LDPC Codes and Message-Passing Decoders: An Introductory Survey

Valentin Savin

CEA-LETI, MINATEC Campus, 17 rue des Martyrs, 38054 Grenoble, France

E-mail: valentin.savin@cea.fr

Abstract

The outstanding success of Low Density Parity Check (LDPC) codes in providing practical constructions that closely approach the theoretical Shannon limit is rooted in the way they are decoded. They feature iterative message-passing decoders able to convey information between coded bits, so that to progressively improve the estimation of the sent codeword. This tutorial provides first an overall survey of LDPC decoders, and then a more detailed insight into some of the most widely used decoders. We also discuss the asymptotic analysis of these decoders and explain how this analysis made possible the optimization of LDPC codes operating very close to the Shannon limit.

Key words: LDPC codes, Iterative decoders, Message-Passing, Belief-Propagation, Sum-Product, Min-Sum.

1 Introduction

It is widely recognized that one of the most significant contributions to coding theory is the invention of Low-Density Parity-Check (LDPC) codes by Gallager in the early 60's [1]. Yet, rather than a family of codes, Gallager invented a new method of decoding linear codes, by using iterative message-passing (MP) algorithms. Such a decoding algorithm consists of an exchange of messages between coded bits and parity checks they participate in. Each message provides an estimation of either the sender or the recipient coded bit, and the exchange of messages takes place in several rounds, or iterations. At each iteration, new messages are computed in an *extrinsic manner*, meaning that the message received by a coded bit from a parity-check (or vice versa) does not depend on the message just sent the other way around. Consequently, coded bits collect more and more information with each new decoding iteration, which gradually improves the estimation of the sent codeword.

Even if LDPC codes came equipped with a class of MP decoding algorithms, a substantial effort had to be made in order to advance our knowledge on iterative decoding techniques. Most of the research on decoding algorithms focused on connections with closely-related areas and the design of practical MP decoders [2]. It worth mentioning here one of the most celebrated works, namely the work of Tanner [3], who described LDPC codes in terms of sparse bipartite graphs and proposed a more general construction of graph-based linear codes. He also generalized the decoding algorithms proposed by Gallager to this new class of graph-based codes, and gave a unified treatment of decoding algorithms for LDPC and product codes.

The capability of MP decoding algorithms to deal with long block lengths opened the way to Shannon limit. They led to the development of graph-based codes and belief-propagation decoding, closely related to the probabilistic approach to coding devised by Shannon. A detailed survey that traces the evolution of channel coding from Hamming codes to capacity-approaching codes can be found in [4]. It is worth noting that unlike the classical coding approach, in which codes are considered and optimized on an individual basis, in the context of probabilistic coding the goal is to find a family of codes that optimizes the average performance under a given MP decoding algorithm. A decisive contribution was made by Richardson and Urbanke [5], who derived a general method for determining the correction capacity of LDPC codes under MP decoding algorithms. They introduced new ensembles of LDPC codes and showed that in the asymptotic limit of the block length, almost all codes (in the same ensemble) behave alike and exhibit a threshold phenomenon, separating the region where reliable transmission is possible from that where it is not. This made possible the design of *irregular* LDPC codes that perform very close to the Shannon limit [6]. Nowadays, LDPC codes are known to be capacity approaching codes for a wide range of channel models, which motivated the increased interest of the scientific community over the last 15 years and supported the rapid transfer of this technology to the industrial sector.

Ackowledgment

This work was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i-RISC project).

References

[1] R. G. Gallager, “Low density parity check codes,” MIT Press, Cambridge, 1963, Research Monograph series.

[2] V. Savin, “LDPC decoders”, in *Channel coding: Theory, algorithms, and applications*, D. Declercq, M. Fossorier, and E. Biglieri editors, Academic Press Library in Mobile and Wireless Communications, Elsevier, 2014.

[2] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. on Inf. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[3] G.D. Forney and D.J. Costello, “Channel coding: The road to channel capacity,” *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

[4] T.J. Richardson and R.L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[5] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

LDPC Codes and Message-Passing Decoders: An Introductory Survey

TINKOS Conference, Niš-Serbia, June 16, 2014

Valentin Savin

CEA-LETI, MINATEC Campus, France

Part of the research reported in this presentation was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (**i-RISC** project)

MINATEC CAMPUS

Outline

- Coding for noisy channels: from Shannon to Shannon
  - Linear codes and Shannon’s Theorem
  - Iterative message passing decoders and LDPC codes
  - Approaching the Shannon limit

- Coding for noisy channels with noisy devices
  - Noisy message-passing decoders
  - Impact of the “computation noise” on the error correction performance

Coding for noisy channels: Shannon's theory

Noisy channel: $X \longrightarrow \boxed{\begin{array}{c}\textbf{Channel}\\ p(y\,|\,x)\end{array}} \longrightarrow Y$

- Add *redundancy* to X to allow correcting transmission errors
- Redundancy decreases the *information rate*: fraction between number of source (information) symbols and number of transmitted symbols

- Shannon's theorem (1948)

  Tightest upper bound on the rate of information that can be reliably transmitted over the channel, known as *channel capacity*, is given by:

$$C = \max_{p_X} I(X, Y)$$

- Practical constructions that closely approach the Shannon limit
  - LDPC codes & **MP decoders** (Gallager 1962)
  - Analysis & optimization (Richardson et. al 2001)

Coding for noisy channels: Shannon's theory

Noisy channel: $X \longrightarrow \boxed{\textbf{Channel} \; p(y \mid x)} \longrightarrow Y$

- The information is transmitted in the form of *codewords*, belonging to a *codebook* (*the code*) known by both TX and RX

- *Error detection:* received word does not belong to the codebook
- *Error correction:* find the codeword closest to the received word

Linear Codes

- Codewords: binary vectors satisfying a system of linear equations

$$\begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

- $X = (x_1, x_2, \ldots, x_{10})$ such that $H \cdot X^T = 0$
  - $x_1 + x_2 + x_4 + x_7 = 0$

    ….
  - $x_7 + x_8 + x_9 + x_{10} = 0$

Linear Codes

- *Error Detection:*

from channel →

$$
\begin{array}{cccccccccc}
\mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1}
\end{array}
$$

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

- $X = (x_1, x_2, \ldots, x_{10})$ such that $H \cdot X^T = 0$
  - $x_1 + x_2 + x_4 + x_7 = 0$

    ....
  - $x_7 + x_8 + x_9 + x_{10} = 0$

Linear Codes

- *Error Detection:*

from channel → $\begin{matrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{matrix}$

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

✓ check#1 satisfied

- $X = (x_1, x_2, \ldots, x_{10})$ such that $H \cdot X^T = 0$
  - $x_1 + x_2 + x_4 + x_7 = 0$

    ….
  - $x_7 + x_8 + x_9 + x_{10} = 0$

Linear Codes

- *Error Detection:*

from channel

$$
\begin{array}{cccccccccc}
0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1
\end{array}
$$

$$
\begin{pmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix}
\begin{array}{l}
= 0 \\
= 0 \\
\\
\\
\\
\end{array}
$$

✓ check#1 satisfied

✓ check#2 satisfied

- $X = (x_1, x_2, \dots, x_{10})$ such that $H \cdot X^T = 0$
  - $x_1 + x_2 + x_4 + x_7 = 0$

    ....
  - $x_7 + x_8 + x_9 + x_{10} = 0$

Linear Codes

- *Error Detection:*

$$\xrightarrow{\text{from channel}} \quad 0 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 1$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{array}{l} = 0 \\ = 0 \\ = 1 \\ \\ \\ \end{array}$$

✓ check#1 satisfied

✓ check#2 satisfied

✘ check#3 violated

$\Rightarrow$ not a codeword

- $X = (x_1, x_2, \ldots, x_{10})$ such that $H \cdot X^T = 0$
  - $x_1 + x_2 + x_4 + x_7 = 0$

    ....
  - $x_7 + x_8 + x_9 + x_{10} = 0$

Linear Codes

- *Error Correction:*
  - Find the closest codeword

$$\begin{array}{cccccccccc} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & \color{red}{1} & 1 \end{array}$$

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

  - How to do it in general?
    - Large codes (thousands of bits)
    - Many errors

Linear Codes

- *Error Correction:*
  - Find the closest codeword

$$\begin{array}{cccccccccc} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}$$

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

  - Gallager (1962)
    - Iterative exchange of information between coded-bits and parity-check equations

Bipartite Graph Representation

- *Error Correction:*
  - Find the closest codeword

$$
\begin{array}{c c}
 & \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{array} \\
\begin{array}{c} c_1: \\ c_2: \\ c_3: \\ c_4: \\ c_5: \end{array} &
\begin{pmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix}
\end{array}
$$

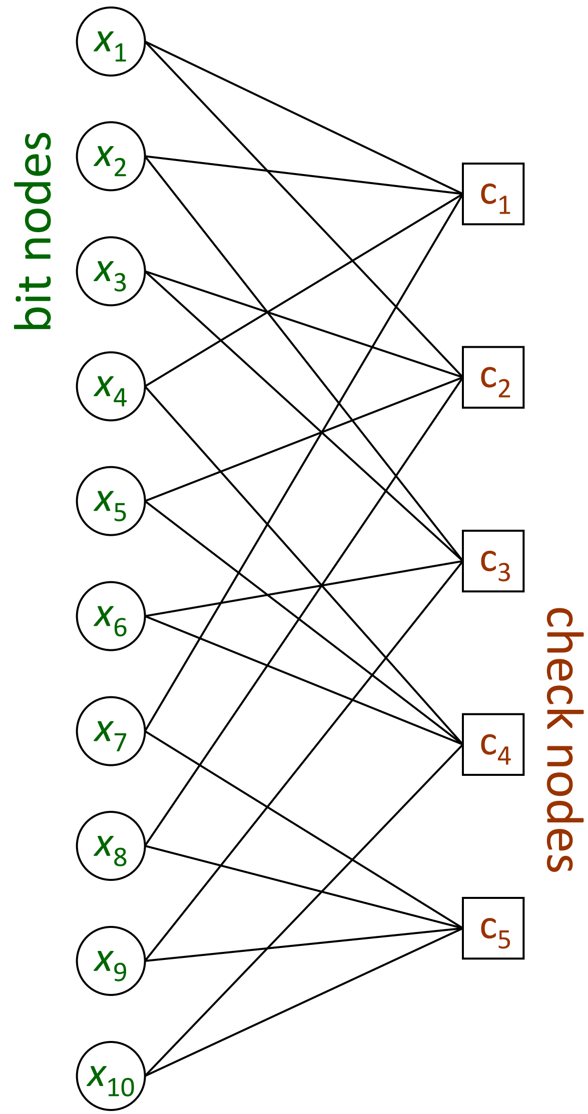
  - Gallager (1962)
    - Iterative exchange of information between coded-bits and parity-check equations
  - Tanner (1981): bipartite graph representation

Bipartite Graph Representation

- *Error Correction:*
  - Find the closest codeword

$$
\begin{array}{r c}
 & \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{array} \\
\begin{array}{r} c_1: \\ c_2: \\ c_3: \\ c_4: \\ c_5: \end{array} & \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}
\end{array}
$$

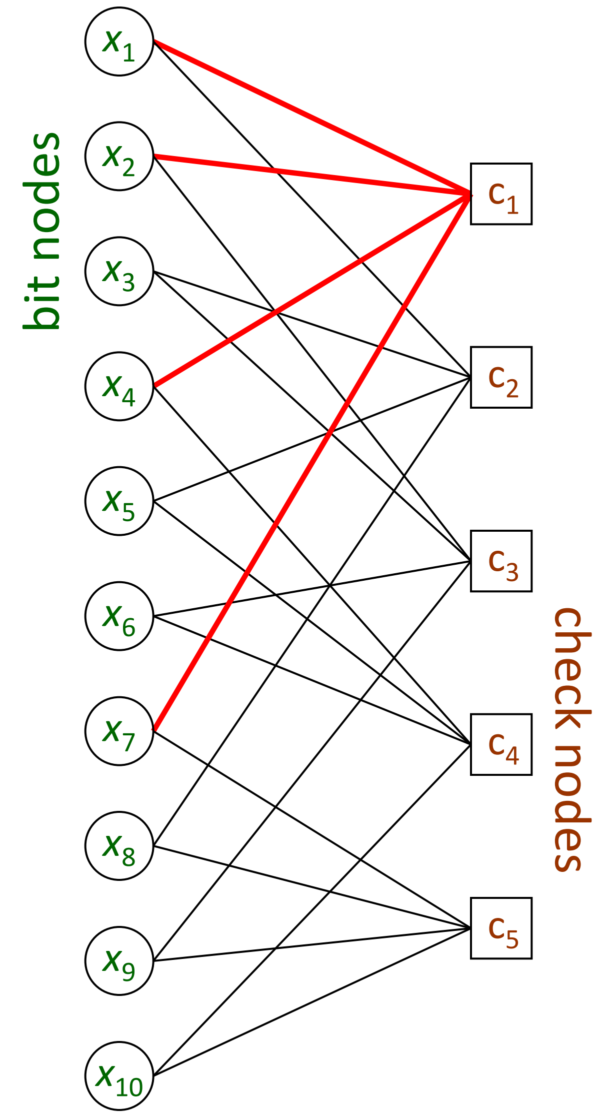
  - Gallager (1962)
    - Iterative exchange of information between coded-bits and parity-check equations
  - Tanner (1981): bipartite graph representation



Bipartite Graph Representation

- *Error Correction:*
  - Find the closest codeword

$$
\begin{array}{c c}
 & \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{array} \\
\begin{array}{c} c_1: \\ c_2: \\ c_3: \\ c_4: \\ c_5: \end{array} & \left(\begin{array}{cccccccccc}
\boxed{1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0} \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}\right)
\end{array}
$$

