Multidimensional Decoding Networks for Trapping Set Analysis

August 28, 2017

Allison Beemer & Christine A. Kelley

University of Nebraska - Lincoln
Department of Mathematics

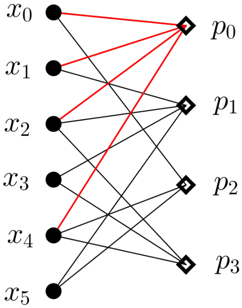
Outline

1. LDPC Codes & Trapping Sets
2. Multidimensional Decoding Networks
3. Trapping Set Characterization
4. Applications

Low-density parity-check (LDPC) codes

- Codes constructed from sparse parity-check matrices
- Can be viewed in the form of a **Tanner graph**
  - $x_0 + x_1 + x_2 + x_4 = 0$

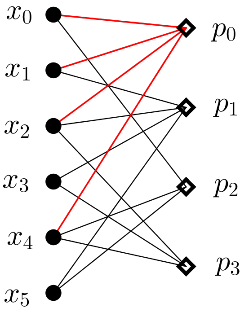
$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$



Low-density parity-check (LDPC) codes

- Codes constructed from sparse parity-check matrices
- Can be viewed in the form of a **Tanner graph**
  - $x_0 + x_1 + x_2 + x_4 = 0$
- Asymptotically good [Gallager '62]
- Capacity-approaching [Richardson, Shokrollahi, Urbanke '01]
- Efficient graph-based decoders

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



Iterative Graph-Based Decoding Algorithms

Messages are passed back and forth across the edges of the Tanner graph in order to decode.

Iterative Graph-Based Decoding Algorithms

Messages are passed back and forth across the edges of the Tanner graph in order to decode.

- Bit-flipping algorithms
  - For example: flip the values of bits that are adjacent to more unsatisfied check nodes than satisfied.

Iterative Graph-Based Decoding Algorithms

Messages are passed back and forth across the edges of the Tanner graph in order to decode.

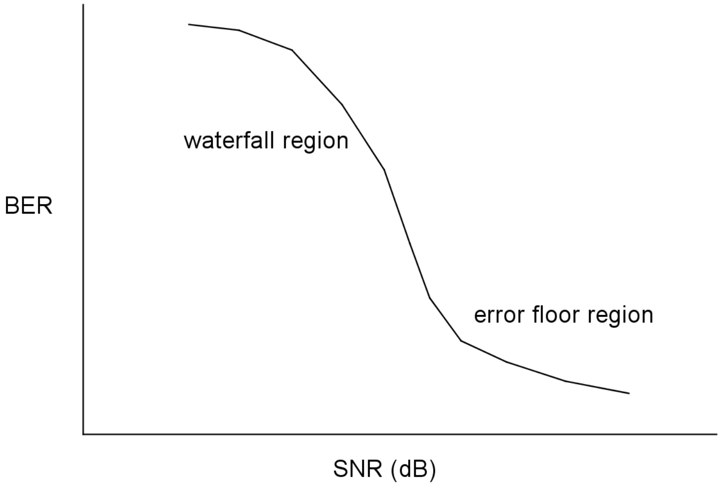
- Bit-flipping algorithms
  - For example: flip the values of bits that are adjacent to more unsatisfied check nodes than satisfied.
- Gallager A/B
  - Variable to check: send the value received from the channel, unless all (or some portion of) the other incoming messages disagree.
  - Check to variable: send the sum of all other incoming messages.

Iterative Graph-Based Decoding Algorithms

Messages are passed back and forth across the edges of the Tanner graph in order to decode.

- Bit-flipping algorithms
  - For example: flip the values of bits that are adjacent to more unsatisfied check nodes than satisfied.
- Gallager A/B
  - Variable to check: send the value received from the channel, unless all (or some portion of) the other incoming messages disagree.
  - Check to variable: send the sum of all other incoming messages.
- Belief-Propogation
  - "Soft" information is passed between the parts of the Tanner graph.

Error Floor



Trapping sets

Suppose that the codeword $\mathbf{x}$ is transmitted, and $\hat{\mathbf{x}}$ is received. Let $\mathcal{T}(\hat{\mathbf{x}})$ denote the set of variable nodes in the Tanner graph G that are not eventually correct as the decoder runs on $\hat{\mathbf{x}}$.

Trapping sets

Suppose that the codeword $\mathbf{x}$ is transmitted, and $\hat{\mathbf{x}}$ is received. Let $\mathcal{T}(\hat{\mathbf{x}})$ denote the set of variable nodes in the Tanner graph G that are not eventually correct as the decoder runs on $\hat{\mathbf{x}}$.

- $\mathcal{T}(\hat{\mathbf{x}})$ is an (a, b)-**trapping set** if $|\mathcal{T}(\hat{\mathbf{x}})| = a$, and $G[\mathcal{T}]$, the subgraph it induces with its neighbors, has b odd-degree check nodes [Richardson '03].

