Application of Optimization Algorithms for Channel Decoding

Midterm Presentation, 27.01.2023

Andreas Tsouchlos

Overview

- Theoretical Background
  - Motivation
  - Presumptions
  - Optimization as a Decoding Method
- Decoding Algorithms
  - Proximal Decoding
  - LP Decoding
- Analysis
  - Proximal Decoding
- Forthcoming Analysis
  - LP Decoding

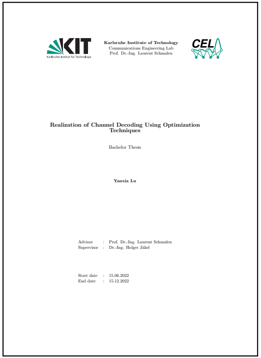
Overview

- Theoretical Background
  - Motivation
  - Presumptions
  - Optimization as a Decoding Method
- Decoding Algorithms
  - Proximal Decoding
  - LP Decoding
- Analysis
  - Proximal Decoding
- Forthcoming Analysis
  - LP Decoding

Motivation

- The general [ML] decoding problem for linear codes and the general problem of finding the weights of a linear code are both NP-complete. [BMT78]
- The iterative message–passing algorithms preferred in practice do not guarantee optimality and may fail to decode correctly when the graph contains cycles. [KTP19]
- The standard message-passing algorithms used for decoding [LDPC and turbo codes] are often difficult to analyze. [Fel03]

Previous Work



Karlsruhe Institute of Technology
Communications Engineering Lab
Prof. Dr.-Ing. Laurent Schmalen

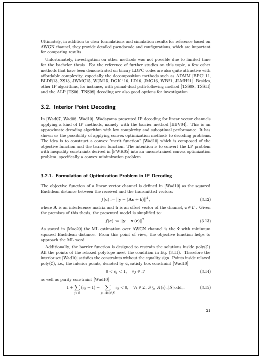
Realization of Channel Decoding Using Optimization Techniques

Bachelor Thesis

Yanxia Lu

Advisor : Prof. Dr.-Ing. Laurent Schmalen
Supervisor : Dr.-Ing. Holger Jäkel

Start date : 15.06.2022
End date : 15.12.2022



Ultimately, in addition to clear formulations and simulation results for reference based on AWGN channel, they provide detailed pseudocode and configurations, which are important for comparing results.

Unfortunately, investigation on other methods was not possible due to limited time for the bachelor thesis. For the reference of further studies on this topic, a few other methods that have been demonstrated on binary LDPC codes are also quite attractive with affordable complexity, especially the decomposition methods such as ADMM [BPC+11, BLDR13, ZS13, JWMC15, WJM15, DGK+16, LD16, JMG16, WB21, JLMH21]. Besides, other IP algorithms, for instance, with primal-dual path-following method [TSS08, TSS11] and the ALP [TS06, TNS08] decoding are also good options for investigation.

3.2. Interior Point Decoding

In [Wad07, Wad08, Wad10], Wadayama presented IP decoding for linear vector channels applying a kind of IP methods, namely with the barrier method [BBV04]. This is an approximate decoding algorithm with low complexity and suboptimal performance. It has shown us the possibility of applying convex optimization methods to decoding problems. The idea is to construct a convex 'merit function' [Wad10] which is composed of the objective function and the barrier function. The intention is to convert the LP problem with inequality constraints derived in [FWK05] into an unconstrained convex optimization problem, specifically a convex minimization problem.

3.2.1. Formulation of Optimization Problem in IP Decoding

The objective function of a linear vector channel is defined in [Wad10] as the squared Euclidean distance between the received and the transmitted vectors:

$$f(\mathbf{c}) := \|\mathbf{y} - (\mathbf{A}\mathbf{c} + \mathbf{b})\|^2 , \tag{3.12}$$

where $\mathbf{A}$ is an interference matrix and $\mathbf{b}$ is an offset vector of the channel, $\mathbf{c} \in \mathcal{C}$. Given the premises of this thesis, the presented model is simplified to:

$$f(\mathbf{c}) := \|\mathbf{y} - \mathbf{x}(\mathbf{c})\|^2 . \tag{3.13}$$

As stated in [Moo20] the ML estimation over AWGN channel is the $\hat{\mathbf{x}}$ with minimum squared Euclidean distance. From this point of view, the objective function helps to approach the ML word.

