The Benefits of Being Erratic:

Correcting Errors with Noise-Enhanced Gradient Algorithms

Chris Winstead, Tasnuva Tithi (Utah State University)
Emmanuel Boutillon (Lab-STICC, Université de Bretagne Sud)
Fakhreddine Ghaffari (ETIS Lab, Université Cergy-Pontoise)

November 2017

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but. . .

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but. . .
  - Noise can be complex to produce;

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but...
  - Noise can be complex to produce;
  - Simpler procedures can supply the required diversity.

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but. . .
  - Noise can be complex to produce;
  - Simpler procedures can supply the required diversity.
- New results on diversity approaches:

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but. . .
  - Noise can be complex to produce;
  - Simpler procedures can supply the required diversity.
- New results on diversity approaches:
  - Simpler periodic perturbations

Outline

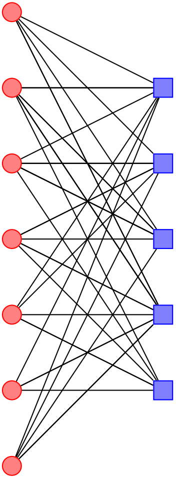
- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but...
  - Noise can be complex to produce;
  - Simpler procedures can supply the required diversity.
- New results on diversity approaches:
  - Simpler periodic perturbations
  - Parity-only bit flipping

Outline

- Noise enhancement is a recently studied effect in error correction algorithms, especially low-density parity-check (LDPC) codes.
- We interpret noise as a form of algorithm diversity.
  - Noise is analyzable using statistical methods, but...
  - Noise can be complex to produce;
  - Simpler procedures can supply the required diversity.
- New results on diversity approaches:
  - Simpler periodic perturbations
  - Parity-only bit flipping
  - Threshold sequences tailored to dominant trapping sets.

Introduction

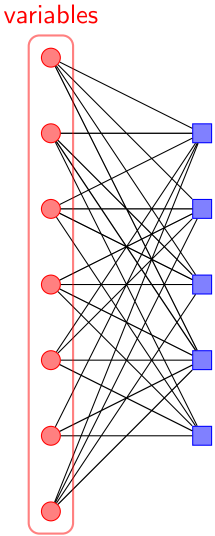
LDPC codes are a class of large-frame error correcting code defined by a constraint graph or Tanner graph.



Introduction

LDPC codes are a class of large-frame error correcting code defined by a constraint graph or Tanner graph.

The graph's variables are binary bits in a data frame.

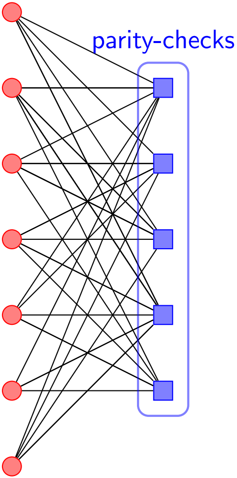


Introduction

LDPC codes are a class of large-frame error correcting code defined by a constraint graph or Tanner graph.

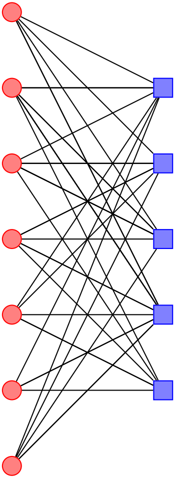
The graph's variables are binary bits in a data frame.

The graph's parity-checks constrain even parity among adjacent bits.



Introduction

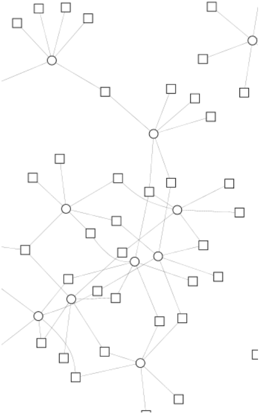
The goal of decoding is to obtain the *most probable* bit values that satisfy all of the parity-check constraints.



Introduction

The goal of decoding is to obtain the *most probable* bit values that satisfy all of the parity-check constraints.

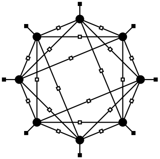
This problem is usually modeled as a Bayesian Belief Network, using the Belief Propagation (BP) algorithm. Probability calculations can be expensive, and **cycles in the network** distort the calculations, leading to failures.