Trapping sets

Suppose that the codeword $\mathbf{x}$ is transmitted, and $\hat{\mathbf{x}}$ is received. Let $\mathcal{T}(\hat{\mathbf{x}})$ denote the set of variable nodes in the Tanner graph G that are not eventually correct as the decoder runs on $\hat{\mathbf{x}}$.

- $\mathcal{T}(\hat{\mathbf{x}})$ is an (a,b)-**trapping set** if $|\mathcal{T}(\hat{\mathbf{x}})| = a$, and $G[\mathcal{T}]$, the subgraph it induces with its neighbors, has b odd-degree check nodes [Richardson '03].
- The set of variable nodes in error in $\hat{\mathbf{x}}$ is an **inducing set** for $\mathcal{T}(\hat{\mathbf{x}})$.

Trapping sets

Suppose that the codeword $\mathbf{x}$ is transmitted, and $\hat{\mathbf{x}}$ is received. Let $\mathcal{T}(\hat{\mathbf{x}})$ denote the set of variable nodes in the Tanner graph G that are not eventually correct as the decoder runs on $\hat{\mathbf{x}}$.

- $\mathcal{T}(\hat{\mathbf{x}})$ is an (a, b)-**trapping set** if $|\mathcal{T}(\hat{\mathbf{x}})| = a$, and $G[\mathcal{T}]$, the subgraph it induces with its neighbors, has b odd-degree check nodes [Richardson '03].
- The set of variable nodes in error in $\hat{\mathbf{x}}$ is an **inducing set** for $\mathcal{T}(\hat{\mathbf{x}})$.
- The size of a minimum inducing set of a trapping set is its **critical number**.

Trapping sets

Suppose that the codeword $\mathbf{x}$ is transmitted, and $\hat{\mathbf{x}}$ is received. Let $\mathcal{T}(\hat{\mathbf{x}})$ denote the set of variable nodes in the Tanner graph G that are not eventually correct as the decoder runs on $\hat{\mathbf{x}}$.

- $\mathcal{T}(\hat{\mathbf{x}})$ is an (a, b)-**trapping set** if $|\mathcal{T}(\hat{\mathbf{x}})| = a$, and $G[\mathcal{T}]$, the subgraph it induces with its neighbors, has b odd-degree check nodes [Richardson '03].
- The set of variable nodes in error in $\hat{\mathbf{x}}$ is an **inducing set** for $\mathcal{T}(\hat{\mathbf{x}})$.
- The size of a minimum inducing set of a trapping set is its **critical number**.
- Notice that trapping sets are heavily decoder-dependent.
- Under certain symmetry conditions on the channel and decoder, we may assume that the codeword $\mathbf{0}$ was sent.

Trapping Sets of Familiar Decoders

- **Stopping sets:** trapping sets over the Binary Erasure Channel (BEC) with a peeling decoder.
- **Fully absorbing sets:** trapping sets over the Binary Symmetric Channel (BSC) with a parallel bit-flipping decoding algorithm (e.g. the "Simple Parallel Decoding Algorithm" of [Sipser, Spielman '96]).

Trapping Sets of Familiar Decoders

- **Stopping sets:** trapping sets over the Binary Erasure Channel (BEC) with a peeling decoder.
- **Fully absorbing sets:** trapping sets over the Binary Symmetric Channel (BSC) with a parallel bit-flipping decoding algorithm (e.g. the "Simple Parallel Decoding Algorithm" of [Sipser, Spielman '96]).
- Trapping sets grow more complicated with more involved decoders, and lose their combinatorial definitions.

Trapping Sets of Familiar Decoders

- **Stopping sets:** trapping sets over the Binary Erasure Channel (BEC) with a peeling decoder.
- **Fully absorbing sets:** trapping sets over the Binary Symmetric Channel (BSC) with a parallel bit-flipping decoding algorithm (e.g. the "Simple Parallel Decoding Algorithm" of [Sipser, Spielman '96]).
- Trapping sets grow more complicated with more involved decoders, and lose their combinatorial definitions.
- Simulation results suggest that structures such as absorbing sets affect performance regardless of chosen decoder.

Trapping Sets of Familiar Decoders

- **Stopping sets:** trapping sets over the Binary Erasure Channel (BEC) with a peeling decoder.
- **Fully absorbing sets:** trapping sets over the Binary Symmetric Channel (BSC) with a parallel bit-flipping decoding algorithm (e.g. the "Simple Parallel Decoding Algorithm" of [Sipser, Spielman '96]).
- Trapping sets grow more complicated with more involved decoders, and lose their combinatorial definitions.
- Simulation results suggest that structures such as absorbing sets affect performance regardless of chosen decoder.
- But, we would like a more efficient method of finding trapping sets specific to the chosen decoder.