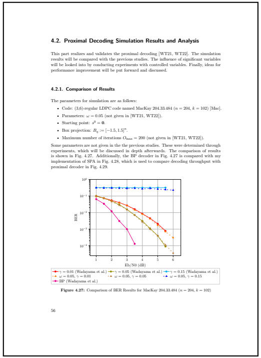
Additionally, the barrier function is designed to restrain the solutions inside poly($\mathcal{C}$). All the points of the relaxed polytope meet the condition in Eq. (3.11). Therefore the interior set [Wad10] satisfies the constraints without the equality sign. Points inside relaxed poly($\mathcal{C}$), i.e., the interior points, denoted by $\bar{\mathbf{c}}$, satisfy box constraint [Wad10]

$$0 < \bar{c}_j < 1, \quad \forall j \in \mathcal{J} \tag{3.14}$$

as well as parity constraint [Wad10]

$$1 + \sum_{j \in S} (\bar{c}_j - 1) - \sum_{j \in A(i) \setminus S} \bar{c}_j < 0, \quad \forall i \in \mathcal{I},\ S \subseteq A(i), |S| \text{ odd}, . \tag{3.15}$$

21



4.2. Proximal Decoding Simulation Results and Analysis

This part realizes and validates the proximal decoding [WT21, WT22]. The simulation results will be compared with the previous studies. The influence of significant variables will be looked into by conducting experiments with controlled variables. Finally, ideas for performance improvement will be put forward and discussed.

4.2.1. Comparison of Results

The parameters for simulation are as follows:

- Code: (3,6)-regular LDPC code named MacKay 204.33.484 ($n = 204$, $k = 102$) [Mac].
- Parameters: $\omega = 0.05$ (not given in [WT21, WT22]).
- Starting point: $s^0 = \mathbf{0}$.
- Box projection: $B_\eta := [-1.5, 1.5]^n$.
- Maximum number of iterations $O_{max} = 200$ (not given in [WT21, WT22]).

Some parameters are not given in the the previous studies. These were determined through experiments, which will be discussed in depth afterwards. The comparison of results is shown in Fig. 4.27. Additionally, the BP decoder in Fig. 4.27 is compared with my implementation of SPA in Fig. 4.28, which is used to compare decoding throughput with proximal decoder in Fig. 4.29.

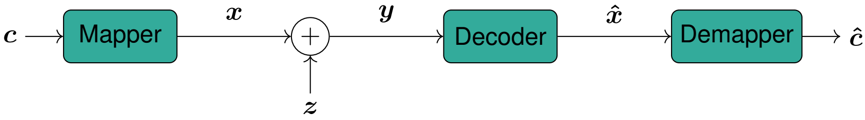
Figure 4.27: Comparison of BER Results for MacKay 204.33.484 ($n = 204$, $k = 102$)

56

Figure: Bachelor's Thesis by Yanxia Lu [Lu23]

- Analysis of "Proximal Decoding"
- Analysis of "Interior Point Decoding"

Presumptions: Channel & Modulation



- All simulations are performed with BPSK:

$$\boldsymbol{x} = (-1)^{\boldsymbol{c}}, \qquad \boldsymbol{c} \in \mathbb{F}_2^n, \quad \boldsymbol{x} \in \mathbb{R}^n$$

- The channel model is AWGN:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}, \qquad \boldsymbol{z} \sim \mathcal{N}\left(0, \frac{1}{2}\left(\frac{k}{n}\frac{E_b}{N_0}\right)^{-1}\right), \quad \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$$

- All-zeros assumption:

$$\boldsymbol{c} = \boldsymbol{0}$$

Optimization as a Decoding Method

- Reformulate decoding problem as optimization problem
  - Establish objective function
  - Establish constraints
- Use optimization method to solve the new problem

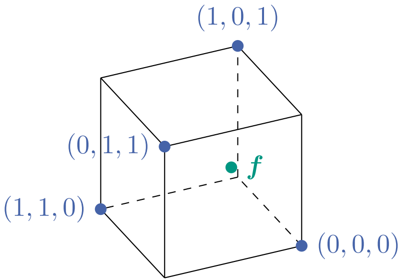


Figure: Hypercube ($n = 3$) with valid codewords

Overview

Karlsruhe Institute of Technology

- Theoretical Background
  - Motivation
  - Presumptions
  - Optimization as a Decoding Method
- Decoding Algorithms
  - Proximal Decoding
  - LP Decoding
- Analysis
  - Proximal Decoding
- Forthcoming Analysis
  - LP Decoding

Proximal Decoding: General Idea [WT22]

- MAP rule:

$$\hat{\boldsymbol{x}} = \underset{x \in \mathbb{R}^n}{\arg\max}\, f_{\boldsymbol{Y}}\left(\boldsymbol{y}|\boldsymbol{x}\right) f_{\boldsymbol{X}}\left(\boldsymbol{x}\right)$$

$$= \underset{x \in \mathbb{R}^n}{\arg\max}\, e^{-L(\boldsymbol{y}|\boldsymbol{x})} f_{\boldsymbol{X}}\left(\boldsymbol{x}\right), \qquad L\left(\boldsymbol{y}|\boldsymbol{x}\right) = -\ln\left(f_{\boldsymbol{Y}}\left(\boldsymbol{y}|\boldsymbol{x}\right)\right)$$

