MCMC Decoding of LDPC Codes with BP Preprocessing

Jiun-Ting Huang and Young-Han Kim
Department of Electrical and Computer Engineering
University of California, San Diego

***Abstract*—Monte Carlo Markov chain (MCMC) decoding is a randomized algorithm which has been proven to be near-optimal in terms of decoding error probability. However, the exponentially slow mixing rate of Markov chains seems to preclude MCMC decoding from applications concerning even short blocklength codes. In contrast, belief propagation (BP) is a deterministic algorithm that is empirically fast but sub-optimal in error rate when it is used to decode low-density parity-check (LDPC) codes. In this paper, a code-independent BP–MCMC hybrid decoder is devised for short-blocklength LDPC codes. Theoretical error analysis of the hybrid algorithm is provided. Preliminary experiments show that the preprocessing of BP successfully reduces the time complexity of MCMC decoding and hence significantly improves the applicability of MCMC decoders to short LDPC codes.**

I. Introduction

Consider the channel decoding problem in which we wish to infer the transmitted message $\mathbf{m} \in \mathbb{F}_2^k$, which was encoded by a given code $\mathbf{x} : \mathbb{F}_2^k \to \mathcal{X}^n$, $\mathbf{x}(\mathbf{m}) = (x_1, \ldots, x_n) \in \mathcal{X}^n$, from an output sequence $\mathbf{y} = (y_1, \ldots, y_n)$ of a memoryless communication channel $p(\mathbf{y}|\mathbf{x})$. The performance of any decoding mapping (deterministic or random) $\widehat{\mathbf{M}} : \mathcal{Y}^n \to \mathbb{F}_2^k$ is evaluated by the average probability of decoding error:

$$p_e = \mathsf{P}(\mathbf{M} \neq \widehat{\mathbf{M}}(\mathbf{Y})). \tag{1}$$

A recent discovery founded on the Liu–Cuff–Verdú (LCV) lemma [1]–[3] asserts that the *randomized likelihood* (RL) decoding algorithm, which generates the estimate by sampling from the posterior

$$\widehat{\mathbf{M}}_{\mathrm{RL}}(\mathbf{y}) \sim p(\mathbf{m}|\mathbf{y}), \tag{2}$$

has error probability bounded by twice of the best achievable value from the *maximum a posteriori probability* (MAP) decoding rule

$$\mathsf{P}(\mathbf{M} \neq \widehat{\mathbf{M}}_{\mathrm{RL}}(\mathbf{Y})) \leq 2p_e^*, \tag{3}$$

where

$$p_e^* = \arg\max_{\mathbf{m}} p(\mathbf{m}|\mathbf{y}). \tag{4}$$

Since most good codes operate at $p_e^* \ll 1$, the factor of two here is essentially negligible.

Given the strong performance guarantee in (3), many algorithms based on Monte Carlo Markov chain (MCMC) methods were proposed for practical implementation of RL decoders. In 2001, Radford Neal developed a decoding algorithm for low-density parity-check (LDPC) codes based on Gibbs sampling, a classical MCMC method [4]. Then, the asymptotic decoding performance of the Gibbs decoder was compared to that of a belief propagation (BP) decoder in the textbook by Mezard and Montanari [5]. Besides Gibbs sampling, the use of other MCMC techniques such as Metropolis algorithm was also studied in our previous work [6].

In decoding LDPC codes, it is known that Gibbs decoders have a message-passing representation, which is structurally similar to that of the conventional BP decoding. Both involve iterative "message passing" between check nodes and variable nodes. There are also major differences, which make BP more appealing for LDPC codes (at the moment), but not a complete winner:

- BP is deterministic, fast (low-complexity), suboptimal (getting trapped and not converging to the MAP performance), and soft (producing log likelihood ratios).
- Gibbs is random, slow (high-complexity), near-optimal (eventually converging to the MAP performance), and hard (producing bit estimates).