Multidimensional Networks

Definition (Berlingerio et. al. '11)

A *multidimensional network* is an edge-labeled directed graph $\mathcal{D} = (V, E, D)$, where V is a set of vertices, D a set of edge labels, called *dimensions*, and E is a set of triples of the form (u, v, d) where $u, v \in V$ and $d \in D$.

We say that an edge (or vertex) *belongs to* a given dimension d if it is labeled d (or is incident to an edge labeled d).

Multidimensional Decoding Networks: Example

- The binary repetition code of length 3, defined by the parity-check matrix H shown below, for transmission over the BSC with Gallager A decoding.

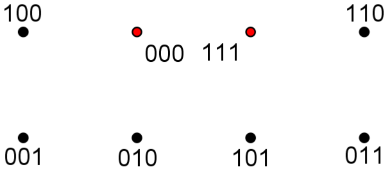
$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Multidimensional Decoding Networks: Example

- The binary repetition code of length 3, defined by the parity-check matrix H shown below, for transmission over the BSC with Gallager A decoding.

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

- Vertex set: $V = \{\mathbf{x} : \mathbf{x} \in \mathbb{F}_2^3\}$ (more generally: $\{\mathbf{x} : \mathbf{x} \in S\}$)

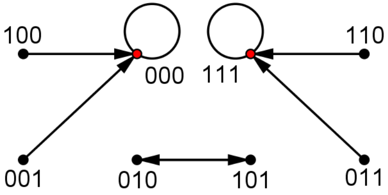


Multidimensional Decoding Networks: Example

- The binary repetition code of length 3, defined by the parity-check matrix H shown below, for transmission over the BSC with Gallager A decoding.

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

- Vertex set: $V = \{\mathbf{x} : \mathbf{x} \in \mathbb{F}_2^3\}$ (more generally: $\{\mathbf{x} : \mathbf{x} \in S^n\}$)
- Edge set: $E = \bigcup_{\ell \geq 1} \{(\mathbf{x}_i, \mathbf{x}_j, \ell) : \mathbf{x}_i^\ell = \mathbf{x}_j\}$

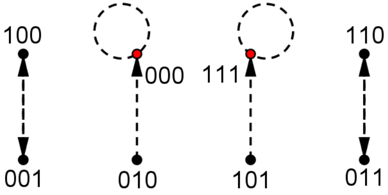


Multidimensional Decoding Networks: Example

- The binary repetition code of length 3, defined by the parity-check matrix H shown below, for transmission over the BSC with Gallager A decoding.

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

- Vertex set: $V = \{\mathbf{x} : \mathbf{x} \in \mathbb{F}_2^3\}$ (more generally: $\{\mathbf{x} : \mathbf{x} \in S^n\}$)
- Edge set: $E = \bigcup_{\ell \geq 1} \{(\mathbf{x}_i, \mathbf{x}_j, \ell) : \mathbf{x}_i^\ell = \mathbf{x}_j\}$

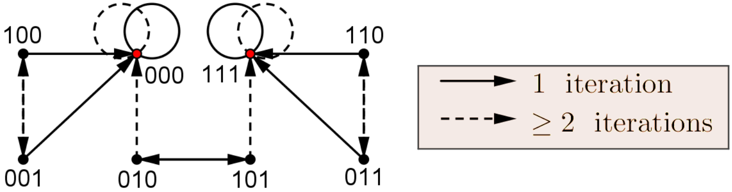


Multidimensional Decoding Networks: Example

- The binary repetition code of length 3, defined by the parity-check matrix H shown below, for transmission over the BSC with Gallager A decoding.

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

- Vertex set: $V = \{\mathbf{x} : \mathbf{x} \in \mathbb{F}_2^3\}$ (more generally: $\{\mathbf{x} : \mathbf{x} \in S^n\}$)
- Edge set: $E = \bigcup_{\ell \geq 1} \{(\mathbf{x}_i, \mathbf{x}_j, \ell) : \mathbf{x}_i^\ell = \mathbf{x}_j\}$



Multidimensional Decoding Networks

Factors that affect the structure of the decoding network:

- Channel over which information is transmitted
- Choice of decoder
- Tanner graph representation (i.e. choice of parity-check matrix)

Transitive Decoding Networks

Definition

We say a decoding network is **transitive** if $(v_1, v_2, \ell) \in E$ if and only if for every choice of $1 \leq k \leq \ell - 1$, there exists $v_k \in V$ such that (v_1, v_k, k), $(v_k, v_2, \ell - k) \in E$.

Transitive Decoding Networks

Definition

We say a decoding network is **transitive** if $(v_1, v_2, \ell) \in E$ if and only if for every choice of $1 \leq k \leq \ell - 1$, there exists $v_k \in V$ such that (v_1, v_k, k), $(v_k, v_2, \ell - k) \in E$.