- Approximation of prior PDF:

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}\right) = \frac{1}{\left|\mathcal{C}\left(\boldsymbol{H}\right)\right|} \sum_{\boldsymbol{c} \in \mathcal{C}(\boldsymbol{H})} \delta\left(\boldsymbol{x} - (-1)^{\boldsymbol{c}}\right) \approx \frac{1}{Z} e^{-\gamma h(\boldsymbol{x})}$$

- Code constraint polynomial:

$$h\left(\boldsymbol{x}\right) = \underbrace{\sum_{j=1}^{n} \left(x_j^2 - 1\right)^2}_{\text{Bipolar constraint}} + \underbrace{\sum_{i=1}^{m} \left[\left(\prod_{j \in \mathcal{A}(i)} x_j\right) - 1\right]^2}_{\text{Parity constraint}},$$

$$\mathcal{I} \equiv [1:n], \quad \mathcal{J} \equiv [1:m]$$

$$\mathcal{A}\left(i\right) \equiv \left\{j | j \in \mathcal{J}, \boldsymbol{H}_{j,i} = 1\right\}, i \in \mathcal{I}$$

Proximal Decoding: General Idea

Karlsruhe Institute of Technology

- Objective function:

$$f(\boldsymbol{x}) = L(\boldsymbol{y}|\boldsymbol{x}) + \gamma h(\boldsymbol{x})$$

- Proximal operator [PB14]:

$$\begin{aligned}\text{prox}_{\gamma h}(\boldsymbol{x}) &\equiv \underset{\boldsymbol{t}\in\mathbb{R}^n}{\arg\min}\left(\gamma h(\boldsymbol{t}) + \frac{1}{2}\|\boldsymbol{t}-\boldsymbol{x}\|^2\right)\\ &\approx \boldsymbol{x} - \gamma\nabla h(\boldsymbol{x}), \qquad \gamma \text{ small}\end{aligned}$$

- Iterative decoding process:

$$\begin{aligned}\boldsymbol{r} &\leftarrow \boldsymbol{s} - \omega\nabla L(\boldsymbol{y}|\boldsymbol{s}), \qquad \omega > 0 \qquad \text{“Gradient descent step”}\\ \boldsymbol{s} &\leftarrow \boldsymbol{r} - \gamma\nabla h(\boldsymbol{r}), \qquad \gamma > 0 \qquad \text{“Code proximal step”}\end{aligned}$$

Proximal Decoding: Algorithm

- Iterative decoding algorithm [WT22]:

$\boldsymbol{s} \leftarrow \boldsymbol{0}$
for K iterations **do**
    $\boldsymbol{r} \leftarrow \boldsymbol{s} - \omega \nabla L \left(\boldsymbol{y} \mid \boldsymbol{s} \right)$
    $\boldsymbol{s} \leftarrow \boldsymbol{r} - \gamma \nabla h \left(\boldsymbol{r} \right)$
    $\hat{\boldsymbol{x}} \leftarrow \text{sign}\left(\boldsymbol{s} \right)$
    if $\boldsymbol{H} \hat{\boldsymbol{c}} = \boldsymbol{0}$ **do**
        return $\hat{\boldsymbol{c}}$
    end if
end for
return $\hat{\boldsymbol{c}}$

LP Decoding [FWK05]

- Codeword polytope:

$$\text{poly}\left(\mathcal{C}\right) = \left\{ \sum_{\boldsymbol{c}\in\mathcal{C}} \lambda_{\boldsymbol{c}} \boldsymbol{c} : \lambda_{\boldsymbol{c}} \geq 0, \sum_{\boldsymbol{c}\in\mathcal{C}} \lambda_{\boldsymbol{c}} = 1 \right\}, \qquad \lambda_{\boldsymbol{c}} \in \mathbb{R}_{\geq 0}$$

- Cost function:

$$\sum_{i=1}^{n} \gamma_i c_i, \qquad \gamma_i = \ln\left(\frac{P\left(Y = y_i | C = 0\right)}{P\left(Y = y_i | C = 1\right)}\right)$$