These observations motivated us to combine the complementary strengths (low-complexity and accuracy, in particular) to design a hybrid of the two. In an earlier study [7], Ahn et al. developed such a BP–Gibbs hybrid method to approximate a partition function of a graphical model, whereby BP is run first to approximate the partition function and then Gibbs is used to correct the approximation error.

Although the objective of approximating partition functions is quite different from our decoding application, the high-level idea still transcends to our case. Thus inspired, we developed a hybrid decoding scheme in which BP decoding is used as a preprocessor to Gibbs (or any Monte Carlo decoding for that matter). Our preliminary simulations show that the hybrid algorithm outperforms both the pure BP and the pure Gibbs decoder. As a variant of an MCMC decoder, the preprocessing of BP also successfully reduces the time complexity of the Gibbs decoding algorithm. It is also fortuitous that the downside of Gibbs in producing hard bits no longer exists, since it is used in the second (and final) stage.

The rest of the paper is organized as follows. In Section II we propose a BP–MCMC hybrid decoder for LDPC codes and provide a preliminary theoretical analysis of its performance. Then, a key step in our algorithm, which we call the subset selection step, is described in details in Section III. Numerical experiment results are presented in Section IV. Finally, Secion V concludes the paper.

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

II. BP–MCMC HYBRID DECODING FOR LDPC CODES

LDPC codes are commonly decoded using iterative algorithms operating over a factor graph (e.g., the sum-product algorithm [8]), where soft messages like bit-wise log-likelihood ratios (LLRs) are passed between variable nodes and check nodes. After a maximum number $T_{\mathrm{BP,max}}$ of iterations being performed, the BP decoder produces an estimate of the input codeword if the bit-wise hard decisions satisfy all the parity-check constraints, or declares decoding failure if otherwise. Thus, a decoding error occurs when some of the LLRs converge to the opposite polarity or fail to converge.

To correct the BP decoding error, we devise a simple, MCMC-based method. Given the channel output $\mathbf{y}$, we first run BP to obtain the LLRs for each message bit. We then identify the set $\mathcal{I}$ of wrongly decoded message bits. Finally we employ the MCMC method to re-decode the bits in $\mathcal{I}$ by generating a random sample from the conditional posterior as

$$\widehat{\mathbf{M}}_{\mathrm{MCMC}} \sim p(\mathbf{m}|\mathbf{y}, \mathbf{m}_{\neg\mathcal{I}} = \widehat{\mathbf{M}}_{\mathrm{BP},\neg\mathcal{I}}), \tag{5}$$

where $\neg\mathcal{I}$ denotes the complementary set of $\mathcal{I}$ in $[k]$. If BP works well, the effective blocklength $|\mathcal{I}|$ is much smaller than the dimension of the code k, which lets MCMC under its sweet spot.

This naive hybrid scheme does not work as is, however. The problem of precisely identifying the *trapping set*, the set of bits that are not correctly decoded by the BP decoder, is in fact NP-hard [9]. There is still a silver lining if we target the low error rate region of LDPC codes, where the block error rate of BP decoders is contributed only by a small trapping set [10], [11]. Inspired by some existing post-processing approaches [12]–[14] for mitigating the error-floor problem, we thus propose three heuristic subset selection methods for capturing the small culprit trapping set (cf. Sec. III).

Provided that a reliable subset selection scheme exists, it can then be shown that, under some mild conditions, our hybrid decoding is able to achieve the decoding error probability comparable to that of optimal decoding. Let $\mathcal{I}$ be the selected subset after BP. The error event $\mathcal{E} := \{\widehat{\mathbf{M}}_{\mathrm{BP-MCMC}} \neq \mathbf{M}\}$ of the hybrid decoding method can be easily expressed as