- If a network is transitive, edges in dimension ℓ correspond to directed paths of length ℓ in dimension 1.

Transitive Decoding Networks

Definition

We say a decoding network is **transitive** if $(v_1, v_2, \ell) \in E$ if and only if for every choice of $1 \leq k \leq \ell - 1$, there exists $v_k \in V$ such that (v_1, v_k, k), $(v_k, v_2, \ell - k) \in E$.

- If a network is transitive, edges in dimension ℓ correspond to directed paths of length ℓ in dimension 1.
- The peeling decoder over the BEC admits a transitive decoding network.

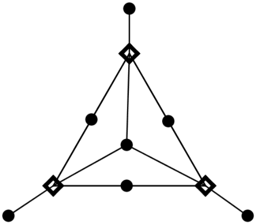
Transitive Decoding Networks

Definition

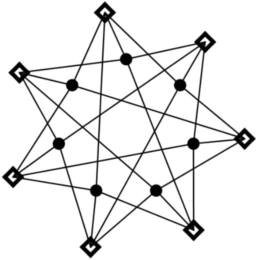
We say a decoding network is **transitive** if $(v_1, v_2, \ell) \in E$ if and only if for every choice of $1 \leq k \leq \ell - 1$, there exists $v_k \in V$ such that (v_1, v_k, k), $(v_k, v_2, \ell - k) \in E$.

- If a network is transitive, edges in dimension ℓ correspond to directed paths of length ℓ in dimension 1.
- The peeling decoder over the BEC admits a transitive decoding network.
- Bit-flipping algorithms that ignore channel values also yield transitive decoding networks.

Example



A



B

Two distinct Tanner graph representations of $\mathscr{H}_3$. Variable nodes are denoted by $\bullet$, and check nodes are denoted by $\Diamond$. Representation A does not admit a transitive decoding network under Gallager A decoding, but Representation B does. While a representation may be transitive without containing all (nonzero) codewords of the dual, transitivity is dependent on the particular choice of representation, and not only the number of included parity checks. In particular, increasing redundancy does necessarily preserve transitivity.

Gallager A and Transitivity

Theorem

Every binary linear code has a parity-check matrix representation whose corresponding decoding network is transitive under Gallager A decoding.

Trapping Sets in the Decoding Network

Theorem

For each vertex $\mathbf{x} \in V = \mathbb{F}_2^n$ *in a decoding network* $\mathcal{D} = (V, E, D)$ *for a code* $\mathcal{C}$*, let* $M_{\mathbf{x}}$ *be the set of vertices* $\mathbf{y} \in V$ *for which there is an edge* $(\mathbf{x}, \mathbf{y}, \ell) \in E$ *for infinitely many choices of* ℓ*. Then the set of variable nodes corresponding to*

$$\bigcup_{\mathbf{y} \in M_{\mathbf{x}}} \mathsf{supp}(\mathbf{y}) \ ,$$

denoted $\mathcal{T}(\mathbf{x})$*, is a trapping set with an inducing set given by the variable nodes corresponding to* $\mathsf{supp}(\mathbf{x})$*.*

Trapping Sets in the Decoding Network

Theorem

For each vertex $\mathbf{x} \in V = \mathbb{F}_2^n$ *in a decoding network* $\mathcal{D} = (V, E, D)$ *for a code* $\mathcal{C}$*, let* $M_{\mathbf{x}}$ *be the set of vertices* $\mathbf{y} \in V$ *for which there is an edge* $(\mathbf{x}, \mathbf{y}, \ell) \in E$ *for infinitely many choices of* ℓ*. Then the set of variable nodes corresponding to*

$$\bigcup_{\mathbf{y} \in M_{\mathbf{x}}} \mathsf{supp}(\mathbf{y}) \ ,$$

denoted $\mathcal{T}(\mathbf{x})$*, is a trapping set with an inducing set given by the variable nodes corresponding to* $\mathsf{supp}(\mathbf{x})$*. Given a trapping set* $\mathcal{T}$*, its set of inducing sets is given by the variable nodes corresponding to*

$$\{\mathsf{supp}(\mathbf{x}) : \mathcal{T}(\mathbf{x}) = \mathcal{T}\} \ ,$$

Trapping Sets in the Decoding Network

Theorem

For each vertex $\mathbf{x} \in V = \mathbb{F}_2^n$ *in a decoding network* $\mathcal{D} = (V, E, D)$ *for a code* $\mathcal{C}$*, let* $M_{\mathbf{x}}$ *be the set of vertices* $\mathbf{y} \in V$ *for which there is an edge* $(\mathbf{x}, \mathbf{y}, \ell) \in E$ *for infinitely many choices of* ℓ*. Then the set of variable nodes corresponding to*