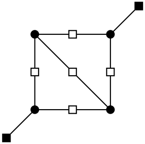


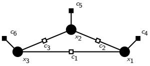
Trapping Sets

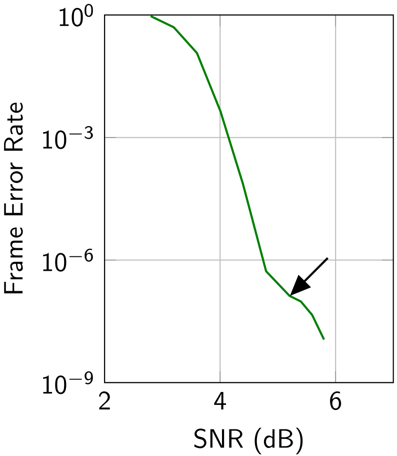
Trapping sets or Absorbing Sets are repeated cyclic patterns in the code graph.

They induce error floors in LDPC decoders based on BP.







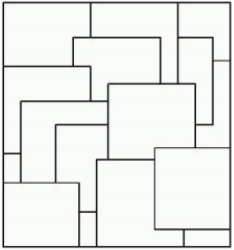


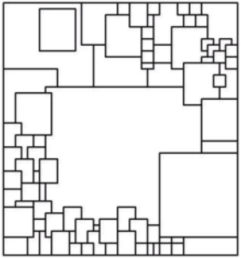
FAIDs & Decoder Diversity

There are many approaches for dealing with absorbing sets.

The general framework of Finite Alphabet Iterative Decoders (FAID) describes a class of decoders by LUTs rather than explicit Bayesian calculations[a].

Any particular FAID has small error coverage, but decoder diversity guarantees total coverage by alternating between different LUTs.





[a]Planjery et al. 2013; Declercq et al. 2013.

Single-Bit Algorithms

Several classes of low-complexity algorithms have been studied. Some notable algorithms:

- Stochastic Decoding[1] – Emulate BP using stochastic arithmetic.

[1] Gaudet and Rapley 2003.
[2] Winstead and Boutillon 2014.
[3] Mobini and Hemati 2009.

Single-Bit Algorithms

Several classes of low-complexity algorithms have been studied. Some notable algorithms:

- Stochastic Decoding[1] – Emulate BP using stochastic arithmetic.
- Maximum Likelihood Single-Bit Message (MLSBM)[2] – Similar to FAID, formal procedure to design LUTs.

[1] Gaudet and Rapley 2003.
[2] Winstead and Boutillon 2014.
[3] Mobini and Hemati 2009.

Single-Bit Algorithms

Several classes of low-complexity algorithms have been studied. Some notable algorithms:

- Stochastic Decoding[1] – Emulate BP using stochastic arithmetic.
- Maximum Likelihood Single-Bit Message (MLSBM)[2] – Similar to FAID, formal procedure to design LUTs.
- Differential Decoding w/ Binary Message Passing (DD-BMP)[3] – Uses a memory effect inspired by relaxation algorithms.

[1] Gaudet and Rapley 2003.
[2] Winstead and Boutillon 2014.
[3] Mobini and Hemati 2009.

Single-Bit Algorithms

Several classes of low-complexity algorithms have been studied. Some notable algorithms:

- Stochastic Decoding[1] – Emulate BP using stochastic arithmetic.
- Maximum Likelihood Single-Bit Message (MLSBM)[2] – Similar to FAID, formal procedure to design LUTs.
- Differential Decoding w/ Binary Message Passing (DD-BMP)[3] – Uses a memory effect inspired by relaxation algorithms.
- Bit-Flipping – Heuristics *loosely based* on BP or gradient descent.

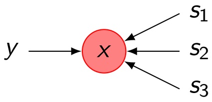
[1]Gaudet and Rapley 2003.
[2]Winstead and Boutillon 2014.
[3]Mobini and Hemati 2009.

Parallel Bit-Flipping Decoders

A binary alternative, simpler than BP or Stochastic decoders.

