{\mathrm{x}'}(\mathbf{f}_m^{(\pi_j(l))})$ for all $l = 0,\dots,m-1$. These π_j define an m-cover $\Gamma^{(m)}(C)$ of $\Gamma(C)$, and by construction

$$\mathbf{x}^{(m)} = (x_0^{(0)},\dots,x_{N-1}^{(0)},\dots,x_0^{(m-1)},\dots,x_{N-1}^{(m-1)})$$

is a codeword of $C^{(m)}$, where we define $x_j^{(l)} = \tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m^{(l)})$ for the first x among $(\mathrm{a},\mathrm{b},\mathrm{c})$ such that $j \in \Phi(\mathrm{x})$. Finally, we see that

$$\begin{aligned}\omega_j(\mathbf{x}^{(m)}) &= \frac{1}{m}\sum_{l=0}^{m-1} x_j^{(l)} = \frac{1}{m}\sum_{l=0}^{m-1}\tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m^{(l)}) \quad \text{(for some x)} \\ &= \frac{1}{m}\tilde{y}_j^{\mathrm{x}}(\mathbf{f}_m) = \tilde{y}_j^{\mathrm{x}}(\mathbf{f}) = y_j\end{aligned}$$

(by definition of $\mathcal{Q}_{\Pi,\Pi_{\mathrm{c}}}$) for any $j = 0,\dots,N-1$, which shows that $\boldsymbol{\omega}(\mathbf{x}^{(m)}) = \mathbf{y}$. ∎

Appendix C
Proof of Lemma 1

The number of vertices in $V_{\mathrm{x},2,t}^{\mathrm{PC}}$ will be the number of distinct ordered 2-tuples of integers modulo $|V_{\mathrm{x},t}|$, which is $|V_{\mathrm{x},t}| + \binom{|V_{\mathrm{x},t}|}{2}$.

Now, let us consider the number of edges, and in particular, the edges from vertex $\Psi(v_{\mathrm{l}},u_{\mathrm{l}})$ to vertex $\Psi(v_{\mathrm{r}},u_{\mathrm{r}})$ in $T_{\mathrm{x},2,t}^{\mathrm{PC}}$, where $v_{\mathrm{l}},u_{\mathrm{l}},v_{\mathrm{r}},u_{\mathrm{r}} \in V_{\mathrm{x},t}$. We have four possible constituent edges to consider, namely the edges $v_{\mathrm{l}} \xrightarrow{a\ /\ b} v_{\mathrm{r}}$, $v_{\mathrm{l}} \xrightarrow{c\ /\ d} u_{\mathrm{r}}$, $u_{\mathrm{l}} \xrightarrow{e\ /\ f} u_{\mathrm{r}}$, and $u_{\mathrm{l}} \xrightarrow{g\ /\ h} v_{\mathrm{r}}$ where the labels above the

arrows are the input/output labels. Note that $a \neq c$ and $e \neq g$ when $v_\mathrm{r} \neq u_\mathrm{r}$. Also, the (integer) label of a vertex $v \in V_{\mathrm{x},t}$ will be denoted by $\ell(v)$.

- Case $v_\mathrm{l} \neq u_\mathrm{l}$ and $v_\mathrm{r} \neq u_\mathrm{r}$: All four constituent edges are distinct, and there will be four edges between the vertices $\Psi(v_\mathrm{l}, u_\mathrm{l})$ and $\Psi(v_\mathrm{r}, u_\mathrm{r})$ in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$ with labels a+e / b+f, c+g / d+h, g+c / h+d, and e+a / f+b. Since there are only two distinct labels (a+e / b+f and c+g / d+h are always distinct for a minimal trellis/encoder, details omitted for brevity), two of the edges can be removed.
- Case $v_\mathrm{l} \neq u_\mathrm{l}$ and $v_\mathrm{r} = u_\mathrm{r}$: In this case there are only two distinct constituent edges to consider, and there will be two edges between the vertices $\Psi(v_\mathrm{l}, u_\mathrm{l})$ and $\Psi(v_\mathrm{r}, u_\mathrm{r})$ in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$ with labels a+g / b+h and g+a / h+b (or c+e / d+f and e+c / f+d). Since both labels are the same, one of the edges can be removed.

In summary, for the first two cases above, we get a total of $\binom{|V_{\mathrm{x},t}|}{2} \cdot (2+1+1)$ edges in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$, since there are $\binom{|V_{\mathrm{x},t}|}{2}$ 2-tuples $(v_\mathrm{l}, u_\mathrm{l})$ with $\ell(v_\mathrm{l}) < \ell(u_\mathrm{l})$ and $v_\mathrm{l}, u_\mathrm{l} \in V_{\mathrm{x},t}$, and two possible values for $v_\mathrm{r} = u_\mathrm{r}$ in the second case (the label is either a+g / b+h or c+e / d+f).