$$\mathcal{E} = \mathcal{E}_1 \cup (\mathcal{E}_2 \backslash \mathcal{E}_1),$$

where

$$\mathcal{E}_1 = \left\{\widehat{\mathbf{M}}_{\mathrm{BP},\neg\mathcal{I}} \neq \mathbf{M}_{\neg\mathcal{I}}\right\}$$

is the subset selection error, and

$$\mathcal{E}_2 = \left\{\widehat{\mathbf{M}}_{\mathrm{MCMC}} \neq \mathbf{M}\right\}$$

is the MCMC decoding error. Note that since the BP algorithm is deterministic, the set $\mathcal{I}$ and the estimate $\widehat{\mathbf{M}}_{\mathrm{BP}}$ are functions of the channel output $\mathbf{Y}$. Suppose that the output of the MCMC decoder $\widehat{\mathbf{M}}_{\mathrm{MCMC}}$ is perfectly sampled according to the conditional posterior $p\left(\mathbf{m}\middle|\mathbf{Y}, \mathbf{m}_{\neg\mathcal{I}} = \widehat{\mathbf{M}}_{\mathrm{BP},\neg\mathcal{I}}(\mathbf{Y})\right)$. Then, an application of LCV lemma yields

$$\begin{aligned}\mathsf{P}(\mathcal{E}) &= \mathsf{P}(\mathcal{E}_1) + \mathsf{P}(\mathcal{E}_2\backslash\mathcal{E}_1)\\ &= \mathsf{P}(\mathcal{E}_1) + \mathsf{P}(\mathcal{E}_1^c)\,\mathsf{P}(\mathcal{E}_2|\mathcal{E}_1^c)\\ &= \mathsf{P}(\mathcal{E}_1) + \mathsf{P}(\mathcal{E}_1^c)\,\mathsf{E}[\mathsf{P}(\mathcal{E}_2|\mathbf{Y}, \mathcal{E}_1^c)]\\ &\leq \mathsf{P}(\mathcal{E}_1) + 2\,\mathsf{P}(\mathcal{E}_1^c)\,\mathsf{E}[\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)],\end{aligned} \tag{6}$$

where

$$\mathbf{M}^*(\mathbf{Y}) = \arg\max_{\mathbf{m}} p\left(\mathbf{m}\middle|\mathbf{Y}, \mathbf{m}_{\neg\mathcal{I}} = \widehat{\mathbf{M}}_{\mathrm{BP},\neg\mathcal{I}}(\mathbf{Y})\right),$$

is the MAP estimate of the conditional posterior. Let

$$E_{\mathbf{Y}} = \{\mathbf{m} \in \{0,1\}^k : \mathbf{m}_{\neg\mathcal{I}} = \widehat{\mathbf{M}}_{\mathrm{BP},\neg\mathcal{I}}(\mathbf{Y})\}$$

denote the space from which the MCMC decoder generates the sample. Then, we can write

$$\begin{aligned}&\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)\\ &= \frac{\sum_{\mathbf{m}\in E_{\mathbf{Y}}, \mathbf{m}\neq\mathbf{M}^*(\mathbf{Y})} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}{\sum_{\mathbf{m}\in E_{\mathbf{Y}}} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}\\ &= \frac{\sum_{\mathbf{m}} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}{\sum_{\mathbf{m}\in E_{\mathbf{Y}}} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})} \frac{\sum_{\mathbf{m}\in E_{\mathbf{Y}}, \mathbf{m}\neq\mathbf{M}^*(\mathbf{Y})} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}{\sum_{\mathbf{m}} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}\\ &= \frac{1}{\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})} \frac{\sum_{\mathbf{m}\in E_{\mathbf{Y}}, \mathbf{m}\neq\mathbf{M}^*(\mathbf{Y})} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}{\sum_{\mathbf{m}} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}.\end{aligned} \tag{7}$$