$$\bigcup_{\mathbf{y} \in M_{\mathbf{x}}} \mathsf{supp}(\mathbf{y}) \ ,$$

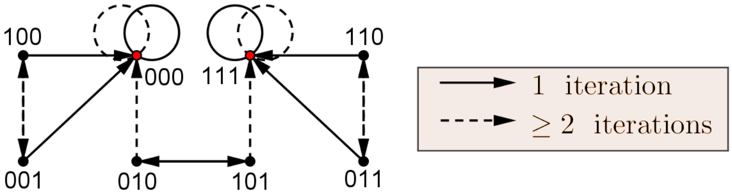
denoted $\mathcal{T}(\mathbf{x})$*, is a trapping set with an inducing set given by the variable nodes corresponding to* $\mathsf{supp}(\mathbf{x})$*. Given a trapping set* $\mathcal{T}$*, its set of inducing sets is given by the variable nodes corresponding to*

$$\{\mathsf{supp}(\mathbf{x}) : \mathcal{T}(\mathbf{x}) = \mathcal{T}\} \ ,$$

and its critical number is

$$m(\mathcal{T}) = \min\{\ |\mathsf{supp}(\mathbf{x})| : \mathcal{T}(\mathbf{x}) = \mathcal{T}\} \ .$$

Example



Trapping Sets in the Decoding Network

Corollary

If the decoding network, $\mathcal{D}$, of a code $\mathcal{C}$ is transitive, then the trapping sets are given by

- *the sets of variable nodes corresponding to supports of vertices with loops in $\mathcal{D}_1$, and*
- *the sets of variable nodes corresponding to unions of the supports of vertices forming directed cycles in $\mathcal{D}_1$.*

Trapping Sets in the Decoding Network

Corollary

If the decoding network, $\mathcal{D}$, of a code $\mathcal{C}$ is transitive, then the trapping sets are given by

- *the sets of variable nodes corresponding to supports of vertices with loops in $\mathcal{D}_1$, and*
- *the sets of variable nodes corresponding to unions of the supports of vertices forming directed cycles in $\mathcal{D}_1$.*

Furthermore, inducing sets of trapping sets in a transitive decoding network are given by the variable nodes corresponding to the support of any vertex which has a directed path to either a (nonzero) vertex with a loop, or to a directed cycle, regardless of where that path enters the cycle.

Decoding Diameter

Definition

Let $\mathcal{D} = (V, E, D)$ be the decoding network of a code $\mathcal{C}$ under a fixed decoder. For $\mathbf{x} \in V$, let $L_{\mathbf{x}}$ be the minimum nonnegative integer such that for all $\ell \geq L_{\mathbf{x}}$, $\mathbf{x}^{\ell}$, the output of the decoder after ℓ iterations, appears an infinite number of times in the sequence $\{\mathbf{x}^k\}_{k=L_{\mathbf{x}}}^{\infty}$. Then, the *decoding diameter* of $\mathcal{D}$ is given by $\Delta(\mathcal{D}) = \max_{\mathbf{x} \in V} L_{\mathbf{x}}$.

Decoding Diameter

Definition

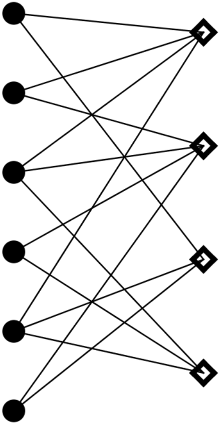
Let $\mathcal{D} = (V, E, D)$ be the decoding network of a code $\mathcal{C}$ under a fixed decoder. For $\mathbf{x} \in V$, let $L_{\mathbf{x}}$ be the minimum nonnegative integer such that for all $\ell \geq L_{\mathbf{x}}$, $\mathbf{x}^{\ell}$, the output of the decoder after ℓ iterations, appears an infinite number of times in the sequence $\{\mathbf{x}^k\}_{k=L_{\mathbf{x}}}^{\infty}$. Then, the *decoding diameter* of $\mathcal{D}$ is given by $\Delta(\mathcal{D}) = \max_{\mathbf{x} \in V} L_{\mathbf{x}}$.

Intuitively: the decoding diameter is the number of iterations required to fall into all trapping sets.