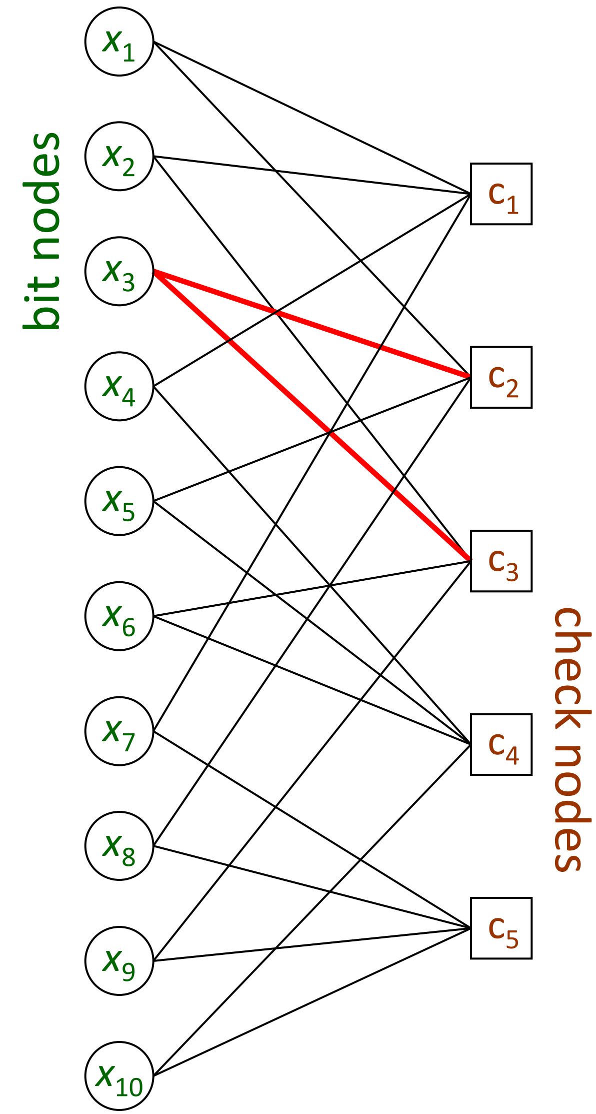


  - Gallager (1962)
    - Iterative exchange of information between coded-bits and parity-check equations
  - Tanner (1981): bipartite graph representation

Bipartite Graph Representation

- *Error Correction:*
  - Find the closest codeword

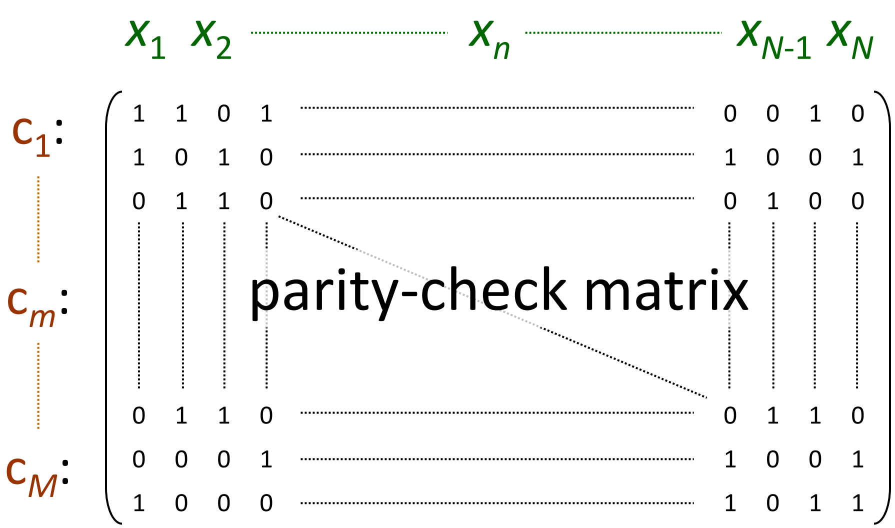
$$
\begin{array}{r|cccccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\
\hline
c_1: & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
c_2: & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
c_3: & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
c_4: & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
c_5: & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}
$$



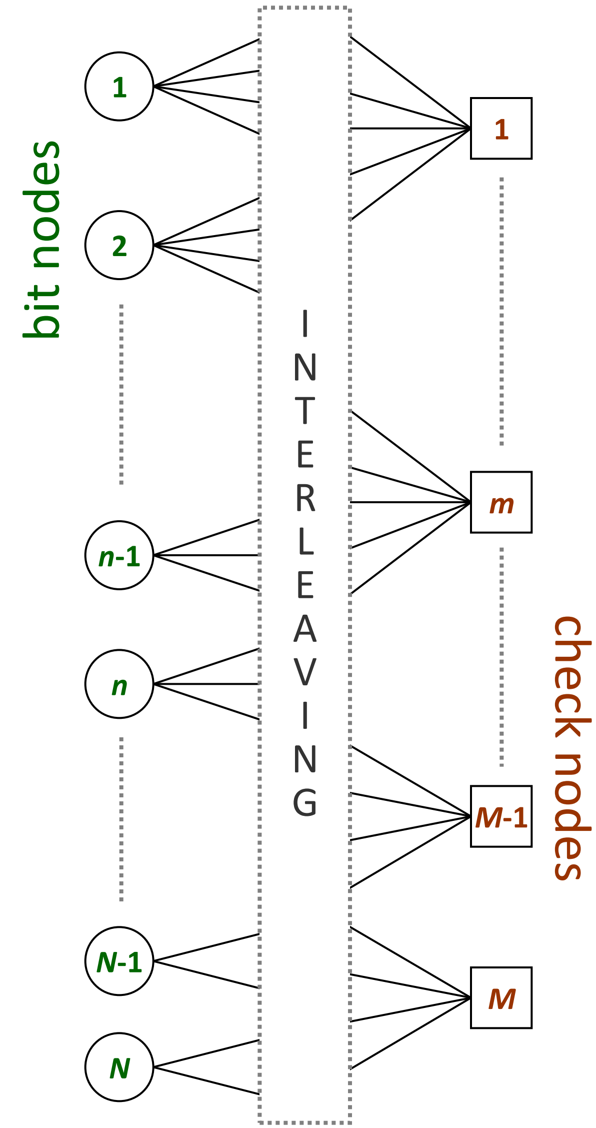
  - Gallager (1962)
    - Iterative exchange of information between coded-bits and parity-check equations
  - Tanner (1981): bipartite graph representation

Bipartite Graph Representation

- *Error Correction:*
  - Find the closest codeword



- Gallager (1962)
  - Iterative exchange of information between coded-bits and parity-check equations
- Tanner (1981): bipartite graph representation



Outline

- Coding for noisy channels: from Shannon to Shannon
  - Linear codes and Shannon's Theorem
  - Iterative message passing decoders and LDPC codes
  - Approaching the Shannon limit

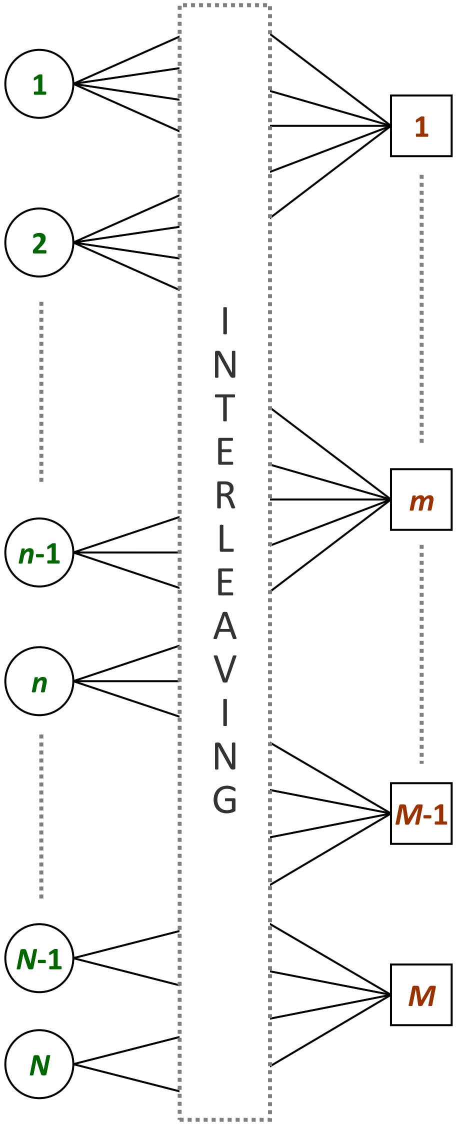
- Coding for noisy channels with noisy devices
  - Noisy message-passing decoders
  - Impact of the "computation noise" on the error correction performance

Message Passing Decoders

The principle

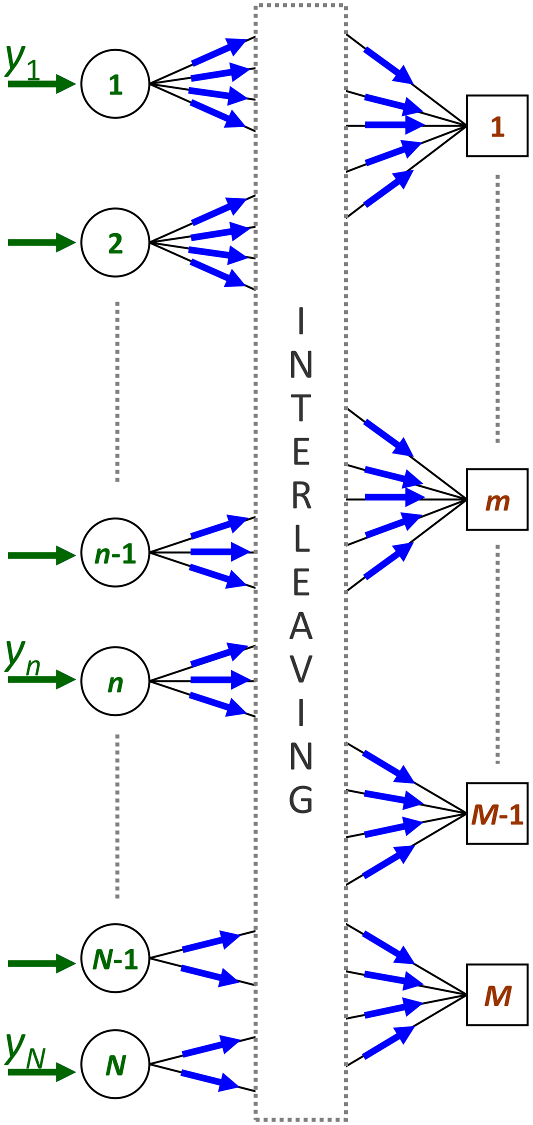
- *Exchange of messages* between bit and check nodes
  - Each message provides an estimation of either the sender or the recipient bit-node

- Exchange of messages takes place in several rounds, or *iterations*
  - Bit-nodes collect more and more information with each new iteration, which gradually improves the estimation of the sent codeword



Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges

Majority Voting Decoding

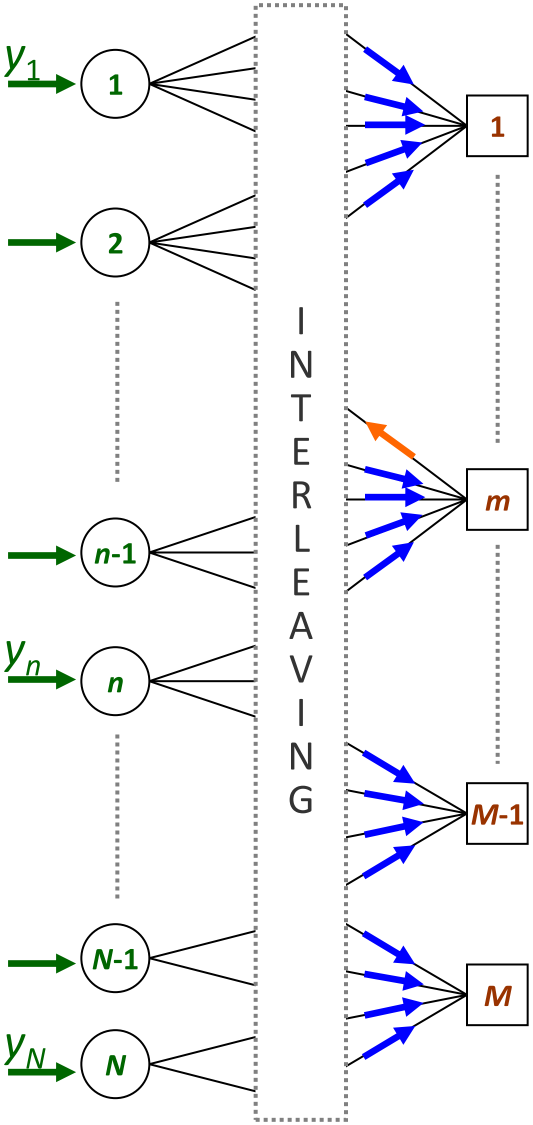
- Decoder is fed with the sequence of bit values (y_n) received from the channel

Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

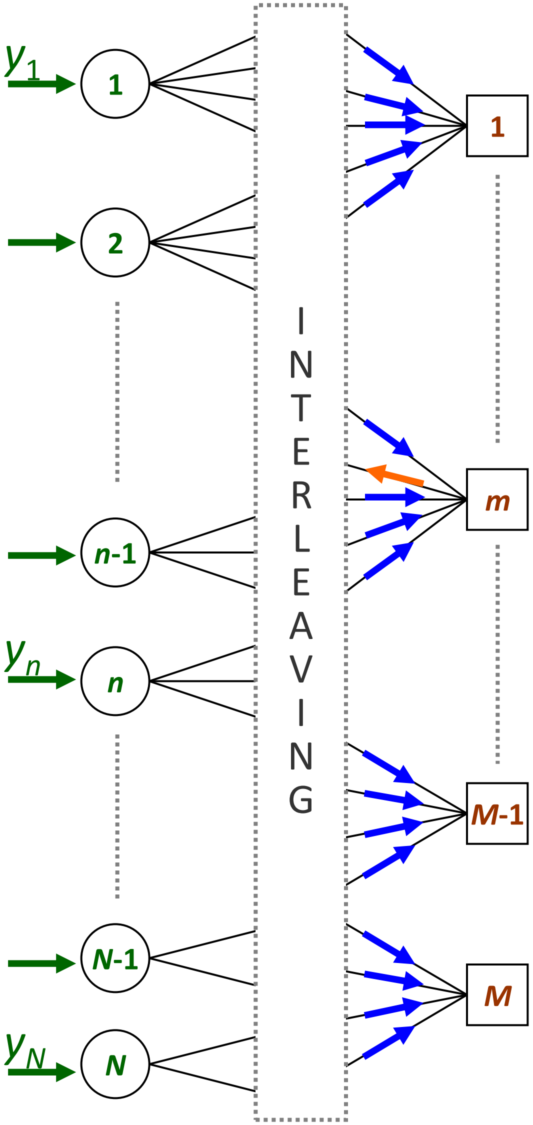
Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges



Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

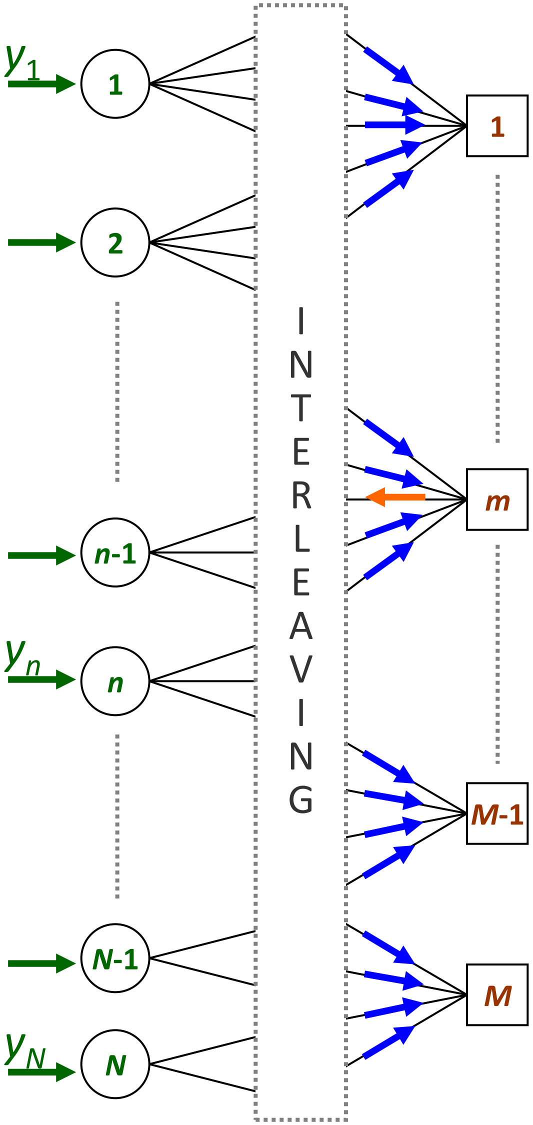
- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

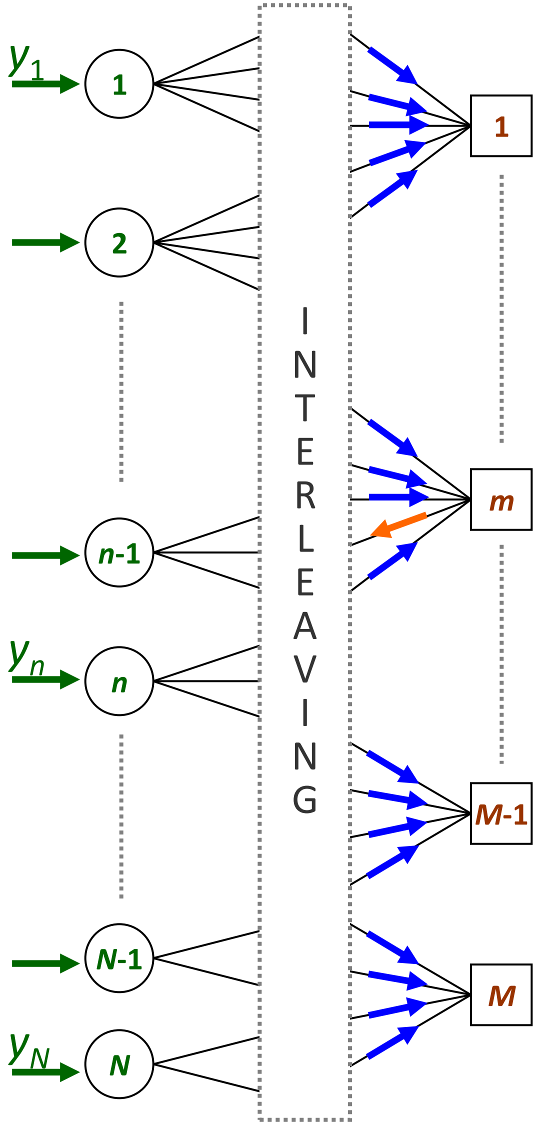
- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

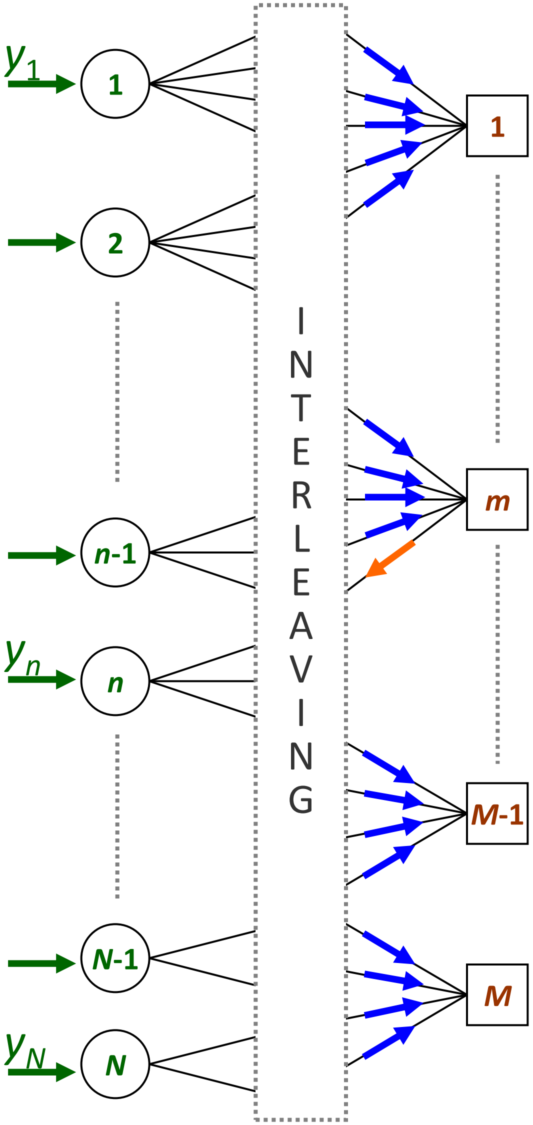
- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel

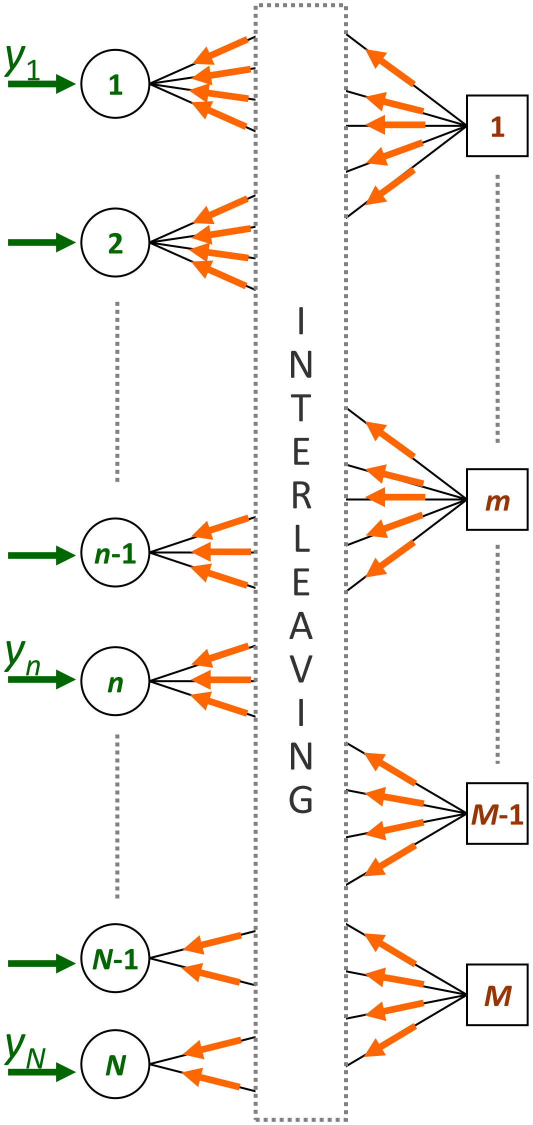
Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

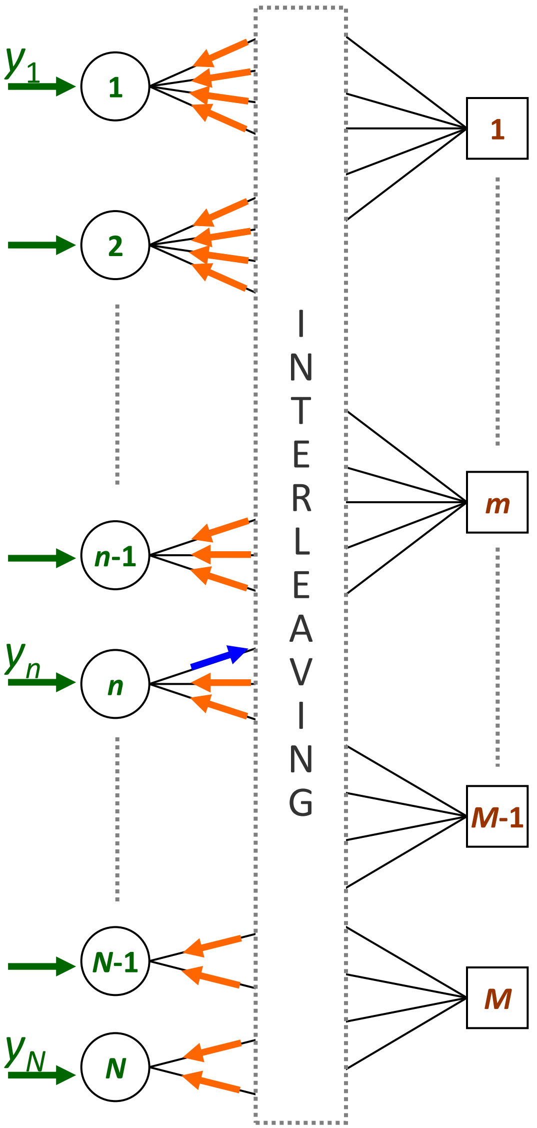
- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges
- Bit-to-check node messages
  - outgoing message = **majority value** among channel output and incoming messages on the other edges

    (NB: different messages may be sent on different edges!)



Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

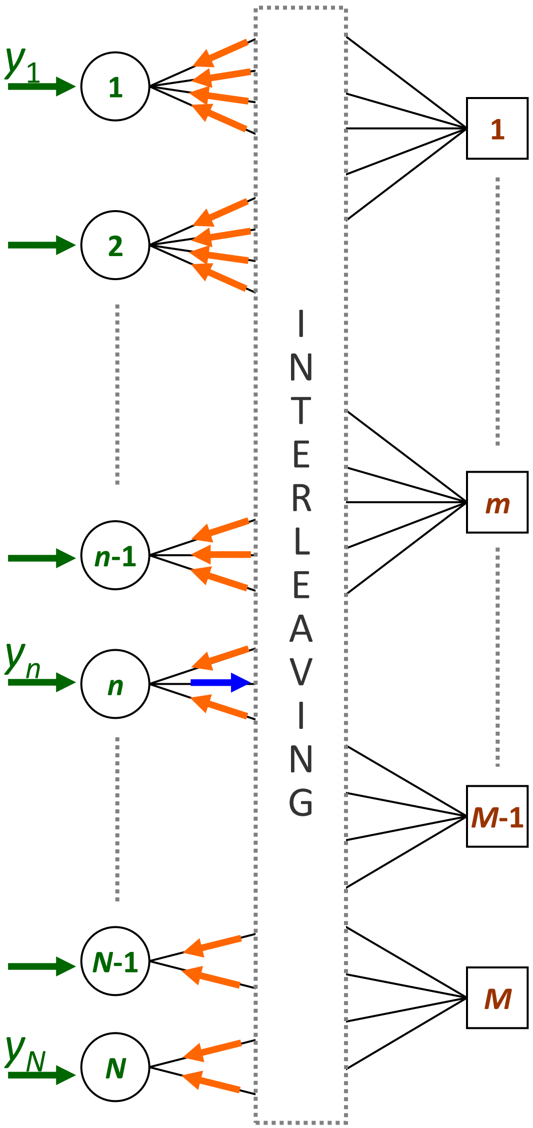
Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges
- Bit-to-check node messages
  - outgoing message = **majority value** among channel output and incoming messages on the other edges

    (NB: different messages may be sent on different edges!)