Note that if the MAP estimate $\widehat{\mathbf{M}}_{\mathrm{MAP}}(\mathbf{Y})$ lies in the set $E_{\mathbf{Y}}$, it must equal to $\mathbf{M}^*(\mathbf{Y})$. Therefore, it follows that

$$\begin{aligned}\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c) &\leq \frac{\sum_{\mathbf{m}\neq\widehat{\mathbf{M}}_{\mathrm{MAP}}(\mathbf{Y})} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}{\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})\sum_{\mathbf{m}} p(\mathbf{Y}|\mathbf{M}=\mathbf{m})}\\ &= \frac{\mathsf{P}(\widehat{\mathbf{M}}_{\mathrm{MAP}}(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y})}{\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})}.\end{aligned} \tag{8}$$

Now, suppose that the random variables $\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})$ and $\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)$ are "nearly positively correlated", that is, there exists some constant $\epsilon > 0$ such that

$$\begin{aligned}&\mathsf{E}[\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})]\,\mathsf{E}[\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)]\\ &\leq (1+\epsilon)\,\mathsf{E}[\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})\,\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)].\end{aligned} \tag{9}$$

Combining (6), (8), and (9) yields

$$\begin{aligned}\mathsf{P}(\mathcal{E}) &\leq \mathsf{P}(\mathcal{E}_1) + 2\,\mathsf{E}[\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})]\,\mathsf{E}[\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)]\\ &\leq \mathsf{P}(\mathcal{E}_1) + 2(1+\epsilon)\,\mathsf{E}[\mathsf{P}(\mathcal{E}_1^c|\mathbf{Y})\,\mathsf{P}(\mathbf{M}^*(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y}, \mathcal{E}_1^c)]\\ &\leq \mathsf{P}(\mathcal{E}_1) + 2(1+\epsilon)\,\mathsf{E}[\mathsf{P}(\widehat{\mathbf{M}}_{\mathrm{MAP}}(\mathbf{Y}) \neq \mathbf{M}|\mathbf{Y})]\\ &= \mathsf{P}(\mathcal{E}_1) + 2(1+\epsilon)p_e^*.\end{aligned} \tag{10}$$

Hence, as long as the state selection error is well controlled, the performance of the hybrid decoder only deviates from the optimal MAP decision by constant factors.

Although the nearly positive correlation condition in (9) seems arbitrary, it holds conceptually when the code operates in the low error-rate region, where the true message $\mathbf{M}$, the MAP estimate $\widehat{\mathbf{M}}_{\mathrm{MAP}}$, and the conditional MAP estimate $\mathbf{M}^*$ are most likely the same. Our simulation result confirms that the subset selection error indeed dominates the error events (cf. Sec IV).

III. Subset Selection Methods

In this section, we describe three subset selection methods. Typically if the BP decoding failure is caused by small trapping sets, the bit LLRs are usually characterized by two types of behaviors: 1) false convergence, i.e., the LLRs reach a fixed false state and stay unchanged for the following iterations; 2) oscillation, i.e., some of the LLRs oscillate among a set of states periodically. Based on this observation, we proposed the following subset selection methods.

1) *LLR thresholding (LLR-TH).* In BP decoding, the LLR of a large number of variable nodes converges to a strong belief within very few iterations. Intuitively, these nodes are already reliably decoded so they can be removed from the subsequent decoding process [12]. Thus we set $\mathcal{I} = \{i \in [k] : |LLR_i| \leq \eta\}$, where LLR_i is the i-th bit LLR and η is a preset threshold.
2) *LLR thresholding and oscillation (LLR-OSC).* The LLR value of bits in the trapping set tends to oscillate about 0. Hence, we select message bit $i \in [k]$ if the sign of LLR_i oscillates too much or LLR_i is below a preset threshold. Clearly, the set thus selected is always a superset of the one selected by the LLR thresholding method.
3) *Unsatisfied check nodes (UN-CHK).* If the decoding failure is caused by a small trapping set, then the set of unsatisfied check nodes is expected to be small as well. Following an idea in the bit flipping algorithm by Gallager [15], we select all the neighboring variable nodes of the unsatisfied check nodes.