Inputs

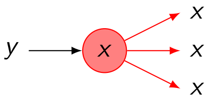
in



channel sample y_i.
hypothesis $x_i \in \{-1, +1\}$.
parity checks $s_j \in \{-1, +1\}$

Operations

out



Reliability metric: Δ_i (unique to algorithm)
Threshold update: flip x_i if $\Delta_i \leq \theta$

Then transmit x_i to adjacent nodes.

Example Algorithms

Impr. Diff. Binary (IDB):[4] $\Delta_i^{(t+1)} = \Delta_i^{(t)} + w \sum_{j \in \mathcal{M}_i \setminus j} s_{i,j} - d\, x_i^{(t)}$

Grad. Desc. BF (GDBF):[5] $\Delta_i^{(t+1)} = x_i y_i + \sum_j s_j$

Noisy GDBF:[6] $\Delta_i^{(t+1)} = x_i y_i + w \sum_j s_j + q$

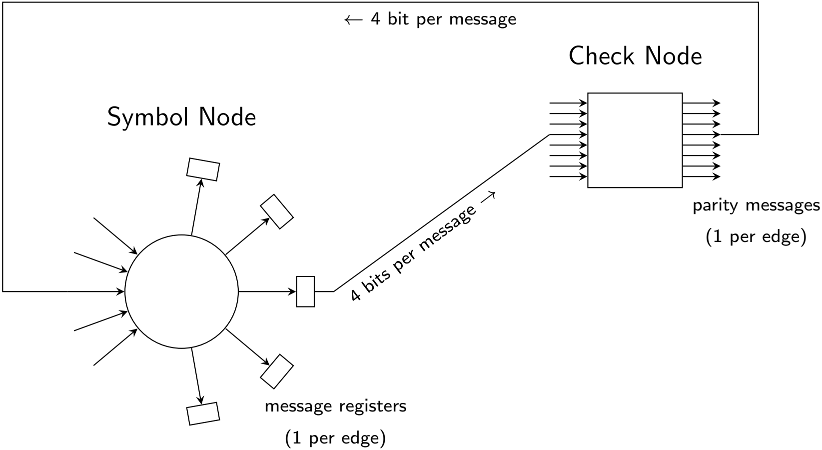
Probabilistic GDBF:[7] If $\Delta_i < \theta$, flip *with probability* p.

Traditional rule of thumb: higher complexity = better performance.

[4] Cushon et al. 2014.
[5] Wadayama et al. 2008.
[6] Sundararajan, Winstead, and Boutillon 2014.
[7] Rasheed, Ivanis, and Vasic 2014.

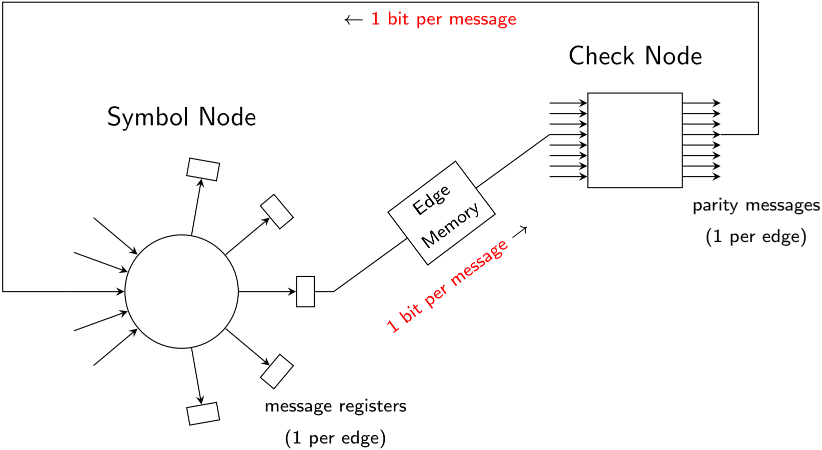
Motivation: Reduced Complexity

Standard BP-based or FAID algorithm:



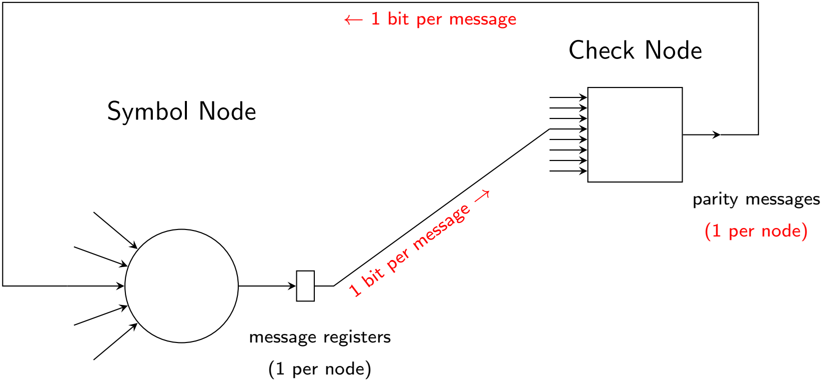
Motivation: Reduced Complexity

Stochastic algorithm (successive relaxation):



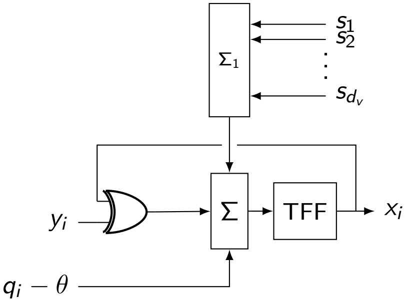
Motivation: Reduced Complexity

Bit-flipping algorithm:

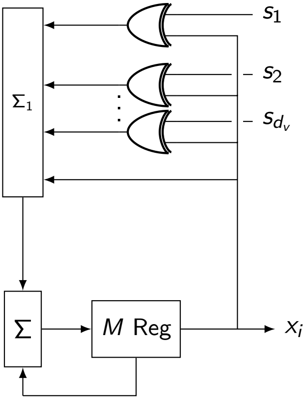


Example Node Designs

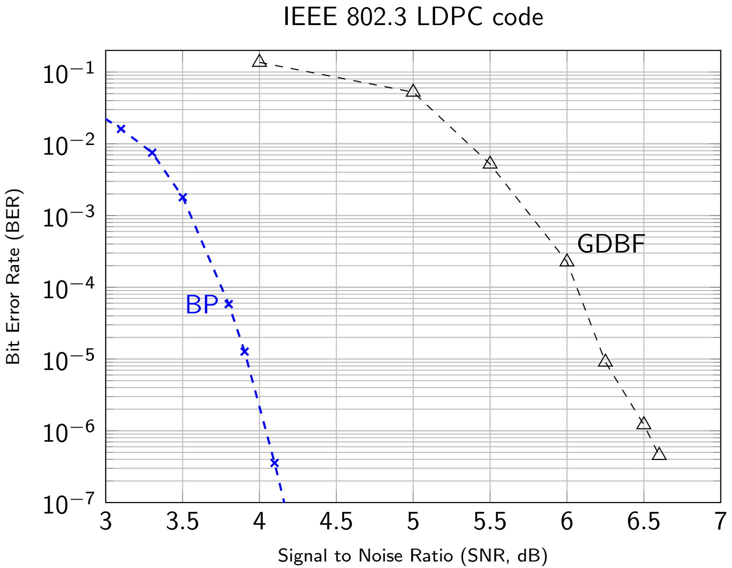
NGDBF



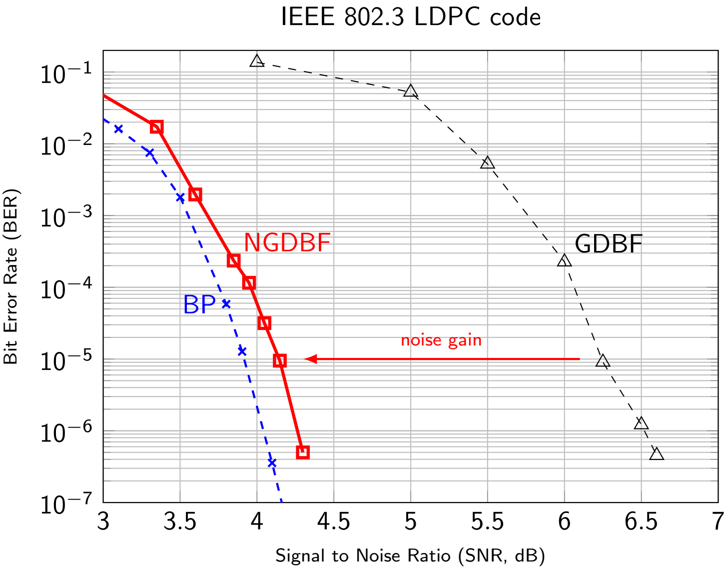
IDB



Performance: Noise Enhancement

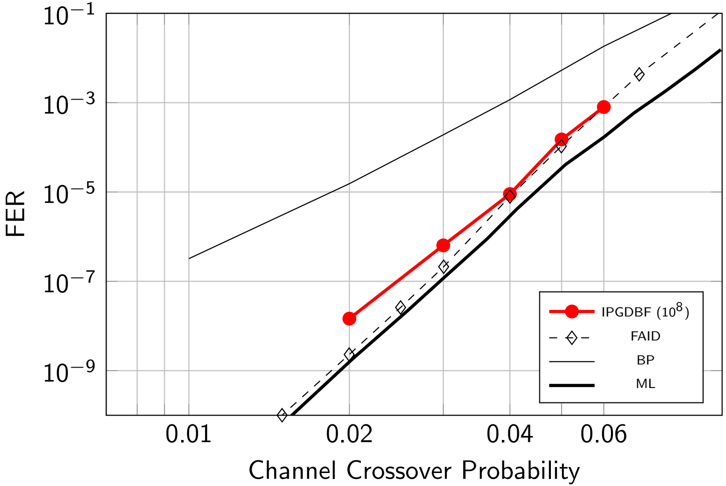


Performance: Noise Enhancement



PGDBF FER

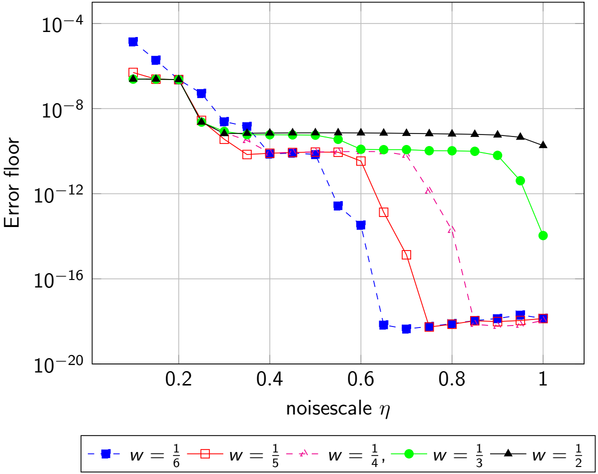
FAID and PGDBF approach Maximum Likelihood (ML) limit[8]



[8]Declercq et al. 2016.

We know that noise works...

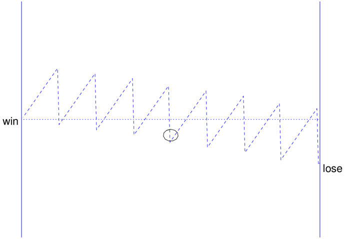
Treating each trapping set as a Markov process, we can predict the error rate and optimize algorithm parameters:

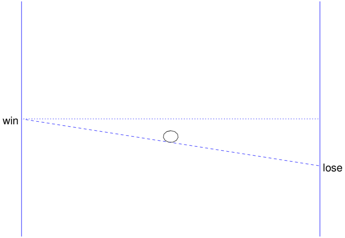


What is the bigger picture? *Why* does non-determinism enhance these algorithms?

Parrondo's "paradox"

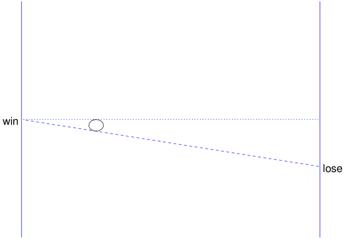
Parrondo's paradox considers noise-perturbed particles on various sloped surfaces. These are "failing" strategies:[9]





Parrondo's "paradox"

But by alternating between losing strategies at the right times, and with the right amount of noise energy, it's possible to "win" with high probability:



Diversity in Parrondo's Ratchet

- Parrondo's paradox is an example of algorithmic diversity.