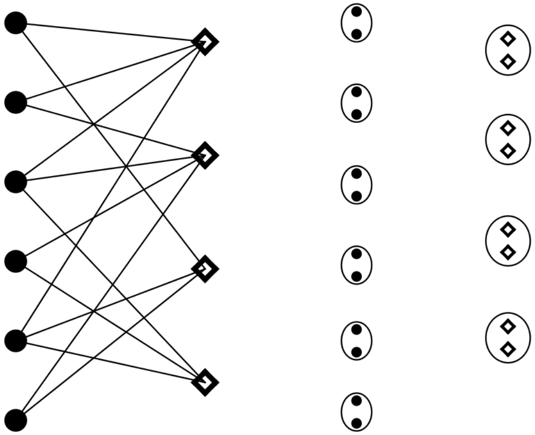
Applications

- Protograph codes
- Product codes
- Half-product codes

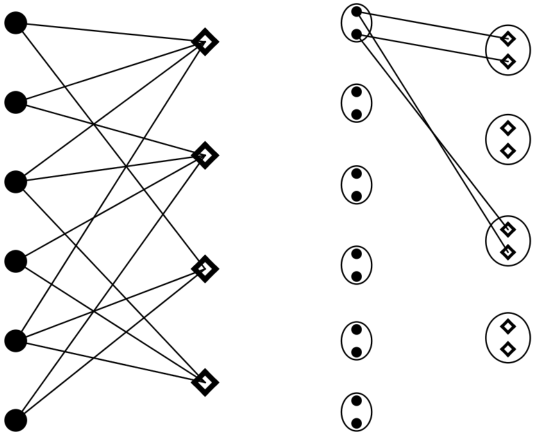
Protograph Codes



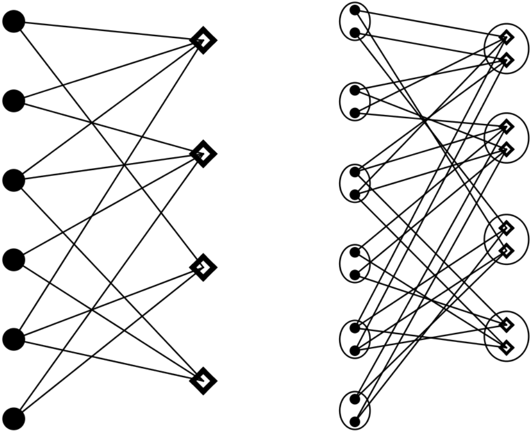
Protograph Codes



Protograph Codes



Protograph Codes



Protograph Codes

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Protograph Codes

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix}
\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}
\end{pmatrix}$$

Protograph Codes

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

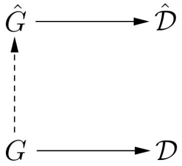
$$\begin{pmatrix} 1\,0 & 0\,1 & 0\,1 & 0\,0 & 1\,0 & 0\,0 \\ 0\,1 & 1\,0 & 1\,0 & 0\,0 & 0\,1 & 0\,0 \\ 0\,0 & 0\,1 & 1\,0 & 1\,0 & 0\,0 & 1\,0 \\ 0\,0 & 1\,0 & 0\,1 & 0\,1 & 0\,0 & 0\,1 \\ 0\,1 & 0\,0 & 0\,0 & 0\,0 & 1\,0 & 0\,1 \\ 1\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,1 & 1\,0 \\ 0\,0 & 0\,0 & 1\,0 & 0\,1 & 0\,1 & 0\,0 \\ 0\,0 & 0\,0 & 0\,1 & 1\,0 & 1\,0 & 0\,0 \end{pmatrix}$$

Protograph-Based LDPC Codes

- Let $\mathcal{C}$ be a binary linear code with Tanner graph G and decoding network $\mathcal{D}$ with respect to a fixed decoder.

Protograph-Based LDPC Codes

- Let $\mathcal{C}$ be a binary linear code with Tanner graph G and decoding network $\mathcal{D}$ with respect to a fixed decoder.
- Viewing G as a protograph, let $\hat{\mathcal{C}}$ be the code corresponding to a degree h lift of G, denoted $\hat{G}$, and (with an abuse of notation), let $\hat{\mathcal{D}}$ be the decoding network of $\hat{\mathcal{C}}$ with respect to the same decoder.



Protograph-Based LDPC Codes

Theorem

There exists a subgraph of $\hat{\mathcal{D}}$ that is isomorphic to $\mathcal{D}$.

In particular, if $\mathcal{D}$ is not transitive, then $\hat{\mathcal{D}}$ is not transitive (however, transitivity of $\mathcal{D}$ does not necessarily imply transitivity of $\hat{\mathcal{D}}$).

Product Codes

Definition

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be binary linear codes of lengths n and m, respectively. Then the *product code* $\mathcal{C}_1 \times \mathcal{C}_2$ is the set of $m \times n$ binary arrays such that each row forms a codeword in $\mathcal{C}_1$, and each column forms a codeword in $\mathcal{C}_2$.

Product Codes

Definition

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be binary linear codes of lengths n and m, respectively. Then the *product code* $\mathcal{C}_1 \times \mathcal{C}_2$ is the set of $m \times n$ binary arrays such that each row forms a codeword in $\mathcal{C}_1$, and each column forms a codeword in $\mathcal{C}_2$.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \\ a_{m1} & \cdots & & a_{mn} \end{pmatrix}$$

Product Codes

- Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be codes of lengths n and m, respectively, with decoding networks $\mathcal{D}^1$ and $\mathcal{D}^2$; consider $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$.