Note that these subset selection methods have marginal computational overhead. The thresholding and oscillation conditions can be examined during the BP iterations. To determine whether the LLR value of the message bits has too much oscillation, it does not have to store all the signs of the LLRs. Instead, only one counter is needed for each message bit to record the number of sign changes. Also, finding the neighboring variable nodes of the unsatisfied check nodes only requires one additional round of message-passing using the hard decisions.

IV. Experiment Results

For experimental purposes, the performance of the hybrid decoder is compared with the pure BP decoder using randomly generated irregular LDPC codes[1] of block length $n = 200$ and rates $1/4$ ($k = 50$), $3/8$ ($k = 75$), and $1/2$ ($k = 100$) transmitted though different binary symmetric channels (BSCs) or additive white Gaussian noise (AWGN) channels. For the LLR-TH and the LLR-OSC methods, the value of the threshold is heuristically chosen to be 3.66 for maximizing the probability of correct subset selection.

For the MCMC decoding stage, we chose the annealed block Gibbs (ABG) decoder, since it has the fastest convergence in the family of Gibbs decoders we previously studied [6]. The general ABG decoding algorithm can be summarized in the following steps.

- Initialize a starting point $\mathbf{m}^{(0)} = (m_1^{(0)}, m_2^{(0)}, \ldots, m_k^{(0)})$.
- On the t-th iteration ($t = 1, 2, \ldots$),
  - ▷ randomly pick a block of s coordinates $B \subset [k]$.
  - ▷ sample the coordinates in B according to the annealed conditional marginal distribution $\mathbf{M}_B^{(t)} \sim p^{\alpha}(\mathbf{m}_B | \mathbf{y}, \mathbf{m}_{\neg B}^{(t-1)})$.
  - ▷ fix the remaining coordinates $m_j^{(t)} = m_j^{(t-1)}$ for all $j \in [k] \backslash B$.

After the preprocessing of BP, the ABG decoder is initialized by the message bits of the BP output $\widehat{\mathbf{M}}_{\mathrm{BP}}^k$. Then, at each iteration, the set of coordinates B to be sampled is chosen randomly from the selected subset of variable nodes $\mathcal{I} \subset [k]$. In our experiment, we set the annealing parameter $\alpha = 0.5$ and blocksize $s = 10$. The maximal numbers of iterations for the BP and MCMC steps are 20 and 1000, respectively.

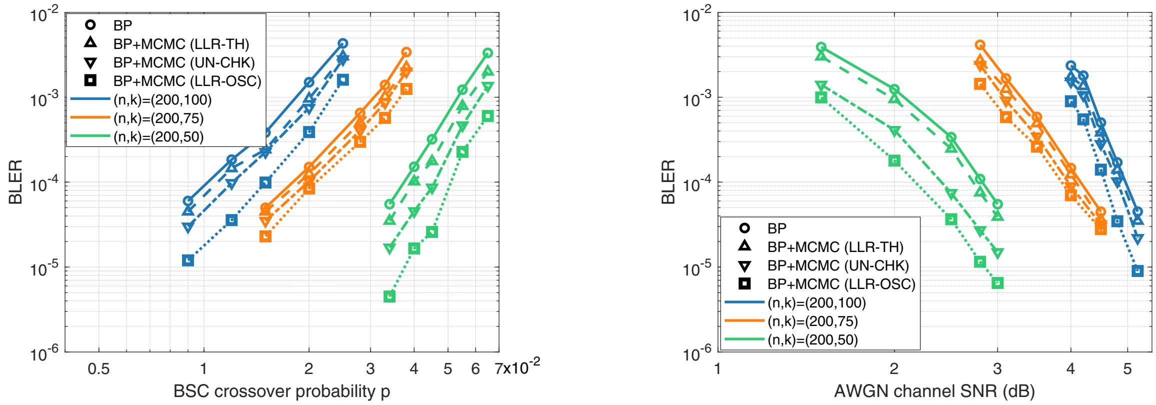
In Figures 1(a) and 1(b), we compare the block error rate (BLER) performance of the proposed hybrid decoders. The subset selection scheme based on LLR thresholding and oscillation performed uniformly better than the other two and pure BP, achieving one order of magnitude improvement over pure BP in the best case. Simulations also show that the LLR-OSC subset selection scheme has a uniformly higher rate of successful selection (i.e., the selected subset contains the trapping set of the given channel output) of $\sim 80\%$, compared to $\sim 60\%$ for the UN-CHK method and $\sim 30\%$ for the LLR-TH method.