Diversity in Parrondo's Ratchet

- Parrondo's paradox is an example of algorithmic diversity.
- Noise is important to the demonstration only as another source of diversity – it could be replaced by additional strategies.

Diversity in Parrondo's Ratchet

- Parrondo's paradox is an example of algorithmic diversity.
- Noise is important to the demonstration only as another source of diversity – it could be replaced by additional strategies.
- Bit-flipping algorithms are "losers," and noise turns them into "winners" . . .

Diversity in Parrondo's Ratchet

- Parrondo's paradox is an example of algorithmic diversity.
- Noise is important to the demonstration only as another source of diversity – it could be replaced by additional strategies.
- Bit-flipping algorithms are "losers," and noise turns them into "winners" . . .
- But maybe there are other ways to achieve this diversity.

Parity-Only Bit Flipping

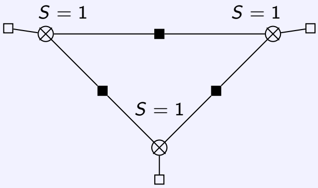
When NGDBF fails in the error floor region, it is most likely due to a single trapping set. We can try to correct it by switching strategies:

$$\text{NGDBF calculation:} \quad E_i = x_i y_i + w \sum s_{ij} + q_i$$

$$\text{New strategy:} \quad E_i = \cancel{x_i y_i} + w \sum s_{ij} + \cancel{q_i}$$

In this strategy, diversity is supplied by a dynamic threshold sequence.

Example



If all symbols are errors, a threshold of $\theta = 1$ will correct them all.

Parity-Only Bit Flipping

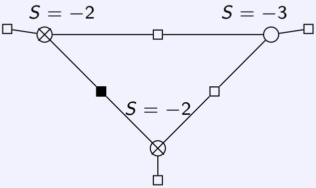
When NGDBF fails in the error floor region, it is most likely due to a single trapping set. We can try to correct it by switching strategies:

$$\text{NGDBF calculation:} \quad E_i = x_i y_i + w \sum s_{ij} + q_i$$

$$\text{New strategy:} \quad E_i = \cancel{x_i y_i} + w \sum s_{ij} + \cancel{q_i}$$

In this strategy, diversity is supplied by a dynamic threshold sequence.

Example



If two are errors, a threshold of $\theta = -3$ will correct two but leave one uncorrected error.

Parity-Only Bit Flipping

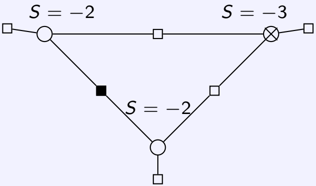
When NGDBF fails in the error floor region, it is most likely due to a single trapping set. We can try to correct it by switching strategies:

$$\text{NGDBF calculation:} \quad E_i = x_i y_i + w \sum s_{ij} + q_i$$

$$\text{New strategy:} \quad E_i = \cancel{x_i y_i} + w \sum s_{ij} + \cancel{q_i}$$

In this strategy, diversity is supplied by a dynamic threshold sequence.

Example



A threshold of $\theta = -3$ corrects a single error.

Parity-Only Bit Flipping

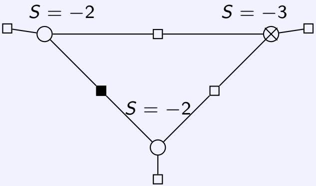
When NGDBF fails in the error floor region, it is most likely due to a single trapping set. We can try to correct it by switching strategies:

$$\text{NGDBF calculation:} \quad E_i = x_i y_i + w \sum s_{ij} + q_i$$

$$\text{New strategy:} \quad E_i = \cancel{x_i y_i} + w \sum s_{ij} + \cancel{q_i}$$