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

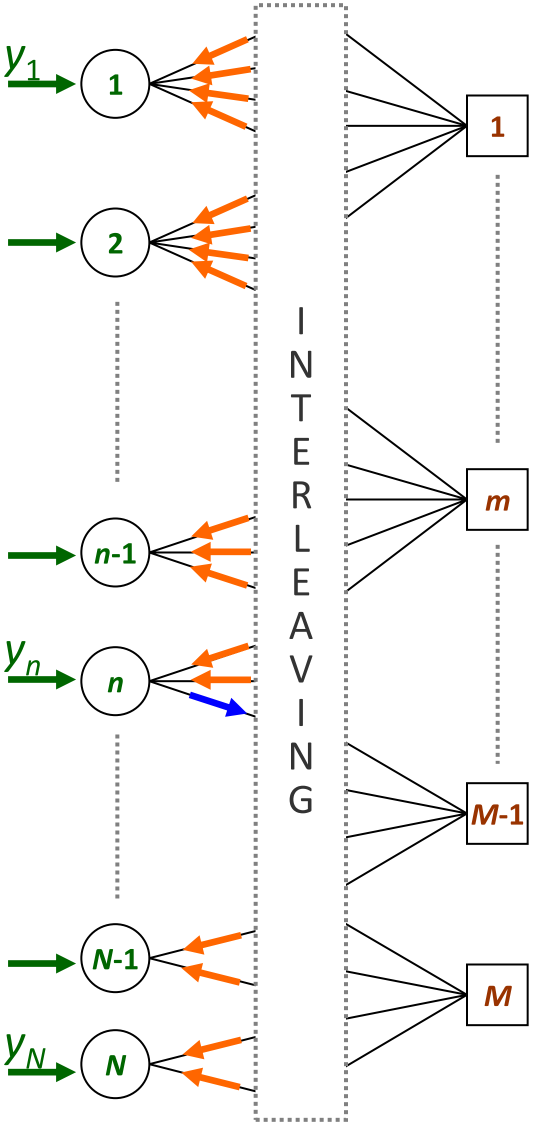
Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges
- Bit-to-check node messages
  - outgoing message = **majority value** among channel output and incoming messages on the other edges

    (NB: different messages may be sent on different edges!)

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

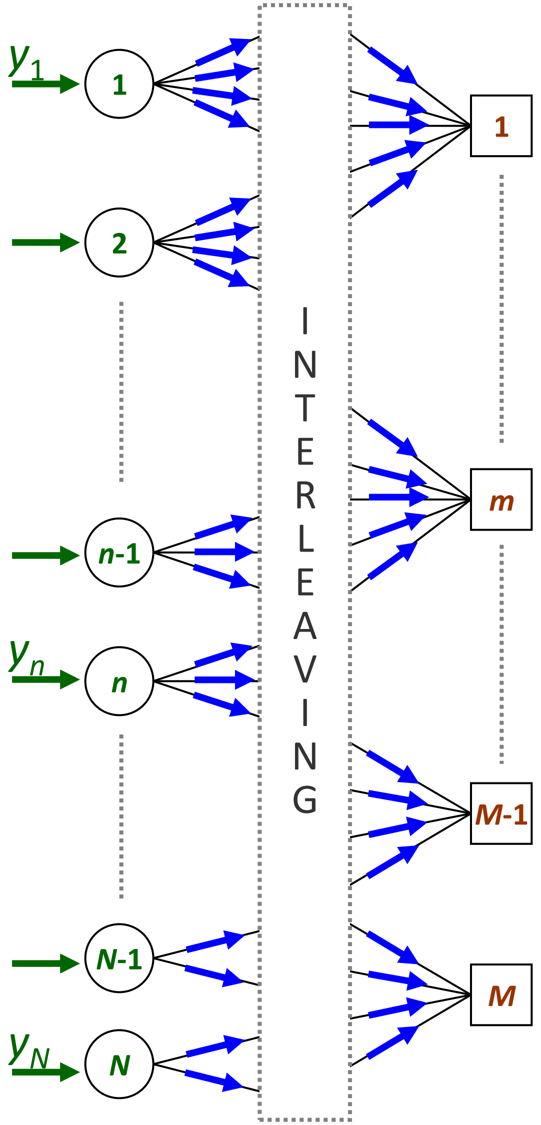
Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges
- Bit-to-check node messages
  - outgoing message = **majority value** among channel output and incoming messages on the other edges

    (NB: different messages may be sent on different edges!)

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel



Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges
- Bit-to-check node messages
  - outgoing message = **majority value** among channel output and incoming messages on the other edges

    (NB: different messages may be sent on different edges!)

Majority Voting Decoding

- Decoder is fed with the sequence of bit values (y_n) received from the channel
  - $\Rightarrow$ hard decision must be taken for soft-output channels
  - $\Rightarrow$ suboptimal

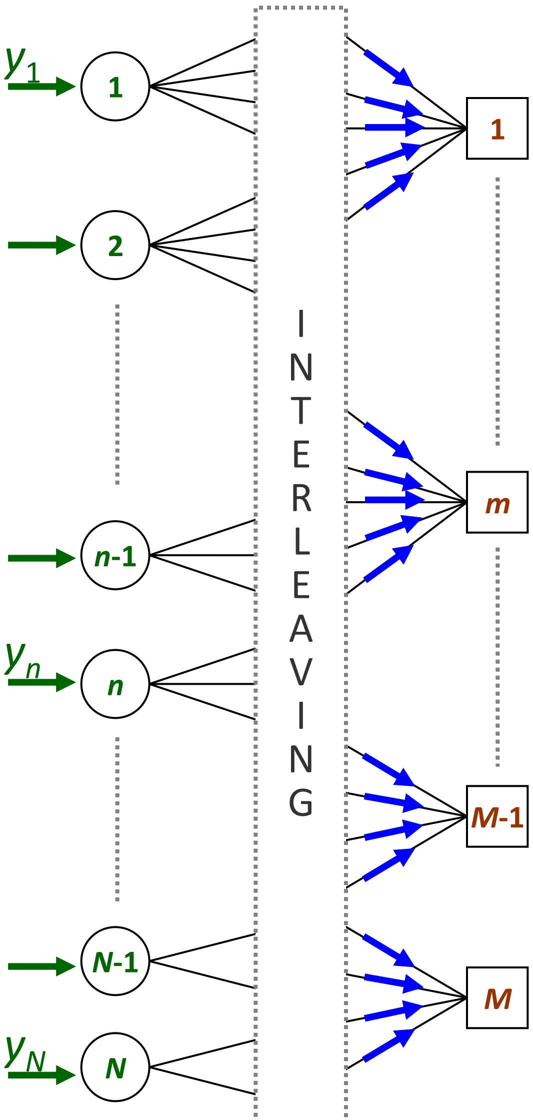
Initialization

- Iterative exchange is initialized by bit-nodes:
  - each bit-node sends its received value to the neighbor check-nodes

Iterations

- Check-to-bit node messages
  - outgoing message = **XOR** of incoming messages received on the other edges
- Bit-to-check node messages
  - outgoing message = **majority value** among channel output and incoming messages on the other edges

    (NB: different messages may be sent on different edges!)



Belief-Propagation Decoding

- Decoder is fed with LLR values
  - $\gamma_n = \mathrm{LLR}(x_n \mid y_n)$
- Exchanged messages (α, β) are also LLR values

Initialization

- Iterative exchange is initialized by bit-nodes:

$$\alpha_{m,n} = \gamma_n$$

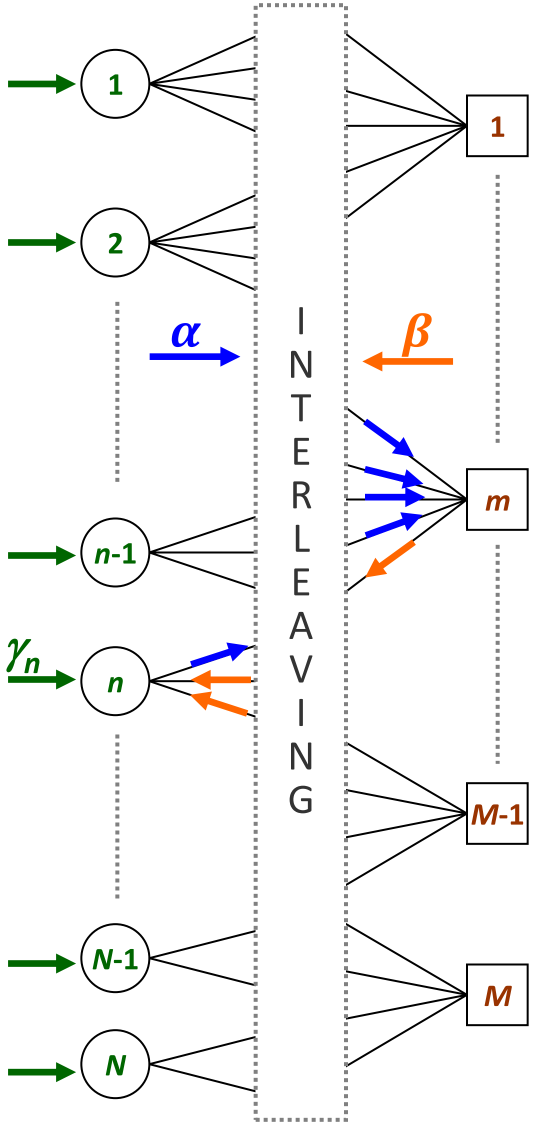
Iterations

- Check-to-bit node messages

$$\beta_{m,n} = \mathrm{LLR}\left(x_n \,\middle|\, \alpha_{m,n'} : n' \in H(m)\backslash n\right)$$

- Bit-to-check node messages

$$\alpha_{m,n} = \mathrm{LLR}\left(x_n \,\middle|\, \gamma_n \text{ and } \beta_{m',n} : m' \in H(n)\backslash m\right)$$



Belief-Propagation Decoding

- Decoder is fed with LLR values
  - $\gamma_n = \mathrm{LLR}(x_n \mid y_n)$
- Exchanged messages (α, β) are also LLR values

Initialization

- Iterative exchange is initialized by bit-nodes:

$$\alpha_{m,n} = \gamma_n$$

Iterations

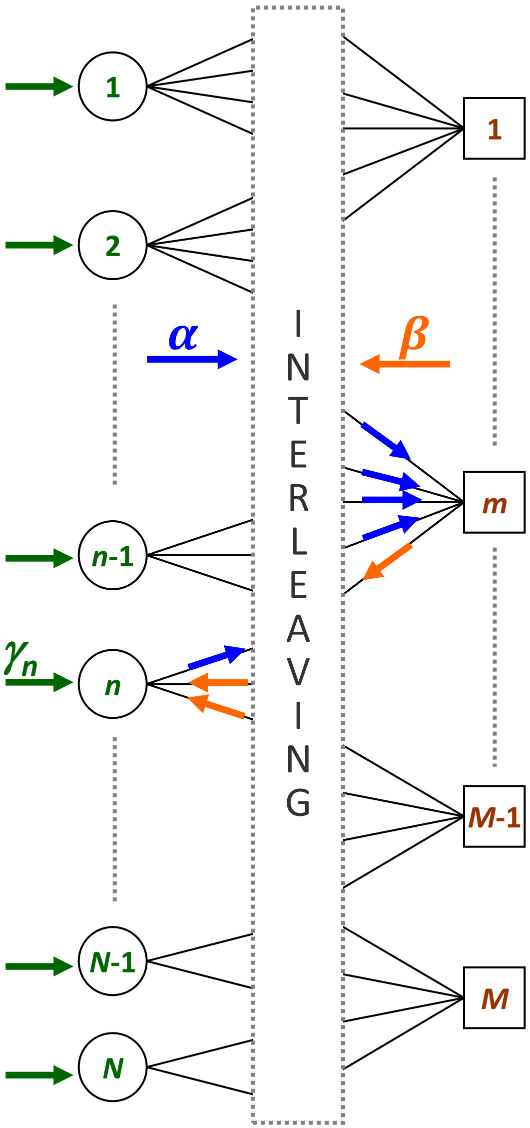
- Check-to-bit node messages

$$\beta_{m,n} = \left(\prod_{n' \in H(m)\setminus n} \mathrm{sgn}(\alpha_{m,n'})\right) \phi\left(\sum_{n' \in H(m)\setminus n} \phi(|\alpha_{m,n'}|)\right)$$

where $\phi(x) = \log\left(\frac{e^x+1}{e^x-1}\right)$

- Bit-to-check node messages

$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n)\setminus m} \beta_{m',n}$$



Min-Sum Decoding

- Decoder is fed with LLR values
  - $\gamma_n = \mathrm{LLR}(x_n \mid y_n)$
- Exchanged messages (α, β) are also LLR values