Product Codes

- Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be codes of lengths n and m, respectively, with decoding networks $\mathcal{D}^1$ and $\mathcal{D}^2$; consider $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$.

- Let the decoder that operate by iteratively decoding one component code at a time (i.e. all rows, and then all columns).

Product Codes

- Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be codes of lengths n and m, respectively, with decoding networks $\mathcal{D}^1$ and $\mathcal{D}^2$; consider $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$.
- Let the decoder that operate by iteratively decoding one component code at a time (i.e. all rows, and then all columns).
- A single decoding iteration of all rows (resp., all columns) is given by the directed graph product $(\mathcal{D}_1^1)^m$ (resp., $(\mathcal{D}_1^2)^n$). Notice these two networks can be considered as on the same vertex set.

Product Codes

- Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be codes of lengths n and m, respectively, with decoding networks $\mathcal{D}^1$ and $\mathcal{D}^2$; consider $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$.

- Let the decoder that operate by iteratively decoding one component code at a time (i.e. all rows, and then all columns).

- A single decoding iteration of all rows (resp., all columns) is given by the directed graph product $(\mathcal{D}^1_1)^m$ (resp., $(\mathcal{D}^2_1)^n$). Notice these two networks can be considered as on the same vertex set.

- Let A_1 be the adjacency matrix of $(\mathcal{D}^1_1)^m$, and let A_2 be the adjacency matrix of $(\mathcal{D}^2_1)^n$.

Product Codes

Theorem

The adjacency matrix of dimension ℓ of the decoding network, $\mathcal{D}$, of the product code $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$ is given by $(A_1 A_2)^\ell$.

Half-Product Codes

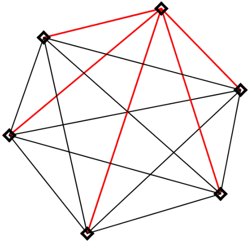
Definition (Justesen '11)

Let $\mathcal{C}$ be a binary linear code of length n, and let

$$\tilde{\mathcal{C}}_H = \{X - X^T : X \in \mathcal{C} \times \mathcal{C}\}.$$

The *half-product code* with component code $\mathcal{C}$, denoted $\mathcal{C}_H$, is obtained from $\tilde{\mathcal{C}}_H$ by setting the symbols below the diagonal of each element of $\tilde{\mathcal{C}}_H$ equal to zero.

$$\begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ 0 & 0 & a_{23} & a_{24} & a_{25} & a_{26} \\ 0 & 0 & 0 & a_{34} & a_{35} & a_{36} \\ 0 & 0 & 0 & 0 & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & 0 & a_{56} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



A half-product code with component code of length 6.

Half-Product Codes

- For each component code, consider the directed graph giving the decoding behavior of that component with a single iteration of the decoder, viewed within the vertex set of the decoding network of the entire half-product code (a disjoint union of $2^{(n-1)(n-2)/2}$ copies of dimension 1 of the decoding network of the component code).

Half-Product Codes

- For each component code, consider the directed graph giving the decoding behavior of that component with a single iteration of the decoder, viewed within the vertex set of the decoding network of the entire half-product code (a disjoint union of $2^{(n-1)(n-2)/2}$ copies of dimension 1 of the decoding network of the component code).

- Let A_i be the adjacency matrix of this directed graph for the ith component code.

Half-Product Codes

- For each component code, consider the directed graph giving the decoding behavior of that component with a single iteration of the decoder, viewed within the vertex set of the decoding network of the entire half-product code (a disjoint union of $2^{(n-1)(n-2)/2}$ copies of dimension 1 of the decoding network of the component code).
- Let A_i be the adjacency matrix of this directed graph for the ith component code.
- Let $\sigma \in S_n$ represent the permutation giving decoding order (i.e. $\sigma(1)$ gives the index of the first component code to be decoded).

Half-Product Codes

Theorem

The product $(A_{\sigma(1)} \cdots A_{\sigma(n)})^{\ell}$*, where* $\sigma \in S_n$*, gives the adjacency matrix of* $\mathcal{D}_{\ell}^{H}$*, dimension* ℓ *of the decoding network of the half-product code* $\mathcal{C}_H$*.*

Ongoing Work

- Determining the redundancy needed for decoding network transitivity for different families of codes
- Developing a Tanner graph condition that captures (non-)transitivity of the network
- Extending results to Gallager B and other decoders
- Analysis of decoding networks of classical families of codes (trapping sets, decoding diameter)

Thank You

References I

Koetter, R., Vontobel, P.: Graph-covers and iterative decoding of finite length codes. In: Proc. of the 3rd Intl Symp. on Turbo Codes (2003)

Dolecek, L., Zhang, Z., Anantharam, V., Wainwright, M., Nikolic, B.: Analysis of Absorbing Sets for Array-Based LDPC Codes. In: IEEE Intl Conf. on Comm., 6261–6268 (2007)