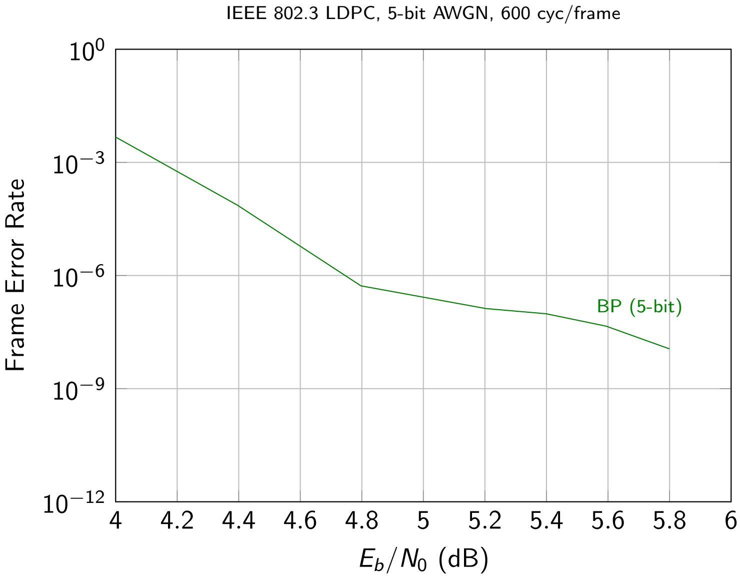
In this strategy, diversity is supplied by a dynamic threshold sequence.

Example

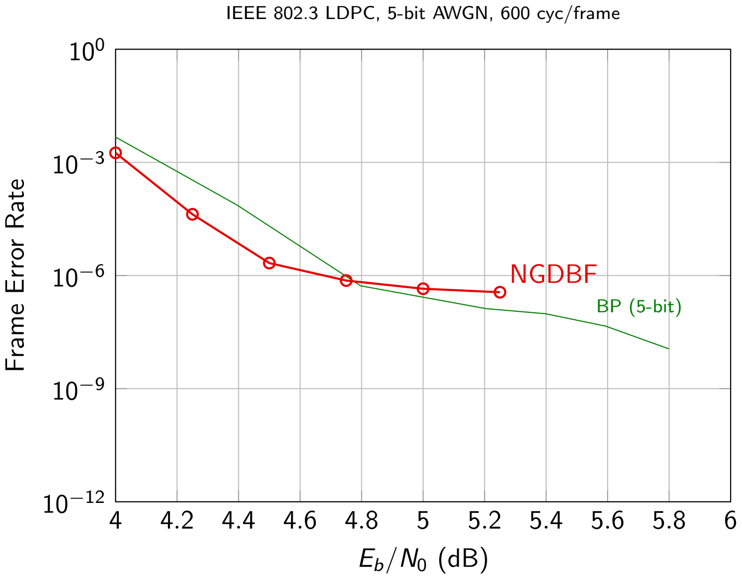


A dynamic sequence $\theta = 1, -3, -3$ covers all errors for this trapping set.