Initialization

- Iterative exchange is initialized by bit-nodes:

$$\alpha_{m,n} = \gamma_n$$

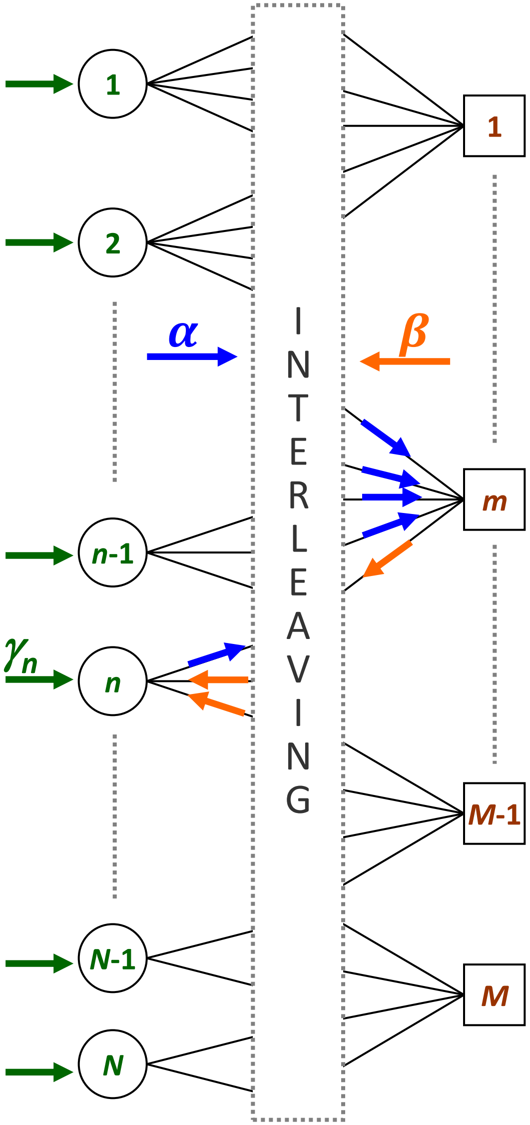
Iterations

- Check-to-bit node messages

$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \mathrm{sgn}(\alpha_{m,n'}) \right) \min_{n' \in H(m) \setminus n} (|\alpha_{m,n'}|)$$

- Bit-to-check node messages

$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n) \setminus m} \beta_{m',n}$$

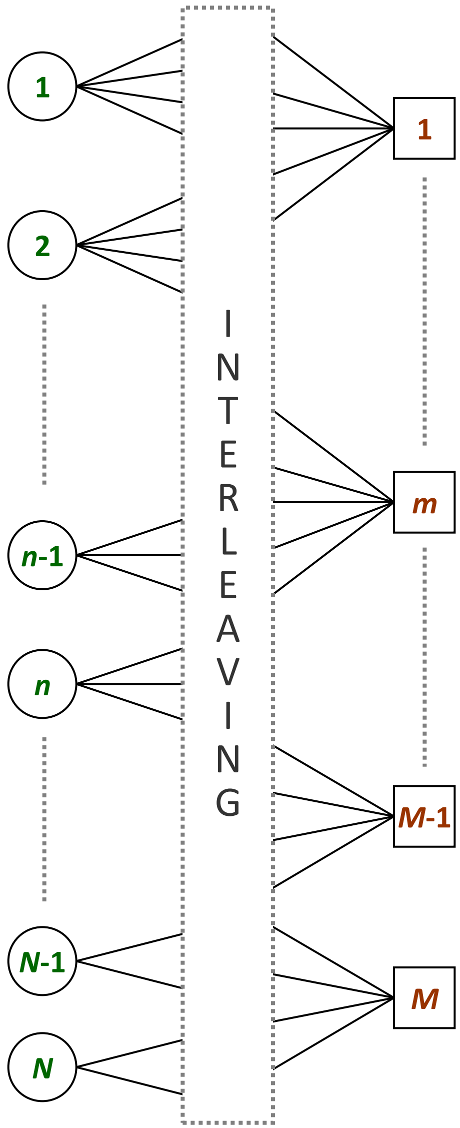


Message Passing Decoders

- **Majority Voting** decoding
  - A particular case of the Gallager B decoding (1962)

- **Belief-Propagation (Sum-Product)** decoding
  - Gallager's probabilistic decoding (1962)
  - Belief-Propagation: MP algorithm proposed by J. Pearl (1982) to perform Bayesian inference on trees, but also successfully used on general graphical models
  - "Optimal" for codes defined by cycle-free bipartite graphs, in the sense that it outputs the MAP estimates of the coded bits

- **Min-Sum** decoding
  - An approximate version of the Belief-Propagation
  - Generalization of the Viterbi algorithm, from trellises to more general graphical models
  - For codes defined by cycle-free bipartite graphs, MS decoding outputs the ML estimate of the sent codeword

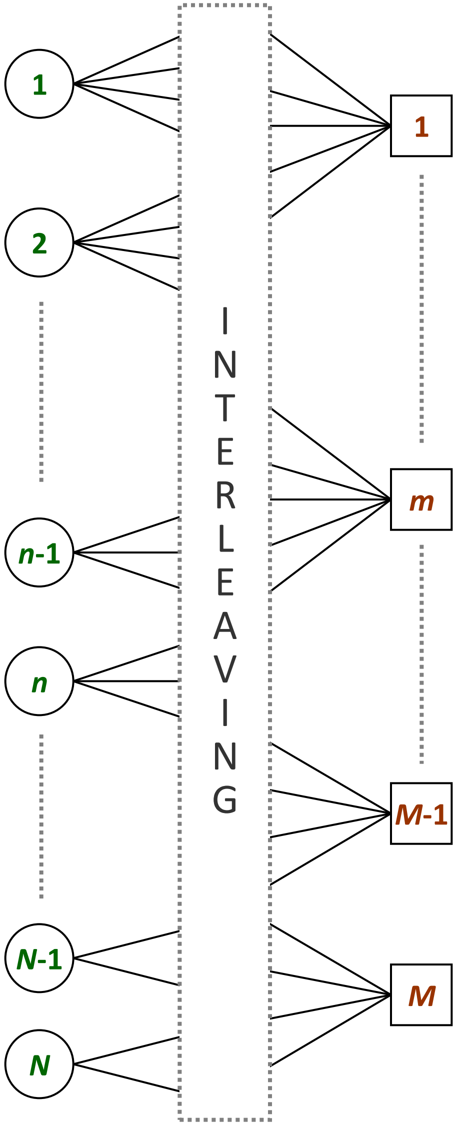


Message Passing Decoders

- **Min-Sum-based** decoders
  - improved versions of the MS algorithm, with only a very limited (usually negligible) increase in complexity
  - “correction” methods to mitigate the performance penalty of MS with respect to BP
  - Normalized MS, Offset MS, Self-Corrected MS

- **Stochastic** decoding
  - Stochastic implementation of the BP

- **Erasure** decoding
  - BP ⇔ MS ⇔ Peeling decoding



Effectiveness of MP decoders

- Codes defined by cycle-free graphs
  - BP = MAP $\Rightarrow$ optimal in terms of “bit error rate”
  - MS = ML $\Rightarrow$ optimal in terms of “word error rate”

- But practical codes are defined by graphs with cycles
  - Cycles may lead to “self-confirmations” in the decoding process
  - Self-confirmations should only occur after the exchanged messages have been sufficiently strengthened by the iterative process

- Avoid short cycles

  $\Rightarrow$ graph must be sparse $\Leftrightarrow$ parity-check matrix is low density

  $\Rightarrow$ **Low Density Parity Check (LDPC) codes**

- **LDPC:** necessary condition, but not sufficient

A Revolutionary Approach to Coding

- Rather than a family of codes, Gallager invented a new method of decoding linear codes, by using iterative MP algorithms
  - LDPC : *necessary condition* for a linear code to be effectively decoded by MP algorithms
  - Completely new and revolutionary approach to coding in the early 60's
    - the classical approach was to construct first a family of codes, and then find a practical decoding algorithm capable to correct any number of errors up to the designed correction capacity (half the minimum distance)
    - no need to invent a decoding algorithm for LDPC codes: they came equipped with iterative MP algorithms.
  - MP algorithms:
    - linear complexity (due to the sparsity of the matrix)
    - allow the use of long codes: indispensable condition to approach the channel capacity

Outline

- Coding for noisy channels: from Shannon to Shannon
  - Linear codes and Shannon's Theorem
  - Iterative message passing decoders and LDPC codes
  - Approaching the Shannon limit

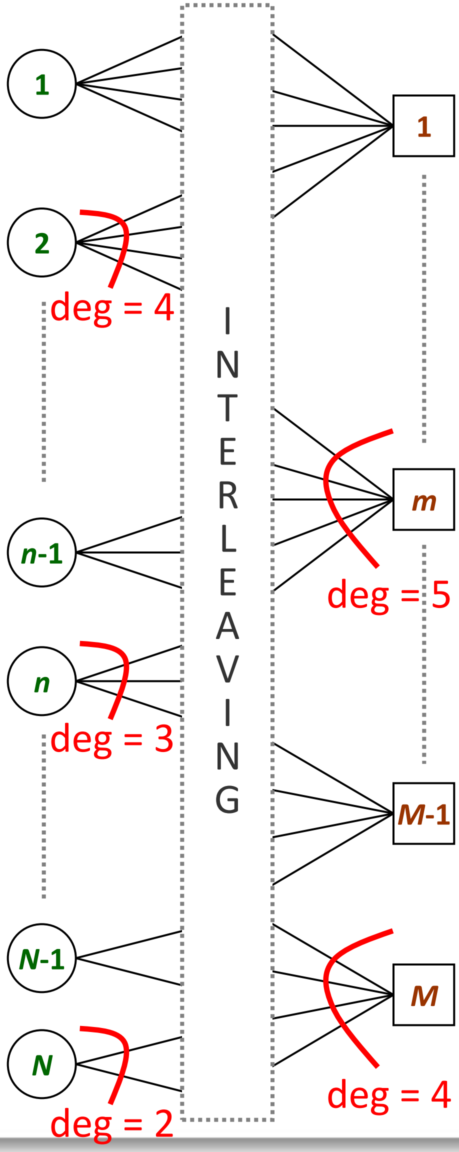
- Coding for noisy channels with noisy devices
  - Noisy message-passing decoders
  - Impact of the "computation noise" on the error correction performance

Error Correction Capacity of MP Decoders

- Richardson et al. (2001)
  - Density Evolution $\Rightarrow$ asymptotic performance
  - Asymptotically, the error correction capacity depends only on the "irregularity profile"

$$\boxed{\lambda(x) = \sum_d \lambda_d x^{d-1}\ , \quad \rho(x) = \sum_d \rho_d x^{d-1}}$$

  - λ_d = fraction of edges incident to deg-*d* bit-nodes
  - ρ_d = fraction of edges incident to deg-*d* check-nodes
  - Threshold phenomenon
    - Threshold value separating the region where reliable decoding is possible from where it is not
      - $p < p_{\text{TH}} \Rightarrow$ successful decoding
      - $p > p_{\text{TH}} \Rightarrow$ unsuccessful decoding
    - Optimization: find (λ, ρ) s.t p_{TH} is close to capacity



Asymptotic analysis of MP decoders

Density evolution

- Recursive relation between the distribution of messages exchanged at iteration ℓ and the distribution at iteration ℓ+1
  - *Easy case:* binary-alphabet decoders (e.g. MV) – messages' distribution is defined by only one probability value
  - *Difficult case:* continuous-alphabet decoders (BP, MS)
  - *In between:* finite-alphabet decoders (e.g. quantized decoders) – messages' distribution is a probability mass function on a finite number of values
- This recursion allows computing the error probability $\boldsymbol{p}_\ell$ at iteration ℓ
- Taking the limit as $\ell \to \infty$, one can determine whether the decoding is successful ($\boldsymbol{p}_\ell \to 0$) or not

Assumption

- independent messages $\Leftrightarrow$ cycle-free graph $\Leftrightarrow$ code length (N) goes to infinity

Density evolution for MV decoding over BSC

- $\mathcal{C}(d_v, d_c)$ – ensemble of (d_v, d_c)-regular LDPC codes
- p_ℓ = error probability at the ℓ^{th} iteration of the MV decoding (probability that a bit-to-check message at iteration ℓ is in error)
- p_0 = crossover probability of the BSC channel

$$p_\ell = p_0 - p_0 \sum_{k=b}^{d_v-1} \binom{d_v-1}{k} \cdot \left(\frac{1 + (1-2p_{\ell-1})^{d_c-1}}{2} \right)^k \cdot \left(\frac{1 - (1-2p_{\ell-1})^{d_c-1}}{2} \right)^{d_v-1-k} +$$

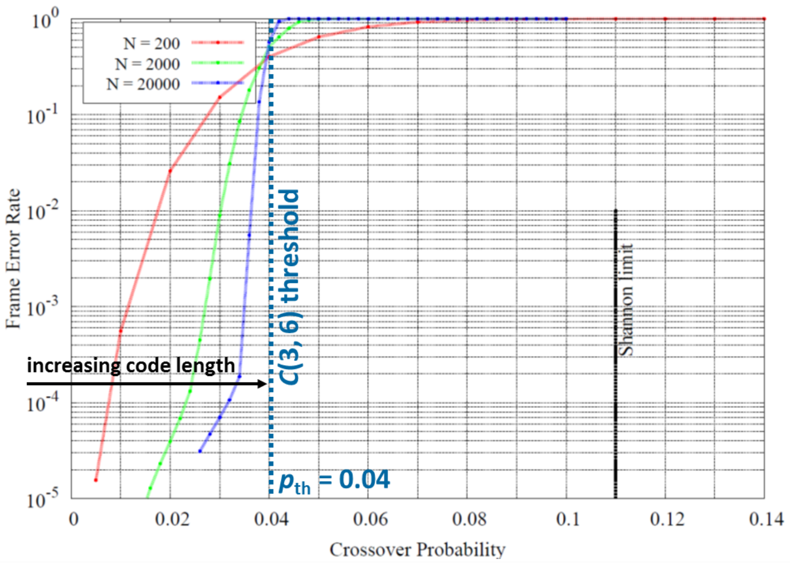
$$(1-p_0) \sum_{k=b}^{d_v-1} \binom{d_v-1}{k} \cdot \left(\frac{1 - (1-2p_{\ell-1})^{d_c-1}}{2} \right)^k \cdot \left(\frac{1 + (1-2p_{\ell-1})^{d_c-1}}{2} \right)^{d_v-1-k} \qquad b = \left\lfloor \frac{d_v+1}{2} \right\rfloor$$

- Threshold value: $p_{TH} = \sup \{ p_0 \mid \lim_{\ell \to \infty} p_\ell = 0 \}$
  - Worst channel condition that allows successful decoding
    (assuming that both code length and number of iterations go to infinity)