- LP formulation of ML decoding:

$$\text{minimize} \sum_{i=1}^{n} \gamma_i f_i$$

subject to $\boldsymbol{f} \in \text{poly}\left(\mathcal{C}\right)$

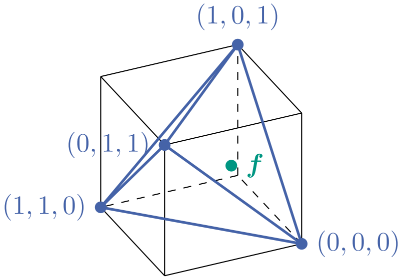


Figure: poly $\left(\mathcal{C}\right)$ for $n = 3$

Overview

- Theoretical Background
  - Motivation
  - Presumptions
  - Optimization as a Decoding Method
- Decoding Algorithms
  - Proximal Decoding
  - LP Decoding
- Analysis
  - Proximal Decoding
- Forthcoming Analysis
  - LP Decoding

Proximal Decoding: Bit Error Rate and Performance

Karlsruhe Institute of Technology

- Comparison of simulation[1] with results of Wadayama et al. [WT22]

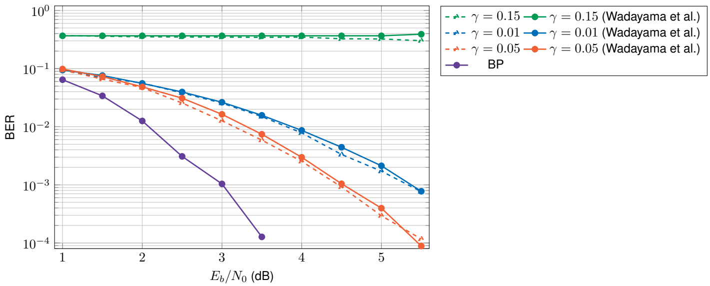


Figure: Simulation results for $\omega = 0.05, K = 100$

- $\mathcal{O}(n)$ time complexity - same as BP; only multiplication and addition necessary [WT22]
- Measured performance: $\sim 10\,000$ frames/s on Intel Core i7-7700HQ @ 2.80GHz; $n = 204$

[1] (3,6) regular LDPC code with $n = 204, k = 102$ [Mac23, 204.33.484]

Proximal Decoding: Choice of γ

Karlsruhe Institute of Technology

- Simulation[1] results for different values of γ

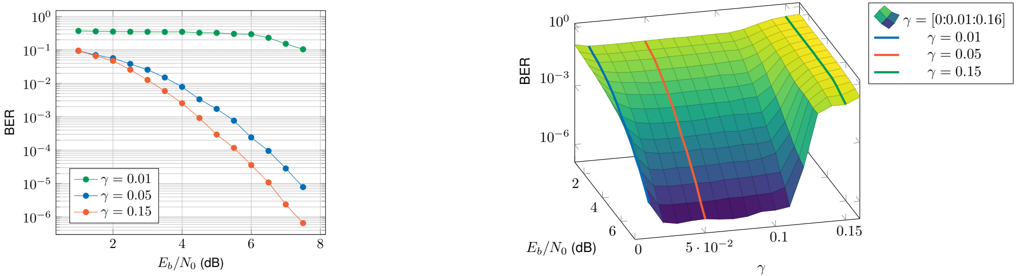
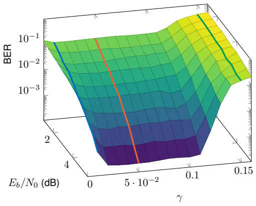


Figure: BER for $\omega = 0.05, K = 100$

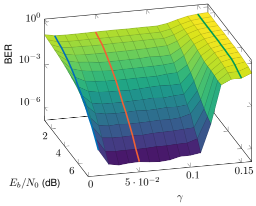
- Not great benefit in finding the optimal value for γ

[1] (3,6) regular LDPC code with $n = 204, k = 102$ [Mac23, 204.33.484]

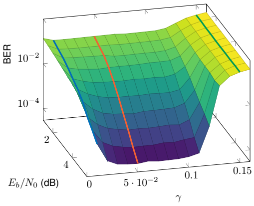
Proximal Decoding: Choice of γ



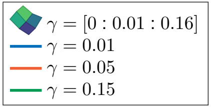
(a) $(3, 6)$-regular LDPC code with $n = 96, k = 48$ [Mac23, 96.3.965]

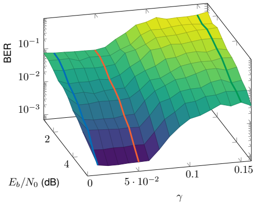


(b) $(3, 6)$-regular LDPC code with $n = 204, k = 102$ [Mac23, 204.33.484]

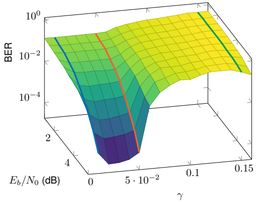


(c) $(3, 6)$-regular LDPC code with $n = 408, k = 204$ [Mac23, 408.33.844]