One interesting observation from these simulations is the success in reducing the dimension of the state space of the Markov chain via the subset selection step. Extrapolating from the toy example of the $(40, 20)$ LDPC code investigated earlier in [6], the estimated number of iterations required by the annealed block Gibbs decoder (without BP preprocessing) to converge would be of order 10^{20} to 10^{25} for 100-bit messages. With the aid of dimension reduction, however, the BLER improvement happens within merely 1000 iterations. Figure 2(a) and 2(b) display part of the statistics of subset sizes. We notice that since the LLR-OSC method has the best error rate performance, the average size of the subsets it selects may be close the effective dimension of the state space of the Markov chain. We also note that the speedup in convergence may be not solely due to the smaller state space, but also due to the better initialization of the Markov chain by BP preprocessing.

Another notable remark from these experiments is the dominance of the subset selection error over the decoding error events of the hybrid method. Table I lists the percentage of decoding error contributed by each of the proposed subset selection schemes. Therefore, investigating how to design better subset selection algorithms will be a future topic of research in this thrust.

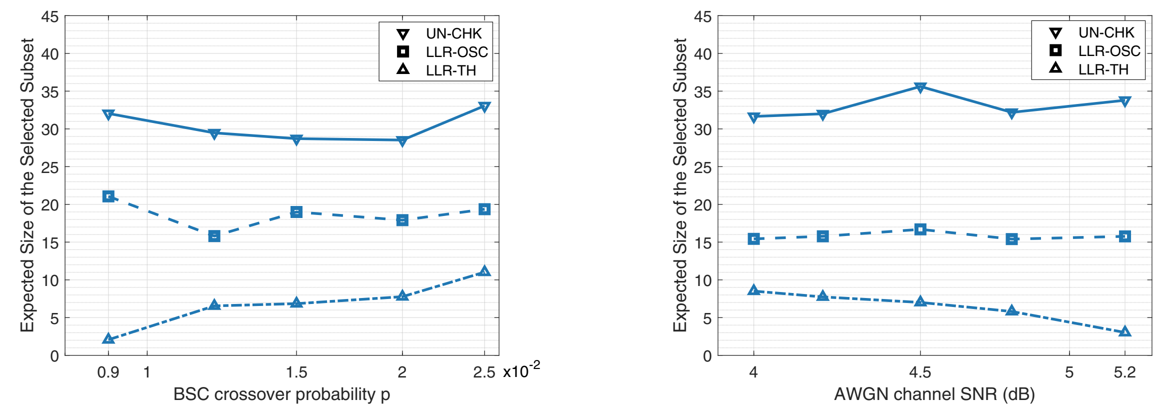
Finally, we study how the performance of the proposed hybrid decoding algorithm scales with the block length. To this

[1]The LDPC codes were generated using the software developed by Radford Neal, which can be found on his website https://www.cs.toronto.edu/∼radford/ldpc.software.html. Each code generated has a fixed variable node degree of 3 and no 4-cycles.



(a) BSCs **(b)** AWGN channels

Fig. 1. BLER performance of BP–ABG hybrid decoders for irregular LDPC codes of length 200.