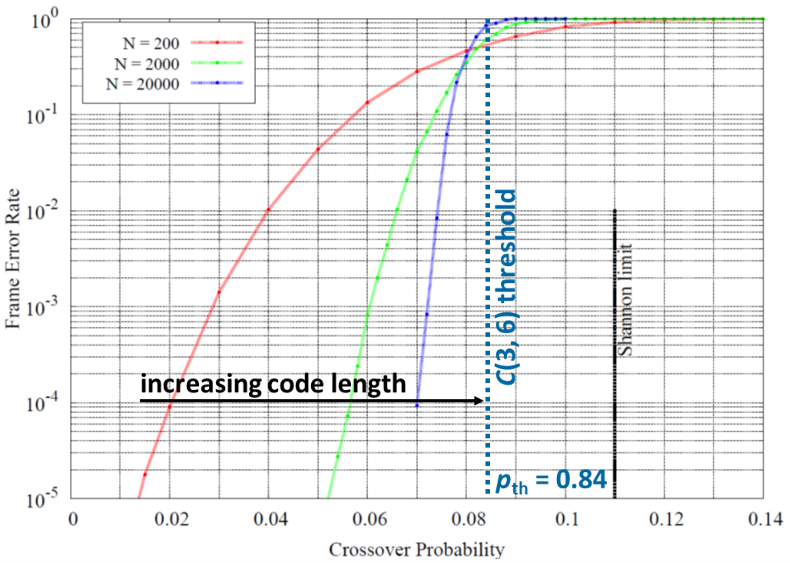
Density evolution for MV decoding over BSC

d_v	d_c	Rate	p_{TH} – MV	p_{TH} – BP	capacity
3	6	0.5	0.040	0.084	0.11
4	8	0.5	0.051	0.076	0.11
5	10	0.5	0.041	0.068	0.11

Regular (3, 6) LDPC code, MV decoding

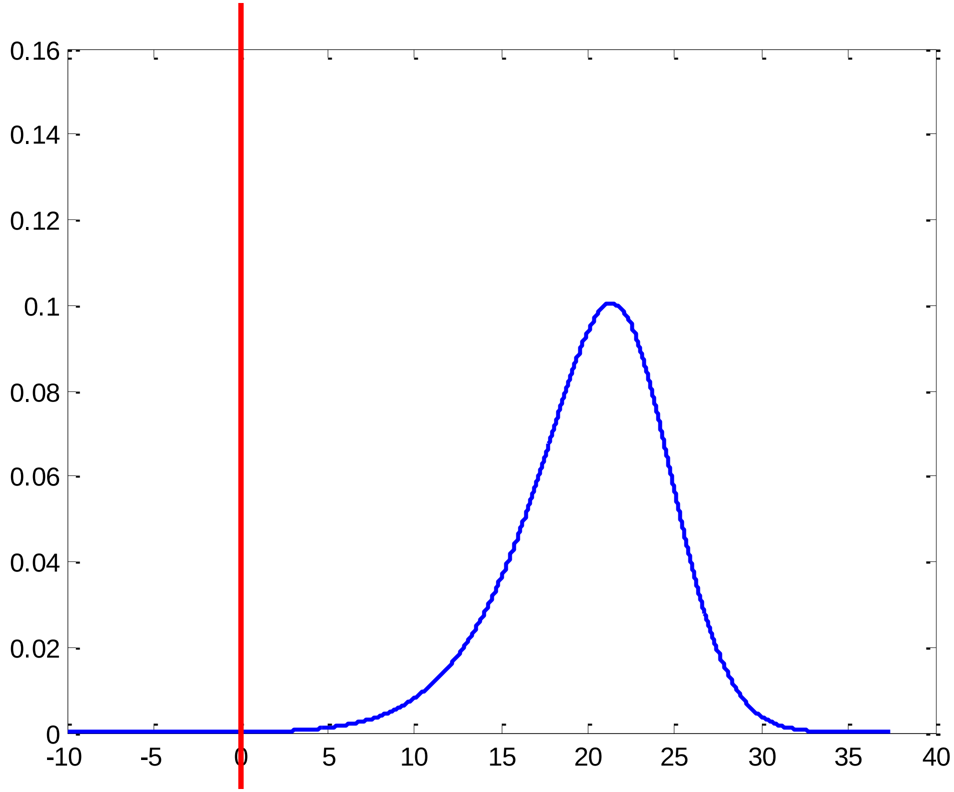


Regular (3, 6) LDPC code, BP decoding



Density evolution for BP over the BI-AWGN

- Recursive relation: $PDF_{l+1} = f(PDF_l)$, where PDF = probability density function of bit-to-check node messages
- Iteration number: l = 11



regular (3,6)-LDPC

σ = 0.8

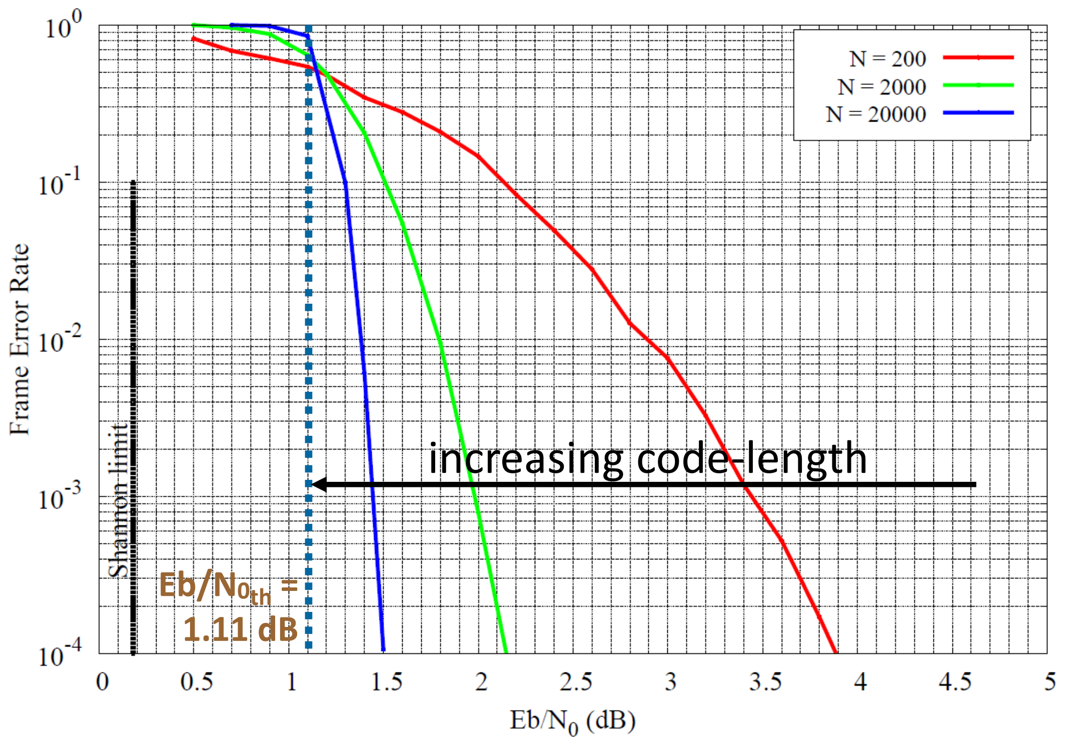
[E_b/N_0 = 1.94 dB]

(N.B: **σ_{TH} = 0.88**)

- Gaussian approximation: $PDF_l \approx N(m_l, 2m_l) \Rightarrow m_{l+1} = f(m_l)$

Density evolution for BP over the BI-AWGN

d_v	d_c	Rate	σ_{th} – SP	[Eb/N_0 dB]	capacity	
3	6	0.5	0.88	[1.11 dB]	0.98	[0.18 dB]
4	8	0.5	0.83	[1.62 dB]	0.98	[0.18 dB]
5	10	0.5	0.79	[2.05 dB]	0.98	[0.18 dB]



Asymptotic optimization of LDPC codes

- The density evolution technique can be applied to irregular codes
- It allows determining a threshold value that depends only on the degree distribution polynomials (λ, ρ)

Optimization

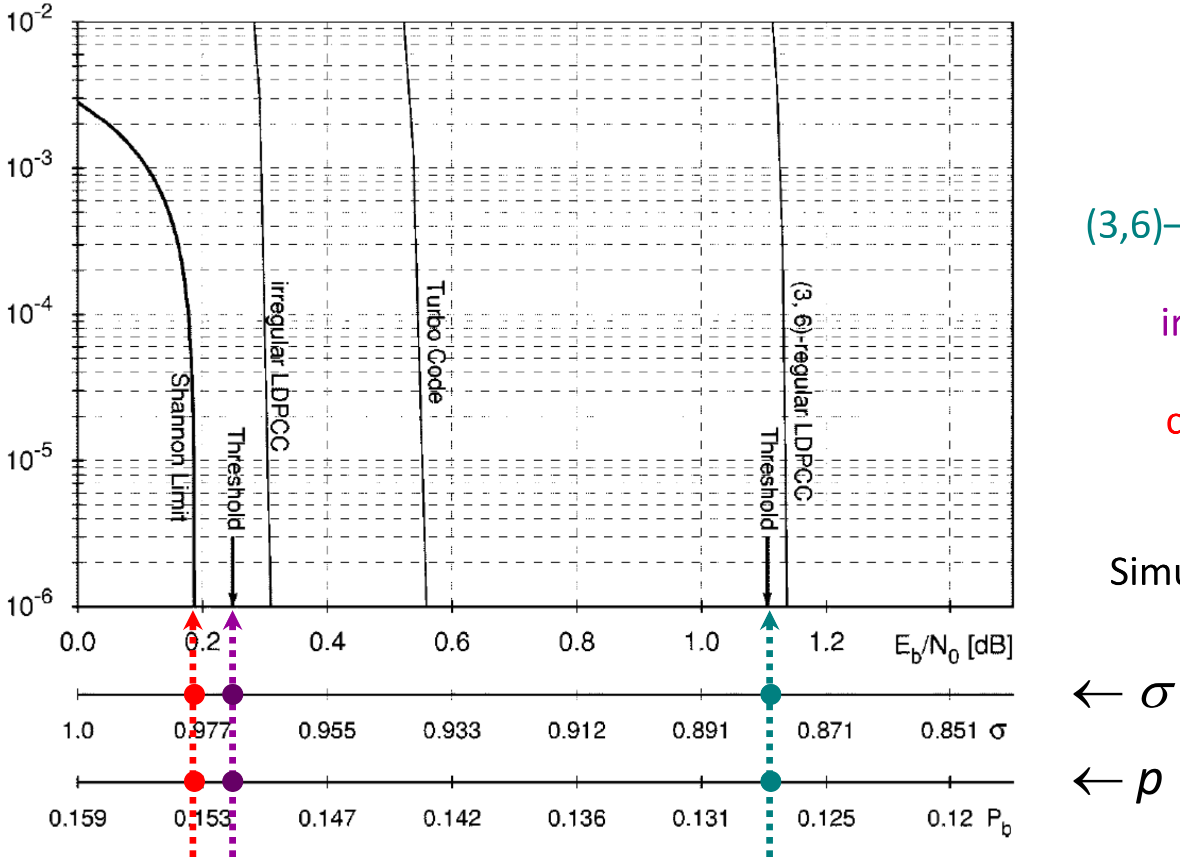
- Find (λ, ρ) that maximize $p_{TH}(\lambda, \rho)$
  - Hopefully, $p_{TH}(\lambda, \rho)$ is close to the channel capacity ☺
  - Linear optimization, genetic algorithms

- Irregular LDPC code over the BI-AWGN (rate = 1/2)
  - $\lambda(X) = 0.17120\,X + 0.21053\,X^2 + 0.00273\,X^3 + 0.00009\,X^6 + 0.15269\,X^7 + 0.09227\,X^8 + 0.02802\,X^9 + 0.01206\,X^{14} + 0.07212\,X^{29} + 0.25830\,X^{49}$
  - $\rho(X) = 0.33620\,X^8 + 0.08883\,X^9 + 0.57497\,X^{10}$

$\sigma_{th} = 0.98 \leftrightarrow Eb/N0_{th} = 0.26$ dB (gap to capacity $\Delta = 0.08$ dB!)