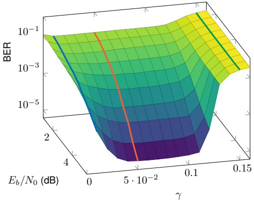




(d) BCH code with $n = 31, k = 26$



(e) $(5, 10)$-regular LDPC code with $n = 204, k = 102$ [Mac23, 204.55.187]



(f) LDPC code (Progressive Edge Growth Construction) with $n = 504, k = 252$ [Mac23, PEGReg252x504]

Proximal Decoding: Frame Error Rate

- Analysis of simulated[1] BER and FER

$s \leftarrow 0$

for K iterations **do**

  $r \leftarrow s - \omega \nabla L(y \mid s)$

  $s \leftarrow r - \gamma \nabla h(r)$

  $\hat{x} \leftarrow \text{sign}(s)$

  if $H\hat{c} = 0$ **do**

    return $\hat{c}$

  end if

end for

return $\hat{c}$

Figure: Proximal decoding algorithm [WT22]

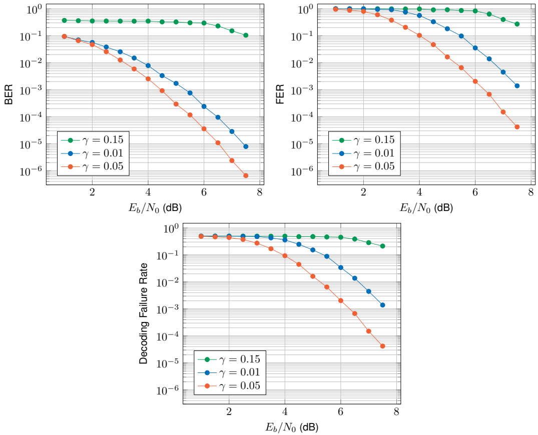


Figure: Simulation results for $\omega = 0.05, K = 100$

[1] (3,6) regular LDPC code with $n = 204$, $k = 102$ [Mac23, 204.33.484]

Proximal Decoding: Oscillation of Estimate

- $\nabla L(\boldsymbol{y} \mid \boldsymbol{x})$ and $\nabla h(\boldsymbol{x})$ generally end up in an equilibrium

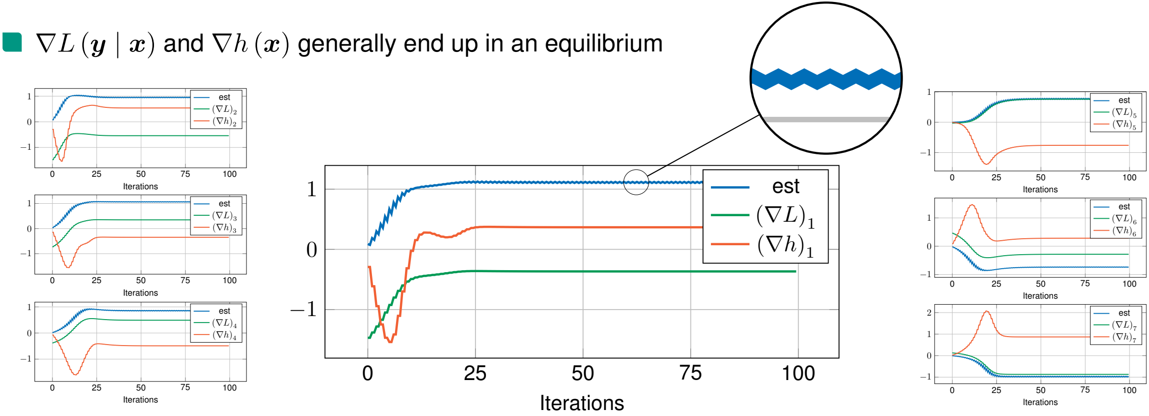
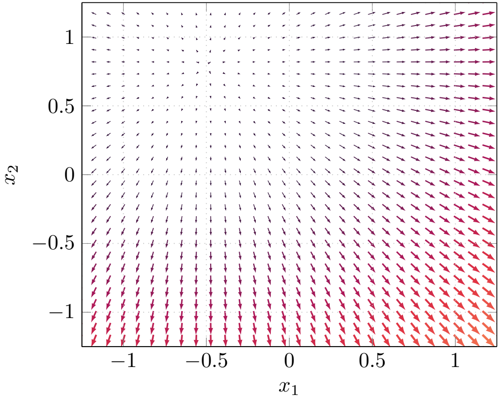