- Case $v_\mathrm{l} = u_\mathrm{l}$ and $v_\mathrm{r} \neq u_\mathrm{r}$: In this case there are again only two distinct constituent edges to consider, and there will be two edges between the vertices $\Psi(v_\mathrm{l}, u_\mathrm{l})$ and $\Psi(v_\mathrm{r}, u_\mathrm{r})$ in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$ with labels a+c / b+d and c+a / d+b. Since both labels are the same, one of the edges can be removed.
- Case $v_\mathrm{l} = u_\mathrm{l}$ and $v_\mathrm{r} = u_\mathrm{r}$: In this case there is only one distinct constituent edge to consider, and there will be a single edge between the vertices $\Psi(v_\mathrm{l}, u_\mathrm{l})$ and $\Psi(v_\mathrm{r}, u_\mathrm{r})$ in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$ with label a+a / b+b (or c+c / d+d).

In summary, for the last two cases above, we get a total of $|V_{\mathrm{x},t}| \cdot (1+1+1)$ edges in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$, since there are $|V_{\mathrm{x},t}|$ 2-tuples $(v_\mathrm{l}, u_\mathrm{l})$ with $v_\mathrm{l} = u_\mathrm{l}$ and $v_\mathrm{l}, u_\mathrm{l} \in V_{\mathrm{x},t}$, and two possible values for $v_\mathrm{r} = u_\mathrm{r}$ in the fourth case (the label is either a+a / b+b or c+c / d+d).

In total, there are $4\binom{|V_{\mathrm{x},t}|}{2} + 3|V_{\mathrm{x},t}| = 2|V_{\mathrm{x},t}|^2 + |V_{\mathrm{x},t}|$ edges in $T^{\mathrm{PC}}_{\mathrm{x},2,t}$, which is the desired result.

Appendix D
Proof of Lemma 2

At first we show that $\dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}} \subseteq \mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}})$, so let $\mathbf{y} \in \dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}}$. By definition of $\dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}}$, this implies the existence of some $\mathbf{f} = (\mathbf{f}^\mathrm{a}, \mathbf{f}^\mathrm{b}, \mathbf{f}^\mathrm{c})$, $\hat{\mathbf{y}} \in \mathbb{R}^{2\lambda K}_{\geq 0}$, and $\tau > 0$ such that $((\mathbf{y}, \hat{\mathbf{y}}), \mathbf{f}) \in \mathcal{F}_{\Pi,\Pi_\mathrm{c}}$ and $\mathbf{f}^\mathrm{x} \in \mathcal{Q}^\tau_\mathrm{x}$ for $\mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}$. We will show that $(\tilde{\mathbf{y}}_\tau, \mathbf{f}_\tau) = (\frac{1}{\tau}(\mathbf{y}, \hat{\mathbf{y}}), \frac{1}{\tau}\mathbf{f}) \in \mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$, from which the claim follows because then $\mathbf{y} = \tau \mathbf{y}_\tau$ is a positive multiple of an element of $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$.

Conditions (2) and (3), which hold for $((\mathbf{y}, \hat{\mathbf{y}}), \mathbf{f})$ by definition of $\mathcal{F}_{\Pi,\Pi_\mathrm{c}}$, are invariant to scaling, so they hold for $(\tilde{\mathbf{y}}_\tau, \mathbf{f}_\tau)$ as well. Because $\mathbf{f}^\mathrm{x} \in \mathcal{Q}^\tau_\mathrm{x}$, it also follows that $\mathbf{f}^\mathrm{x}_\tau$ satisfies (1) for all $\mathrm{x} = \mathrm{a}, \mathrm{b}, \mathrm{c}$.

Equation (1) also ensures that the total $\mathbf{f}_\tau$-value in the first segment of each trellis T_x equals $\frac{1}{\tau}\tau = 1$, and because of (2) this must hold for all other trellis segments as well. Since $\mathbf{f}_\tau$ is also nonnegative, we can conclude from this that each entry of $\mathbf{f}_\tau$ lies in $[0, 1]$. But then also $\tilde{\mathbf{y}}_\tau \in [0, 1]^{N+2\lambda K}$ because each $\tilde{y}_j$, $j = 0, \ldots, N + 2\lambda K - 1$, is a subset of the total flow through a single segment and thus upper-bounded by 1, which concludes this part of the proof.

Now, we show that $\mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}) \subseteq \dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}}$. Let $\mathbf{y} \in \mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}})$. Since $\mathrm{conic}(\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}})$ is the conic hull of the convex set $\dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$, this implies the existence of some $\tau > 0$ and $\mathbf{y}_\mathcal{Q} \in \dot{\mathcal{Q}}_{\Pi,\Pi_\mathrm{c}}$ such that $\mathbf{y} = \tau \cdot \mathbf{y}_\mathcal{Q}$. To $\mathbf{y}_\mathcal{Q}$ then there must exist $\hat{\mathbf{y}}_\mathcal{Q}$ and $\mathbf{f}_\mathcal{Q}$ such that $((\mathbf{y}_\mathcal{Q}, \hat{\mathbf{y}}_\mathcal{Q}), \mathbf{f}_\mathcal{Q}) \in \mathcal{Q}_{\Pi,\Pi_\mathrm{c}}$, from which immediately it follows that $((\mathbf{y} = \tau \cdot \mathbf{y}_\mathcal{Q}, \tau\hat{\mathbf{y}}_\mathcal{Q}), \tau\mathbf{f}_\mathcal{Q}) \in \mathcal{F}_{\Pi,\Pi_\mathrm{c}}$, and thus $\mathbf{y} \in \dot{\mathcal{F}}_{\Pi,\Pi_\mathrm{c}}$.