Asymptotic optimization of LDPC codes

Irregular LDPC codes over AWGN channel



(3,6)–regular: $\sigma_{th} = 0.88$ [$p_{th} = 0.128$]

irregular: $\sigma_{th} = 0.97$ [$p_{th} = 0.151$]

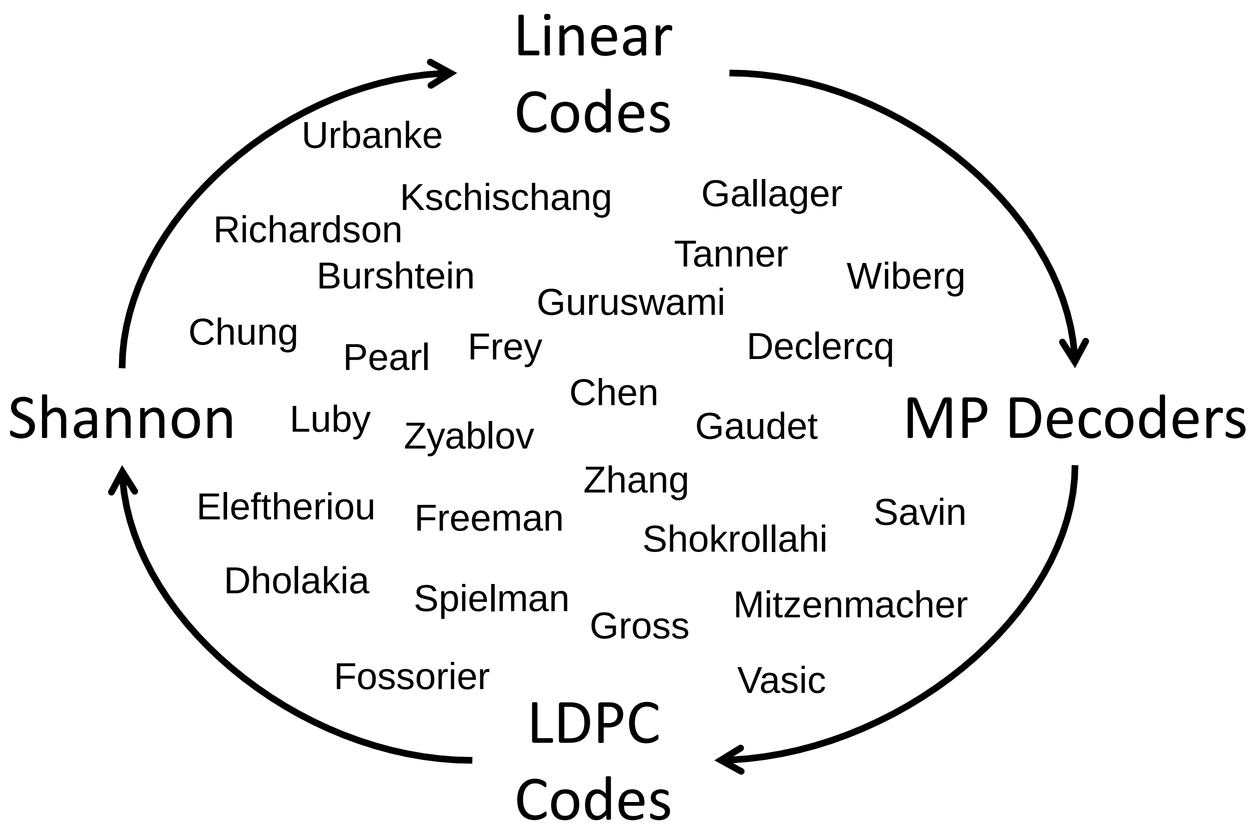
capacity: $\sigma_{Sh} = 0.98$ [$p_{Sh} = 0.153$]

Simulated codes are of length $N = 10^6$

$\lambda(X) = 0.17120X + 0.21053X^2 + 0.00273X^3 + 0.00009X^6 + 0.15269X^7 + 0.09227X^8 + 0.02802X^9 + 0.01206X^{14} + 0.07212X^{29} + 0.25830X^{49}$

$\rho(X) = 0.33620X^8 + 0.08883X^9 + 0.57497X^{10}$

Conclusion (from Shannon to Shannon)



References

- LDPC codes [1]
- Belief Propagation: [1-6]
- Min-Sum: [7-9]
- Min-Sum-based: [10-15]
- Stochastic decoding: [16-19]
- Decoding over erasure channels: [19-23]
- Asymptotic analysis and optimization: [24-25]
- Surveys and introductory readings: [26-30]

References

1. R. G. Gallager, “Low density parity check codes,” MIT Press, Cambridge, 1963, Research Monograph series
2. N. Wiberg, Codes and decoding on general graphs, Ph.D. thesis, Likoping University, Sweden, 1996
3. J. Pearl, “Reverend Bayes on inference engines: A distributed hierarchical approach,” in Proc. of the 2nd National Conference on Artificial Intelligence (AAAI-82), 1982, pp. 133–136
4. Frank R. Kschischang and Brendan J. Frey, “Iterative decoding of compound codes by probability propagation in graphical models,” IEEE Journal on Selected Areas in Communications, vol. 16, no. 2, pp. 219–230, 1998.
5. F.R. Kschischang, B.J. Frey, and H.A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Trans. on Information Theory, vol. 47, no. 2, pp. 498–519, 2001.
6. Yedidia, W.T. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations,” Exploring artificial intelligence in the new millennium, vol. 8, pp. 236–239, 2003
7. M.P.C. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation,” IEEE Trans. on Communications, vol. 47, no. 5, pp. 673–680, 1999.
8. S.Y. Chung, On the construction of some capacity-approaching coding schemes, Ph.D. thesis, Massachusetts Institute of Technology, 2000.
9. E. Eleftheriou, T. Mittelholzer, and A. Dholakia, “Reduced-complexity decoding algorithm for low-density parity-check codes,” IET Electronics Letters, vol. 37, no. 2, pp. 102–104, 2001.
10. J. Chen and M. P. Fossorier, “Near optimum universal belief propagation based decoding of low density parity check codes,” IEEE Trans. on Communications, vol. 50, no. 3, pp. 406–414, 2002.
11. J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X.Y. Hu, “Reduced-complexity decoding of LDPC codes,” IEEE Trans. on Communications, vol. 53, no. 8, pp. 1288–1299, 2005.
12. J. Chen, R.M. Tanner, C. Jones, and Y. Li, “Improved min-sum decoding algorithms for irregular LDPC codes,” in IEEE Int. Symp. on Inf. Theory (ISIT), 2005, pp. 449–453.

References

13. J. Zhang, M. Fossorier, and D. Gu, “Two-dimensional correction for min-sum decoding of irregular LDPC codes,” IEEE Communications Letters, vol. 10, no. 3, pp. 180–182, 2006.
14. V. Savin, “Self-corrected min-sum decoding of LDPC codes,” in IEEE Int. Symp. on Inf. Theory (ISIT), 2008, pp. 146–150.
15. S.K. Planjery, D. Declercq, L. Danjean, and B. Vasic, “Finite alphabet iterative decoders, part i: Decoding beyond belief propagation on bsc,” arXiv preprint arXiv:1207.4800, 2012.
16. V.C. Gaudet and A.C. Rapley, “Iterative decoding using stochastic computation,” IET Electronics Letters, vol. 39, no. 3, pp. 299–301, 2003.
17. W.J. Gross, V.C. Gaudet, and A. Milner, “Stochastic implementation of ldpc decoders,” in Proc. of the 39th Asilomar Conference on Signals, Systems and Computers. IEEE, 2005, pp. 713–717.
18. S. Sharifi Tehrani, W.J. Gross, and S. Mannor, “Stochastic decoding of LDPC codes,” IEEE Communications Letters, vol. 10, no. 10, pp. 716–718, 2006.
19. V. V. Zyablov and M. S. Pinsker, “Decoding complexity of low-density codes for transmission in a channel with erasures,” Problemy Peredachi Informatsii, vol. 10, no. 1, pp. 15–28, 1974.
20. M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in Proc. of the 29th annual ACM symposium on Theory of computing. ACM, 1997, pp. 150–159.
21. M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, “Efficient erasure correcting codes,” IEEE Trans. on Information Theory, vol. 47, no. 2, pp. 569–584, 2001.
22. M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, “Improved low-density parity-check codes using irregular graphs,” IEEE Trans. on Information Theory, vol. 47, no. 2, pp. 585–598, 2001.
23. D. Burshtein and G. Miller, “Efficient maximum-likelihood decoding of ldpc codes over the binary erasure channel,” IEEE Trans. on Information Theory, vol. 50, no. 11, pp. 2837–2844, 2004.

References

24. T.J. Richardson and R.L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” IEEE Trans. on Inf. Theory, vol. 47, no. 2, pp. 599–618, 2001.
25. T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. on Information Theory, vol. 47, no. 2, pp. 619–637, 2001.
26. G.D. Forney and D.J. Costello, “Channel coding: The road to channel capacity,” Proceedings of the IEEE, vol. 95, no. 6, pp. 1150–1177, 2007.
27. W. E. Ryan, “An introduction to LDPC codes,” in CRC Handbook for Coding and Signal Processing for Recording Systems. 2004, CRC Press.
28. A. Shokrollahi, “LDPC codes: An introduction,” Coding, cryptography and combinatorics, pp. 85–110, 2004.
29. V. Guruswami, “Iterative decoding of low-density parity check codes (an introductory survey),” in Bulletin of the European Association for Theoretical Computer Science (EATCS), Computational Complexity Column, 2006, vol. 90, pp. 53–88.
30. V. Savin, “LDPC decoders”, in Channel coding: Theory, algorithms, and applications, D. Declercq, M. Fossorier, and E. Biglieri editors, Academic Press Library in Mobile and Wireless Communications, Elsevier, June 2014.

Outline

- Coding for noisy channels: from Shannon to Shannon
  - Redundancy, linear codes and Shannon’s Theorem
  - LDPC codes and iterative message-passing decoders
  - Approaching the Shannon limit

- Coding for noisy channels with noisy devices
  - Noisy message-passing decoders
  - Impact of the “computation noise” on the error correction performance

Motivation

- Decoders running on noisy (faulty) devices?

1. Low-power / high-throughput decoders
   - Tradeoff between power consumption, latency, and reliability (e.g. aggressive voltage scaling – near/sub-threshold)

decoder implementation

2. Emerging technologies
   - Unreliability is of the most critical challenges for the next-generation electronic circuit design

3. Alternative fault-tolerance solutions
   - Reliable computing with unreliable components
   - Modular redundancy $\Rightarrow$ more powerful error correcting codes

fault tolerant circuit design

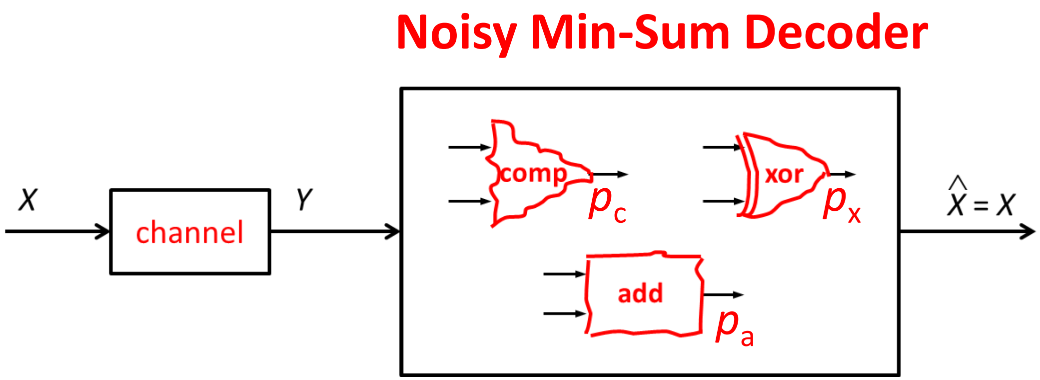
Noisy Message Passing Decoders

- Can MP decoders provide reliable error-protection if they are implemented in unreliable HW?
  - Unreliable HW $\Rightarrow$ new source of errors that occur during the decoding process

- **Intuition 1:** Yes they can!
  - Decoders deal with errors anyway; they should also be able to cope with HW-induced errors!

- **Intuition 2:** No, they can't!
  - Unreliable HW generates "computation errors", not "transmission errors", which can propagate in a catastrophic way through the iterative decoding process!

Noisy Min-Sum Decoder

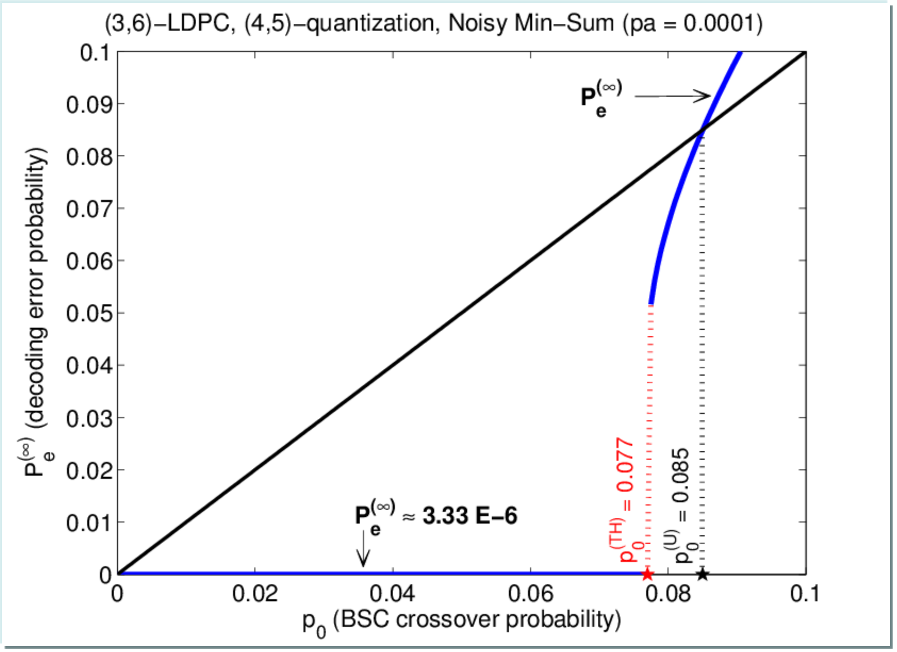


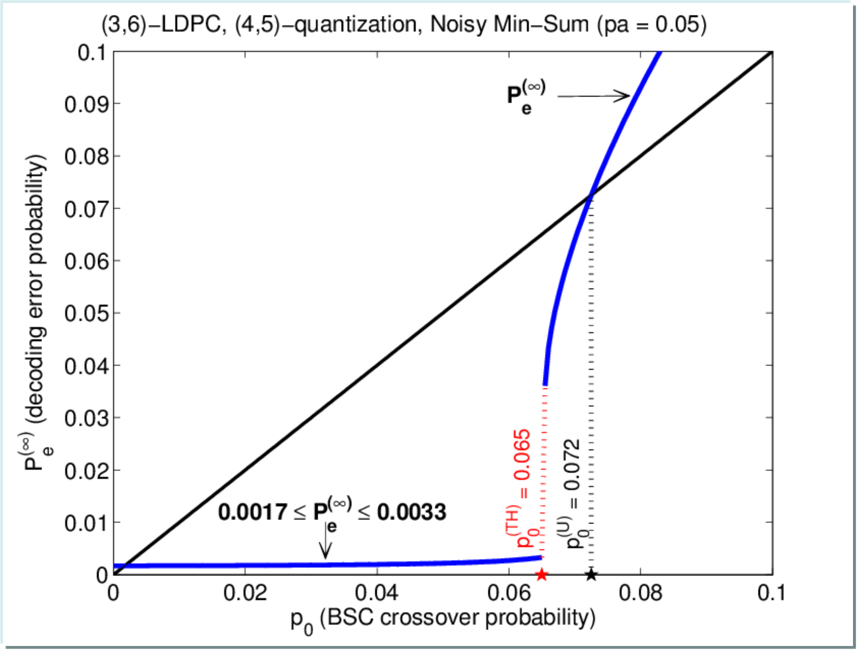
- Generalize the DE analysis to the case of noisy MS
  - Predict the behavior and the performance of the decoder without relying on extensive Monte Carlo simulations
  - Recursive DE equations