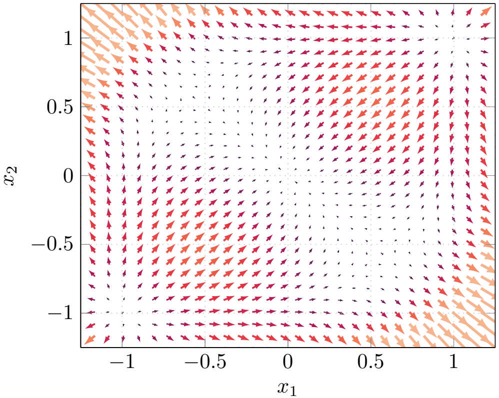
Figure: Internal variables of proximal decoder as a function of the number of iterations ($n = 7$)[1]

[1] A single decoding is shown, using the BCH$(7, 4)$ code; $\gamma = 0.05, \omega = 0.05, E_b/N_0 = 5\,$dB

Proximal Decoding: Visualization of Gradients



(a) $\nabla L\left(\boldsymbol{y} \mid \boldsymbol{x}\right)$ for a repetition code with $n = 2$ [1]

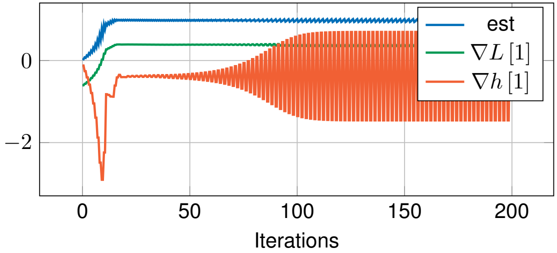


(b) $\nabla h\left(\boldsymbol{x}\right)$ for a repetition code with $n = 2$

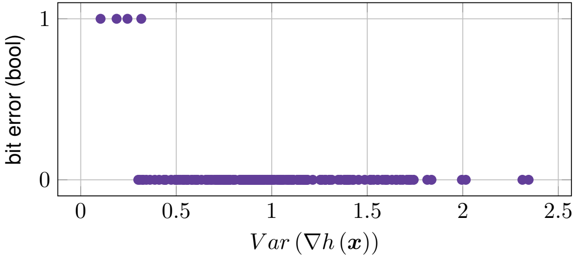
[1] In an AWGN Channel $\nabla L\left(\boldsymbol{y} \mid \boldsymbol{x}\right) \propto \left(\boldsymbol{x} - \boldsymbol{y}\right)$ [WT22, Sec. 4.1]

Proximal Decoder: Oscillation of $\nabla h(\boldsymbol{x})$

- For larger n, the gradient itself starts to oscillate
- The amplitude of the oscillation seems to be highly correlated with the probability of a bit error



(a) Internal variables of proximal decoder as a function of the iteration ($n = 204$)[1]



(b) Correlation between bit error and amplitude of oscillation

[1] A single decoding is shown, using a (3,6) regular LDPC code with $n = 204$, $k = 102$ [Mac23, 204.33.484]; $\gamma = 0.05$, $\omega = 0.05$, $E_b/N_0 = 5$ dB

Proximal Decoding: Improvement using “ML-on-List”

- Improvement of proximal decoding by adding an “ML-on-list” step after iterating

$\boldsymbol{s} \leftarrow \boldsymbol{0}$
for K iterations **do**
    $\boldsymbol{r} \leftarrow \boldsymbol{s} - \omega \nabla L \left(\boldsymbol{y} \mid \boldsymbol{s} \right)$
    $\boldsymbol{s} \leftarrow \boldsymbol{r} - \gamma \nabla h \left(\boldsymbol{r} \right)$
    $\hat{\boldsymbol{x}} \leftarrow \text{sign} \left(\boldsymbol{s} \right)$
    if $\boldsymbol{H} \hat{\boldsymbol{c}} = 0$ **do**
        return $\hat{\boldsymbol{c}}$
    end if
end for
return $\hat{\boldsymbol{c}}$

Figure: Proximal decoding algorithm [WT22]

$\boldsymbol{s} \leftarrow \boldsymbol{0}$
for K iterations **do**
    $\boldsymbol{r} \leftarrow \boldsymbol{s} - \omega \nabla L \left(\boldsymbol{y} \mid \boldsymbol{s} \right)$
    $\boldsymbol{s} \leftarrow \boldsymbol{r} - \gamma \nabla h \left(\boldsymbol{r} \right)$
    $\hat{\boldsymbol{x}} \leftarrow \text{sign} \left(\boldsymbol{s} \right)$
    if $\boldsymbol{H} \hat{\boldsymbol{c}} = \boldsymbol{0}$
        return $\hat{\boldsymbol{c}}$
    end if
end for
Find N most probably wrong bits.
Generate variations $\tilde{\boldsymbol{c}}_i$ of $\hat{\boldsymbol{c}}$ with the N bits modified.
Compute $d_H \left(\tilde{\boldsymbol{c}}_i, \hat{\boldsymbol{c}} \right)$ for all valid codewords $\tilde{\boldsymbol{c}}_i$
Output $\tilde{\boldsymbol{c}}_i$ with lowest $d_H \left(\tilde{\boldsymbol{c}}_i, \hat{\boldsymbol{c}} \right)$