(a) BSCs **(b)** AWGN channels

Fig. 2. Expected size of the selected subset for an irregular $(200, 100)$ LDPC code.

end, we perform experiment on randomly generated irregular LDPC codes of rate $1/2$ and block lengths $n = 100$, 200, and 500. The LLR-OSC method is chosen for the subset selection. The other parameter settings are kept the same. The results are shown in Figure 3(a) and 3(b). From Figure 4(a) and 4(b), we can observe the convergence of the ABG decoder for codes of block lengths $n = 100$ and 200. However, for the $(500, 250)$ code the Markov chain does not seem to reach the convergence within 1000 iterations. According to our simulation results, the effective dimension of the state space grows approximately linearly with the block length. This implies that the proposed algorithm may have exponential time complexity.

V. Concluding Remarks

In this paper, we proposed a BP-MCMC hybrid decoding method for decoding LDPC codes. Experimental results show that with the aid of BP preprocessing, the state space of the Markov chain can be effectively reduced. Hence, our approach successfully widen the application of MCMC decoding to LDPC codes of short block lengths (in the order of a few hundreds bits). However, the performance of the hybrid decoding algorithm heavily relies on the correctness of the subset selection step as well as the dimension of the code. Therefore, designing better subset selection methods and theoretical complexity analysis constitute interesting paths for further investigation.

References

[1] A. Bhatt, J. Huang, Y. Kim, J. J. Ryu, and P. Sen, “Variations on a theme by Liu, Cuff, and Verdú: The power of posterior sampling,” in *2018 IEEE Information Theory Workshop (ITW)*, Nov 2018, pp. 1–5.

[2] J. Liu, P. Cuff, and S. Verdú, “On α-decodability and α-likelihood decoder,” in *Proc. 55th Ann. Allerton Conf. Comm. Control Comput.*, Monticello, IL, Oct. 2017.

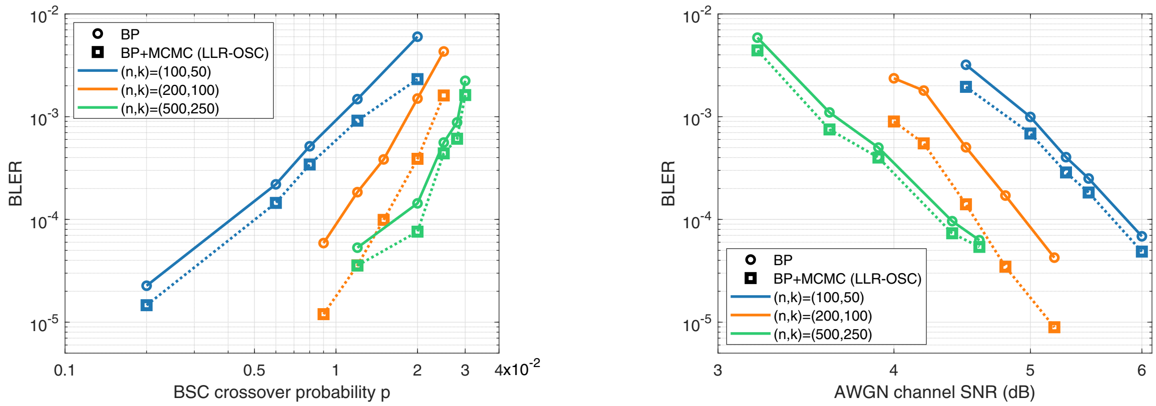
[3] S. Kudekar, S. Kumar, M. Mondelli, H. Pfister, and R. Urbanke, “Comparing the bit-MAP and block-MAP decoding thresholds of Reed-Muller codes on BMS channels,” in *Proc. IEEE Int. Symp. Inf. Theory*, Barcelona, Spain, July 2016, pp. 1755–1759.

[4] R. M. Neal, “Monte Carlo decoding of LDPC codes,” Dept. of Computer Science, University of Toronto, Tech. Rep., 2001, presented at *ICTP Works. Statis. Phys. Capacity-Approaching Codes*.