Acknowledgment

The authors wish to thank the anonymous reviewers for their valuable comments and suggestions that helped improve the presentation of the paper.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] C. Berrou, A. Graell i Amat, Y. Ould-Cheikh-Mouhamedou, and Y. Saouter, "Improving the distance properties of turbo codes using a third component code: 3D turbo codes," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2505–2509, Sep. 2009.

[3] E. Rosnes and A. Graell i Amat, "Performance analysis of 3-D turbo codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3707–3720, Jun. 2011.

[4] A. Graell i Amat and E. Rosnes, "Stopping set analysis of 3-dimensional turbo code ensembles," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 2013–2017.

[5] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta, K. S. Zigangirov, and D. J. Costello, Jr., "Analysis and design of tuned turbo codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4796–4813, Jul. 2012.

[6] J. Feldman, D. R. Karger, and M. Wainwright, "Linear programming-based decoding of turbo-like codes and its relation to iterative approaches," in *Proc. 40th Annual Allerton Conf. Commun., Control, and Computing*, Monticello, IL, Oct. 2002.

[7] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, 2003.

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

[9] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," Computing Research Repository (CoRR). [Online]. Available: http://arxiv.org/abs/cs.IT/0512078/

[10] E. Rosnes, "On the connection between finite graph covers, pseudo-codewords, and linear programming decoding of turbo codes," in *Proc. 4th Int. Symp. Turbo Codes & Rel. Topics*, Munich, Germany, Apr. 2006.

[11] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[12] M. Chertkov and M. G. Stepanov, "Polytope of correct (linear programming) decoding and low-weight pseudo-codewords," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, Jul./Aug. 2011, pp. 1648–1652.

[13] ——, "An efficient pseudocodeword search algorithm for linear programming decoding of LDPC codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1514–1520, Apr. 2008.

[14] S. Abu-Surra, D. Divsalar, and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 858–886, Feb. 2011.

[15] D. Divsalar and L. Dolecek, "Graph cover ensembles of non-binary protograph LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, Jul. 2012, pp. 2526–2530.

[16] ——, "Ensemble analysis of pseudocodewords of protograph-based non-binary LDPC codes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Paraty, Brazil, Oct. 2011, pp. 340–344.

[17] N. Boston, "A multivariate weight enumerator for tail-biting trellis pseudocodewords," in *Proc. Workshop on Algebra, Combinatorics and Dynamics*, Belfast, Northern Ireland, Aug. 2009.

[18] D. Conti and N. Boston, "Matrix representations of trellises and enumerating trellis pseudocodewords," in *Proc. 49th Annual Allerton Conf. Commun., Control, and Computing*, Monticello, IL, Sep. 2011, pp. 1438–1445.

[19] P. Vontobel, "Counting in graph covers: A combinatorial characterization of the Bethe entropy function," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 6018–6048, Sep. 2013.

[20] "3rd generation partnership project; technical specification group radio access network; evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding (release 8)," Dec. 2008, 3GPP TS 36.212 v8.5.0.

[21] G. D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznik, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," in *Codes, Systems, and Graphical Models*, B. Marcus and J. Rosenthal, Eds., vol. 123 of IMA Vol. Math. Appl. Springer Verlag, 2001, pp. 101–112.

[22] A. Tanatmis, S. Ruzika, H. W. Hamacher, M. Punekar, F. Kienle, and N. Wehn, "Valid inequalities for binary linear codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, Korea, Jun./Jul. 2009, pp. 2216–2220.

[23] E. Rosnes and Ø. Ytrehus, "Turbo decoding on the binary erasure channel: Finite-length analysis and turbo stopping sets," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4059–4075, Nov. 2007.

[24] M. F. Flanagan, "Exposing pseudoweight layers in regular LDPC code ensembles," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Taormina, Italy, Oct. 2009, pp. 60–64.

[25] H. D. Pfister and P. H. Siegel, "The serial concatenation of rate-1 codes through uniform random interleavers," *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1425–1438, Jun. 2003.

[26] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. 3rd Int. Symp. Turbo Codes & Rel. Topics*, Brest, France, Sep. 2003, pp. 75–82.

[27] J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 101–119, Jan. 2005.

[28] O. Y. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1249–1253, Mar. 2006.

[29] S. Crozier, P. Guinand, and A. Hunt, "Computing the minimum distance of turbo-codes using iterative decoding techniques," in *Proc. 22nd Biennial Symp. Commun.*, Kingston, ON, Canada, May/Jun. 2004, pp. 306–308.

[30] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. Wiley-Interscience series in discrete mathematics and optimization, John Wiley & Sons, 1988.

[31] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[32] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Prentice Hall, 1993.