Figure: Improved proximal decoding algorithm

Proximal Decoding: Improvement using “ML-on-List”

- Comparison of proximal & improved (correction of $N = 12$ bit) decoding simulation[1] results

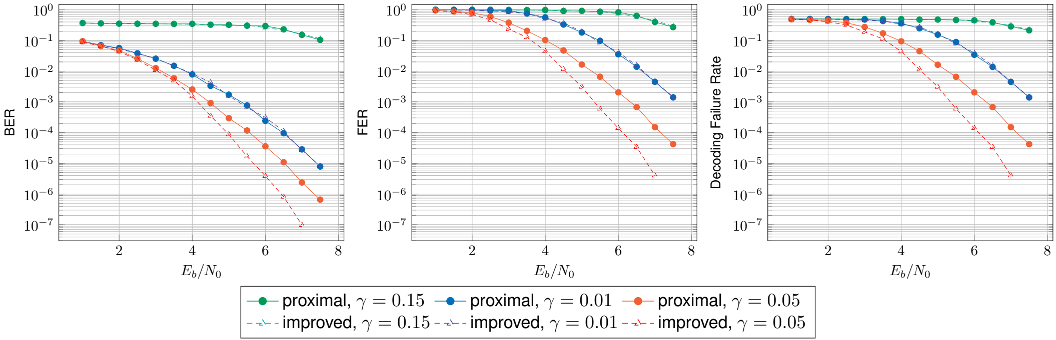
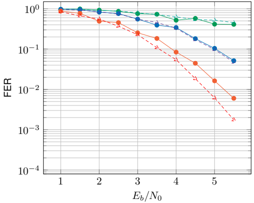


Figure: Simulation results for $\gamma = 0.05, \omega = 0.05, K = 200, N = 12$

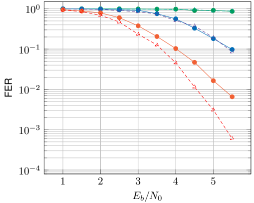
[1] (3,6) regular LDPC code with $n = 204$, $k = 102$ [Mac23, Code: 204.33.484]

Proximal Decoding: Improvement using “ML-on-List”

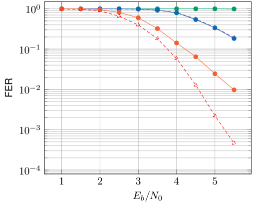
Karlsruhe Institute of Technology



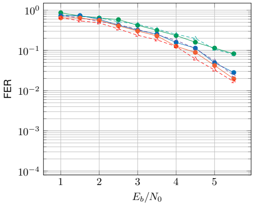
(a) $(3,6)$-regular LDPC code with $n = 96, k = 48$ [Mac23, 96.3.965]



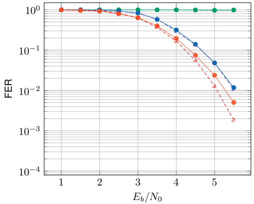
(b) $(3,6)$-regular LDPC code with $n = 204, k = 102$ [Mac23, 204.33.484]



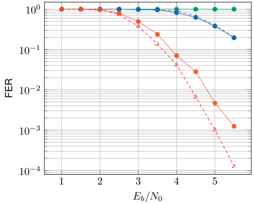
(c) $(3,6)$-regular LDPC code with $n = 408, k = 204$ [Mac23, 408.33.844]



(d) BCH code with $n = 31, k = 26$



(e) $(5,10)$-regular LDPC code with $n = 204, k = 102$ [Mac23, 204.55.187]



(f) LDPC code (Progressive Edge Growth Construction) with $n = 504, k = 252$ [Mac23, PEGReg252x504]

Proximal Decoding: Average Error

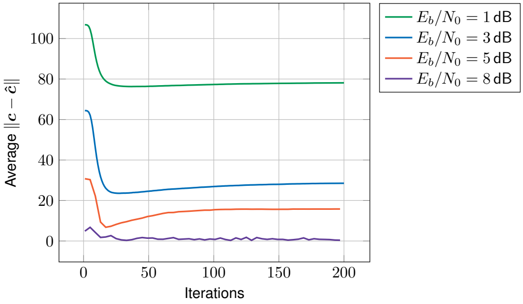


Figure: Average error for $500\,000$ decodings, $\omega = 0.05, \gamma = 0.05, K = 200$[1]