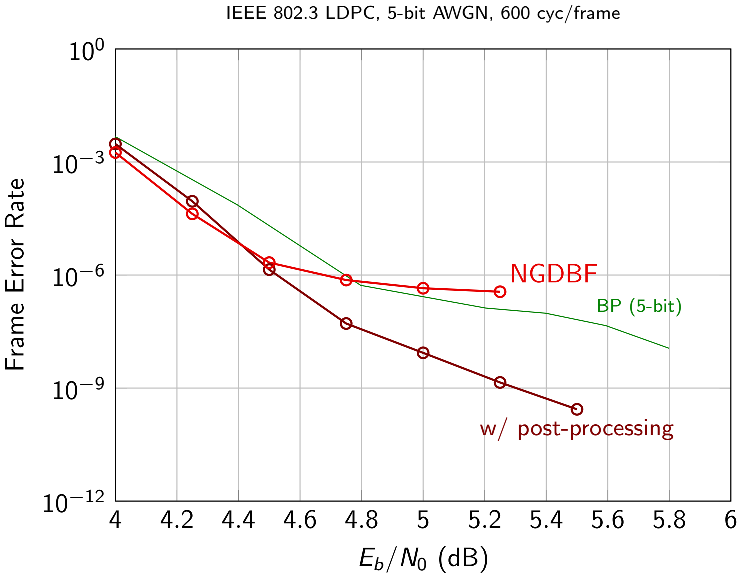
Lowering the Error Floor



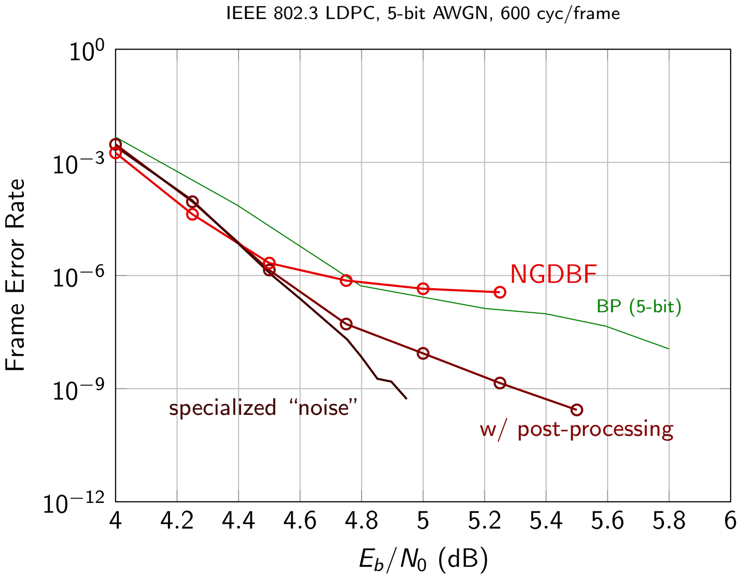
Lowering the Error Floor



Lowering the Error Floor



Lowering the Error Floor



Discussion

- The error floor is steadily disappearing...

Discussion

- The error floor is steadily disappearing...
- Each improvement simultaneously reduces implementation complexity while enhancing performance.

Discussion

- The error floor is steadily disappearing. . .
- Each improvement simultaneously reduces implementation complexity while enhancing performance.
- We now have a basic theory to explain and optimize noise enhancement.

Discussion

- The error floor is steadily disappearing...
- Each improvement simultaneously reduces implementation complexity while enhancing performance.
- We now have a basic theory to explain and optimize noise enhancement.
- Algorithm diversity remains an open frontier.

References I

Planjery, S. K. et al. (2013). "Finite Alphabet Iterative Decoders Part I: Decoding Beyond Belief Propagation on the Binary Symmetric Channel". In: *IEEE Trans. Comm.* DOI: 10.1109/TCOMM.2013.090513.120443.

Declercq, D. et al. (2013). "Finite Alphabet Iterative Decoders Part II: Towards Guaranteed Error Correction of LDPC Codes via Iterative Decoder Diversity". In: *IEEE Trans. Comm.* DOI: 10.1109/TCOMM.2013.090513.120444.

Gaudet and Rapley (2003). "Iterative decoding using stochastic computation". In: *Electronics Letters.*

Winstead and Boutillon (2014). "Decoding LDPC Codes With Locally Maximum-Likelihood Binary Messages". In: *IEEE Comm. Lett.*

Mobini, Banihashemi and Hemati (2009). "A differential binary message-passing LDPC decoder". In: *IEEE Trans. Comm.* DOI: 10.1109/TCOMM.2009.09.070617.

References II

- Cushon, K. et al. (2014). "High-Throughput Energy-Efficient LDPC Decoders Using Differential Binary Message Passing". In: *Signal Processing, IEEE Transactions on* 62.3, pp. 619–631. ISSN: 1053-587X. DOI: 10.1109/TSP.2013.2293116.
- Wadayama, T. et al. (2008). "Gradient descent bit flipping algorithms for decoding LDPC codes". In: *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, pp. 1 –6. DOI: 10.1109/ISITA.2008.4895387.
- Sundararajan, Gopalakrishnan, Chris Winstead, and Emmanuel Boutillon (2014). *Noisy Gradient Descent Bit-Flip Decoding for LDPC Codes.* arXiv:1402.2773. URL: http://arxiv.org/abs/1402.2773.
- Rasheed, O.-A., P. Ivanis, and B. Vasic (2014). "Fault-Tolerant Probabilistic Gradient-Descent Bit Flipping Decoders". In: *IEEE Commun. Letters* 18.9, pp. 1487 –1490.
- Declercq, David et al. (2016). "Noise-aided gradient descent bit-flipping decoders approaching maximum likelihood decoding". In: *IEEE ISTC.*

References III

Parrondo, Juan MR, Gregory P Harmer, and Derek Abbott (2000). "New paradoxical games based on Brownian ratchets". In: *Physical Review Letters* 85.24, p. 5226.