    $\Rightarrow$determine error probability at each iteration ℓ: $P_e^{(\ell)}(p_0, p_a, p_c, p_x)$

Analytical Results

- HW noise $\Rightarrow$ error probability is bounded above zero; However:
  - We can derive lower bound: tight for small p_c, p_x, p_a values
  - Threshold phenomenon similar to the noiseless case

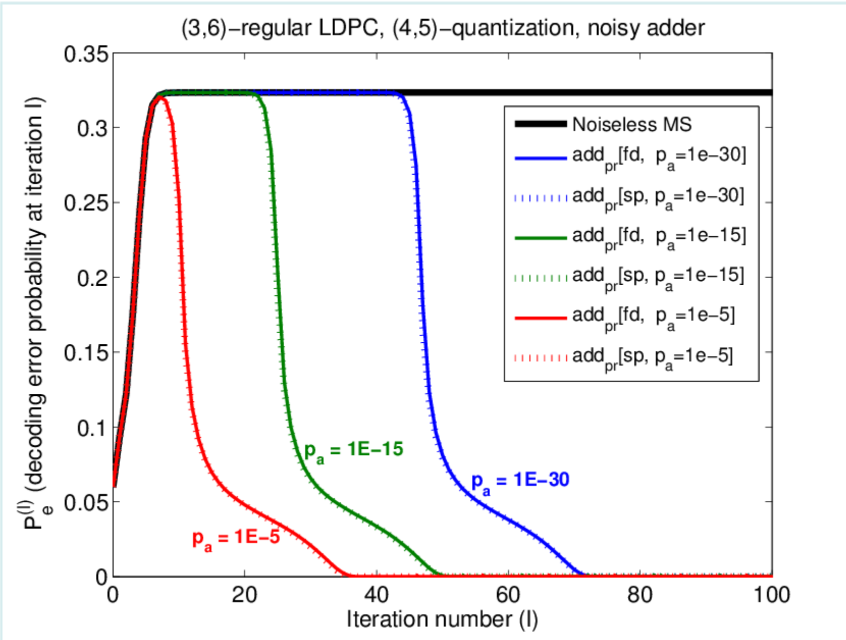


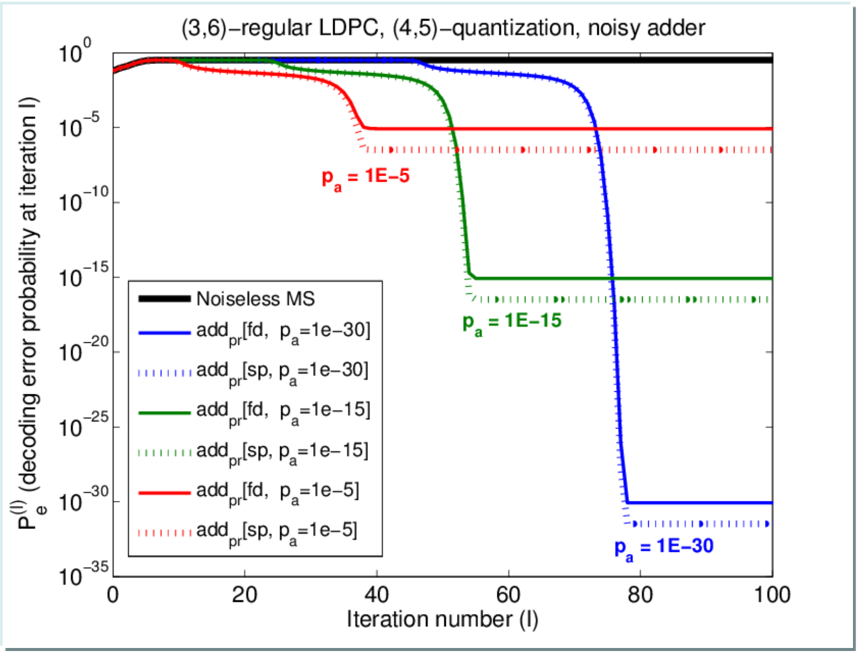


  - "Stable" decoder: error probability close to lower bound can be maintained for infinitely mainly iterations

Analytical Results

- HW noise can <u>sometimes</u> improve the error correction capability

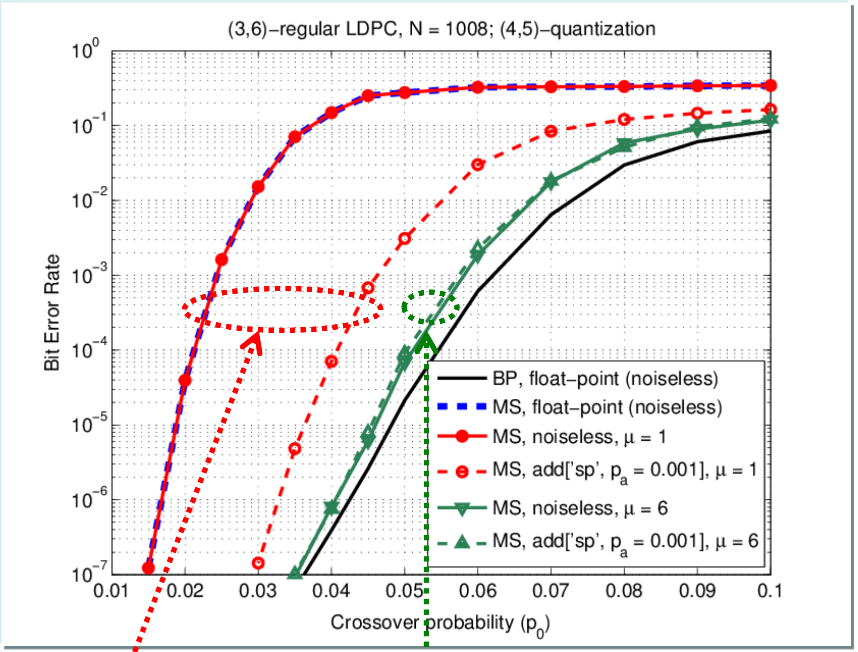




- HW noise helps the MS decoder to escape from fixed point attractors

Simulation Results

- solid curves: noiseless MS
- dashed curves: noisy MS

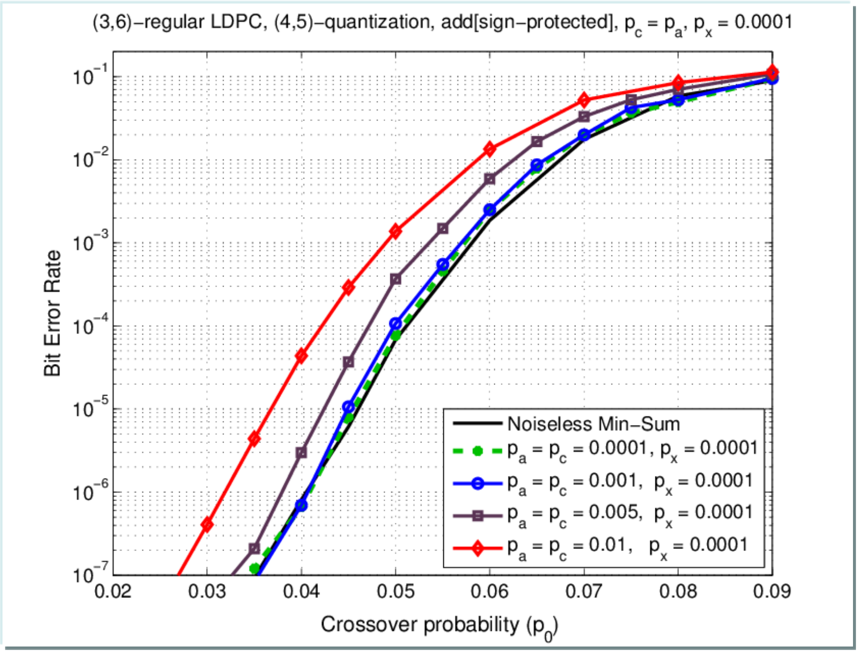


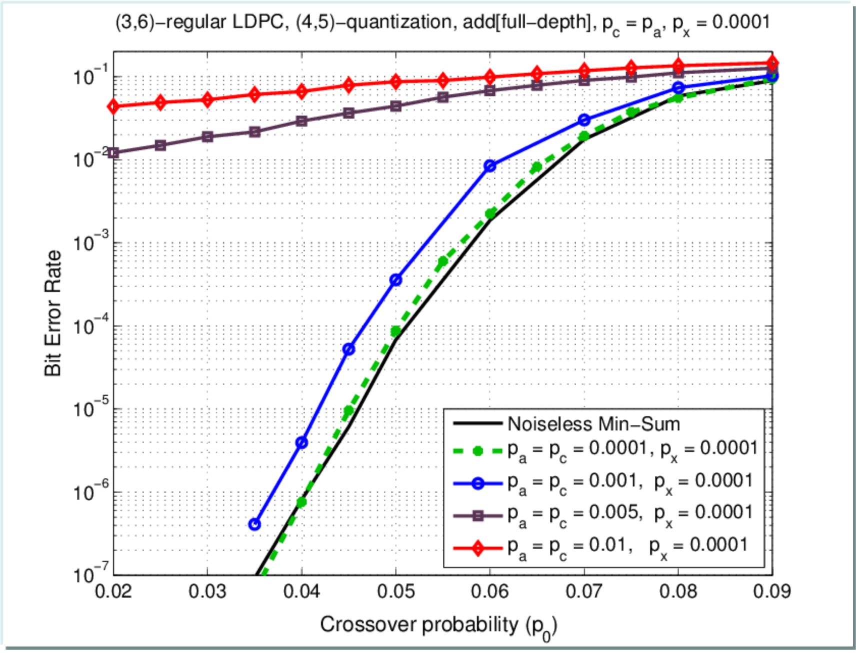
Noisy decoder outperforms noiseless one

similar performance

Simulation Results

- Black curve: noiseless MS
- Other curves: noisy MS with different noise parameters





Outline

- Coding for noisy channels: from Shannon to Shannon
  - Redundancy, linear codes and Shannon's Theorem
  - LDPC codes and iterative message-passing decoders
  - Approaching the Shannon limit
- Coding for noisy channels with noisy devices
  - Noisy message-passing decoders
  - Impact of the "computation noise" on the error correction performance
- Conclusion

Conclusion (second part)

- Unreliable devices: new paradigm in coding theory
- Analytical proof that iterative MP decoders can still operate with faulty hardware
  - we can predict the level of noise that can be tolerated
- Noisy threshold phenomenon
  - The functional threshold
- Corroboration of the asymptotic analysis through finite-length simulations

References (second part)

1. S. K. Chilappagari, M. Ivkovic, and B. Vasic, “Analysis of one step majority logic decoders constructed from faulty gates,” in Proc. of IEEE Int. Symp. on Information Theory, 2006, pp. 469–473.
2. B. Vasic and S. K. Chilappagari, “An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes,” IEEE Trans. on Circuits and Systems I: Regular Papers, vol. 54, no. 11, pp. 2438–2446, 2007.
3. L. R. Varshney, “Performance of LDPC codes under faulty iterative decoding,” IEEE Trans. Inf. Theory, vol. 57, no. 7, pp. 4427–4444, 2011.
4. S. Yazdi, C. Huang, and L. Dolecek, “Optimal design of a Gallager B noisy decoder for irregular LDPC codes,” IEEE Comm. Letters, vol. 16, no. 12, pp. 2052–2055, 2012.
5. S. Yazdi, H. Cho, and L. Dolecek, “Gallager B decoder on noisy hardware,” IEEE Trans. on Comm., vol. 66, no. 5, pp. 1660–1673, 2013.
6. A. Balatsoukas-Stimming, C. Studer, and A. Burg, “Characterization of min-sum decoding of LDPC codes on unreliable silicon,” in Information Theory and Applications Workshop (ITA), 2014.
7. A. Balatsoukas-Stimming and A. Burg, “Density evolution for min-sum decoding of LDPC codes under unreliable message storage,” IEEE Communications Letters, vol. PP, no. 99, pp. 1–4, 2014.
8. C. L. Kameni Ngassa, V. Savin, and D. Declercq, “Min-sum-based decoders running on noisy hardware,” in proc. of IEEE Global Communications Conference (GLOBECOM), 2013.
9. C. L. Ngassa, V. Savin, and D. Declercq, “Analysis of min-sum based decoders implemented on noisy hardware,” in Signals, Systems and Computers, 2013 Asilomar Conference on, 2013.
10. C. L. Kameni Ngassa, V. Savin, and D. Declercq, “Unconventional behavior of the noisy min-sum decoder over the binary symmetric channel,” in Information Theory and Applications Workshop (ITA), 2014

Thank you!