- With increasing iterations, the average error asymptotically approaches a minimum, non-zero value

[1] Simulation performed with (3,6) regular LDPC code with $n = 204, k = 102$ [Mac23, Code: 204.33.484]

Proximal Decoding: Time Complexity

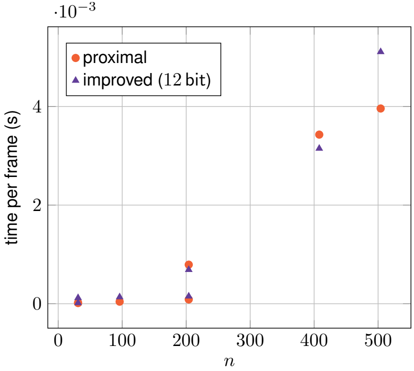


Figure: Time Complexity of Proximal Decoding and Modified Implementation[2]

[2] The points shown were calculated by evaluating the metadata of BER simulation results from the following codes: BCH $(31, 11)$; BCH $(31, 26)$; [Mac23, 96.3.965; 204.33.484; 204.55.187; 408.33.844; PEGReg252x504]

Conclusion

Karlsruhe Institute of Technology

- Analysis of proximal decoding for AWGN channels:
  - Error coding performance (BER, FER, decoding failures)
  - Computational performance ($\mathcal{O}(n)$ time complexity, fast implementation possible)
  - Number of iterations independent of SNR
- Suggestion for improvement of proximal decoding:
  - Addition of “ML-on-list” step
  - Up to $\sim 1\,\mathrm{dB}$ gain under certain conditions

Overview

Karlsruhe Institute of Technology

- Theoretical Background
  - Motivation
  - Presumptions
  - Optimization as a Decoding Method
- Decoding Algorithms
  - Proximal Decoding
  - LP Decoding
- Analysis
  - Proximal Decoding
- Forthcoming Analysis
  - LP Decoding

Forthcoming Analysis

- Test ADMM (Alternating Direction Method of Multipliers) as an optimization method for LP Decoding
  - In LP decoding, the ML decoding problem is reduced to a linear program, which can be solved in polynomial time [Gen+20]
  - ADMM is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers [Boy+11]
  - ADMM has been proposed for efficient LP Decoding [ZS13]
- Compare ADMM implementation with Proximal Decoding implementation with respect to
  - decoding performance (BER, FER)
  - computational performance (time complexity, actual seconds per frame)

Questions

Thank you for your attention!

Any questions?

Bibliography

[BMT78] E. Berlekamp, R. McEliece, and H. van Tilborg. "On the inherent intractability of certain coding problems (Corresp.)". In: *IEEE Transactions on Information Theory* 24.3 (May 1978), pp. 384–386.

[Boy+11] Stephen Boyd et al. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. 2011.

[Fel03] Jon Feldman. "Decoding error-correcting codes via linear programming". PhD thesis. MIT, 2003.

[FWK05] J. Feldman, M.J. Wainwright, and D.R. Karger. "Using linear programming to Decode Binary linear codes". In: *IEEE Transactions on Information Theory* 51.3 (2005), pp. 954–972.

[Gen+20] Florian Gensheimer et al. "A Reduced-Complexity Projection Algorithm for ADMM-Based LP Decoding". In: *IEEE Transactions on Information Theory* 66.8 (2020), pp. 4819–4833.

[KTP19] Banu Kabakulak, Z. Caner Taşkın, and Ali Emre Pusane. "Optimization–based decoding algorithms for LDPC convolutional codes in communication systems". In: *IISE Transactions* 51.10 (2019), pp. 1061–1074.

Bibliography

[Lu23] Yanxia Lu. “Realization of Channel Decoding Using Optimization Techniques”. Bachelor’s Thesis. KIT, 2023.

[Mac23] David J.C. MacKay. *Encyclopedia of Sparse Graph Codes*. Jan. 2023. URL: http://www.inference.org.uk/mackay/codes/data.html.

[PB14] Neal Parikh and Stephen Boyd. “Proximal Algorithms”. In: *Found. Trends Optim.* 1.3 (Jan. 2014), pp. 127–239.

[WT22] Tadashi Wadayama and Satoshi Takabe. “Proximal Decoding for LDPC Codes”. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* advpub (2022), 2022TAP0002.

[ZS13] Xiaojie Zhang and Paul H. Siegel. “Efficient iterative LP decoding of LDPC codes with alternating direction method of multipliers”. In: *2013 IEEE International Symposium on Information Theory*. 2013, pp. 1501–1505.