TABLE I

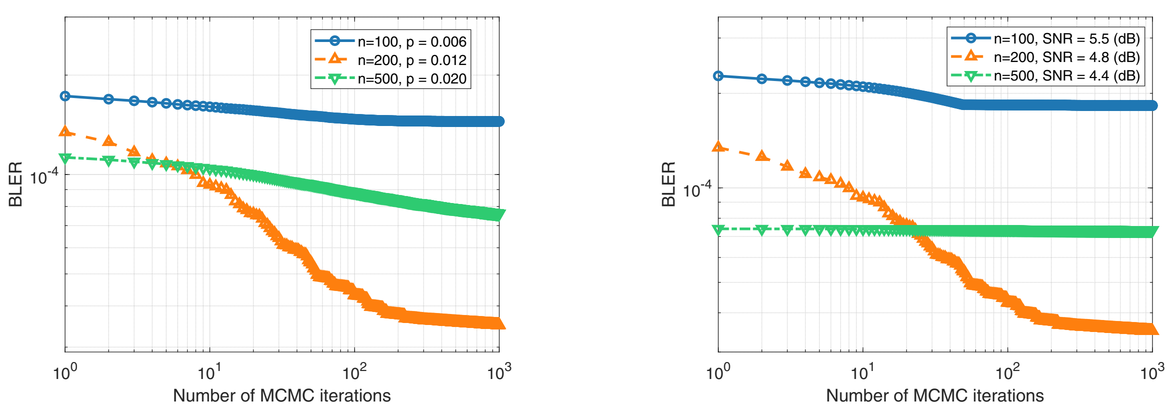
PERCENTAGE OF DECODING ERRORS CAUSED BY INCORRECT SUBSET SELECTION.

Channel	BSC			AWGN		
Subset selection method	LLR-TH	UN-CHK	LLR-OSC	LLR-TH	UN-CHK	LLR-OSC
Percentage (%)	99.39	76.06	61.75	99.12	76.06	76.34



(a) BSCs **(b)** AWGN channels

Fig. 3. BLER performance of the BP–ABG hybrid decoder for irregular LDPC codes of rate $1/2$ with different block lengths.



(a) BSCs **(b)** AWGN channels

Fig. 4. Convergence behavior of the ABG decoder for irregular LDPC codes of rate $1/2$ with different block lengths.

[5] M. Mezard and A. Montanari, *Information, Physics, and Computation*. Oxford University Press, 2009.

[6] A. Bhatt, J.-T. Huang, Y.-H. Kim, J. Ryu, and P. Sen, "Monte Carlo methods for randomized likelihood decoding," in *Proc. 56th Ann. Allerton Conf. Comm. Control Comput.*, Monticello, IL, Oct. 2018.

[7] S.-S. Ahn, M. Chertkov, and J. Shin, "Synthesis of MCMC and belief propagation," in *Adv. Neural Inf. Proc. Syst.*, 2016, pp. 1453–1461.

[8] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[9] A. McGregor and O. Milenkovic, "On the hardness of approximating stopping and trapping sets," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1640–1650, April 2010.

[10] T. Richardson, "Error floors of LDPC codes," in *2003 41st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2003, pp. 1426–1435.

[11] S. Ländner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, June 2005, pp. 630–635.

[12] E. Zimmermann, P. Pattisapu, and G. Fettweis, "Bit-flipping post-processing for forced convergence decoding of LDPC codes," in *2005 13th European Signal Processing Conference*, Sep 2005, pp. 1–4.

[13] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright, "Lowering ldpc error floors by postprocessing," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, Nov 2008, pp. 1–6.

[14] F. Leduc-Primeau, S. Hemati, S. Mannor, and W. J. Gross, "Dithered belief propagation decoding," *IEEE Transactions on Communications*, vol. 60, no. 8, pp. 2042–2047, August 2012.

[15] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.