Richardson, T.: Error-floors of LDPC codes. In: Proc. of the 41st Allerton Conf on Communication, Control and Computing, 1426–1435 (2003)

Di, C., Proietti, D., Telatar, I.E., Richardson, T.J., Urbanke, R.L.: Finite-length analysis of low-density parity-check codes on the binary erasure channel. In: IEEE Trans. on Info. Thy. 48, 1570–1579 (2002)

Nguyen, D.V., Chilappagari, S.K., Marcellin, M.W., Vasic, B.: On the Construction of Structured LDPC Codes Free of Small Trapping Sets. In: IEEE Trans. Info. Thy, 2280–2302 (2012)

Sankaranarayanan, S., Chilappagari, S.K., Radhakrishnan, R., Vasić, B.: Failures of the Gallager B Decoder: Analysis and Applications. In: IEEE Intl Conf on Comm (2006)

Gallager, R.: Low-density parity-check codes. In: IRE Trans. on Info. Thy, 21-28 (1962)

Sun, J.: Studies on Graph-Based Coding Systems. Ph.D. Dissertation, Dept of Elec Eng, The Ohio State Univ, Columbus (2004)

Butler, B.K., Siegel, P.H.: Error Floor Approximation for LDPC Codes in the AWGN Channel. In: IEEE Trans. on Inf. Thy., 7416-7441 (2014)

Vasic, B., Chilappagari, S.K., Nguyen, D.V., Planjery, S.K.: Trapping set ontology. In: Proc. of the 47th Allerton Conf on Communication, Control, and Computing, 1–7 (2009)

Berlingerio, M., Coscia, M., Giannotti, F., Monreale, A., Pedreschi, D.: Foundations of Multidimensional Network Analysis. In: Intl Conf on Adv. in Soc. Networks Analysis and Mining, 485–489 (2011)

References II

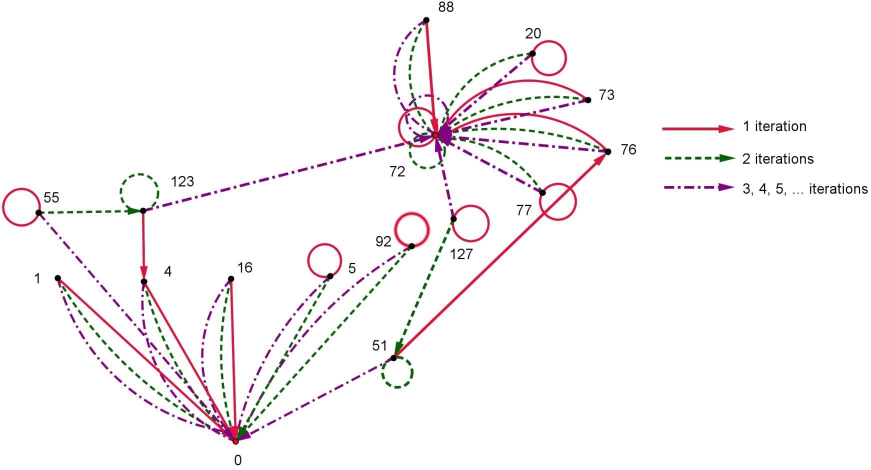
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd ed. MIT Press (2009)
- Beemer, A., Kelley, C.A.: Avoiding Trapping Sets in SC-LDPC Codes under Windowed Decoding. In: Proc. IEEE Intl Symp. Info. Thy. and Apps. (2016)
- Karimi, M., Banihashemi, A.H.: Efficient Algorithm for Finding Dominant Trapping Sets of LDPC Codes. In: IEEE Trans. on Inf. Thy. 58, 6942–6958 (2012)
- Hashemi, Y., Banihashemi, A.H.: Characterization and Efficient Exhaustive Search Algorithm for Elementary Trapping Sets of Irregular LDPC codes. In: Proc. IEEE Intl Symp. Info. Thy. (2017)
- Justesen, J.: Performance of Product Codes and Related Structures with Iterated Decoding. In: IEEE Trans. on Comm. 59, 407–415 (2011)
- Mittelholzer, T., Parnell, T., Papandreou, N., Pozidis, H.: Improving the Error-Floor Performance of Binary Half-Product Codes. In: Proc. IEEE Intl Symp. Info. Thy. and Apps. (2016)
- Pfister, H.D., Emmadi, S.K., Narayanan, K.: Symmetric product codes. In: Info. Thy. and App. Workshop, 282–290 (2015)

Decoding Network Example

Consider the blocklength 7 code given by the parity-check matrix H, shown below, and decoded using Gallager A decoding.

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

Decoding Network Example



Other "connected" components are given by adding codewords to the binary representations of the vertex labels in the figure. Remaining words never decode to codewords.