Optimal Layered Scheduling for Hardware-Efficient Windowed Decoding of LDPC Convolutional Codes

Koike-Akino, T.; Draper, S.C.; Wang, Y.; Sugihara, K.; Matsumoto, W.; Millar, D.S.; Parsons, K.; Arlunno, V.; Kojima, K.

TR2016-124 September 2016

Abstract

We propose an optimal design method for layered scheduling in low-power windowed decoding of LDPC convolutional codes. Our optimal scheduling achieves up to a 70% complexity reduction and a 1 dB gain over conventional scheduling for limited decoding iterations.

European Conference on Optical Communication (ECOC)

Optimal Layered Scheduling for Hardware-Efficient Windowed Decoding of LDPC Convolutional Codes

Toshiaki Koike-Akino[1], Stark C. Draper[2], Ye Wang[1], Kenya Sugihara[3], Wataru Matsumoto[3], David S. Millar[1], Kieran Parsons[1], Valeria Arlunno[1], Keisuke Kojima[1]

[1] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA koike@merl.com
[2] Electrical & Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada
[3] Information Technology R&D Center, Mitsubishi Electric Corp., Ofuna, Kanagawa 247-8501, Japan

Abstract *We propose an optimal design method for layered scheduling in low-power windowed decoding of LDPC convolutional codes. Our optimal scheduling achieves up to a 70% complexity reduction and a 1 dB gain over conventional scheduling for limited decoding iterations.*

Introduction

Modern fiber-optic systems have used soft-decision decoding for capacity-approaching low-density parity-check (LDPC) codes[1–9]. In recent years, LDPC convolutional codes (LDPC-CC), also known as spatially coupled codes, have received a considerable amount of interest because of its saturation property[2] (asymptotic achievement of the maximum *a posteriori* probability bound). Moreover, LDPC-CCs are practically feasible with low-latency and low-memory windowed decoding (WD)[4–9].

In order to reduce the complexity of WD, Chang *et al.*[9] adopted layered scheduling[10,11] for LDPC-CC, achieving a 50% reduction in the required number of iterations. Layered scheduling[10,11] has been used in various practical communications systems for hardware-friendly LDPC decoding due to its flexible parallelism, memory efficiency, and faster convergence than the conventional flooding scheduling. However, most existing work has not discussed how to design layered scheduling, and instead uses naïve round-robin scheduling. Although convergence rates can be further improved by adaptively changing[12] the schedule according to the instantaneous reliability of the received signals, this may not be suitable for hardware implementation, due to the non-deterministic decoding order.

More recently, Schmalen *et al.*[4–6] achieved excellent performance by designing an irregular weight distribution (with maximum degree of 6) for LDPC-CC using only one-iteration WD with large window size (e.g., $W = 15$). However, a large window size and large degree can increase total complexity, and how to optimize the layered scheduling was not addressed.

To improve hardware efficiency, we optimize layered scheduling for 1-iteration WD with a small window size. We find a significant advantage in complexity reduction and in coding gain for optimized scheduling in WD, even for regular LDPC codes. Through threshold analysis, up to a 70% complexity reduction can be achieved by our optimized scheduling, when compared to the conventional flooding scheduling. For one iteration per window of size $W = 6$, a scheduling gain of more than 0.8 dB is obtained in bit-error rate (BER) performance for finite-length LDPC-CC with a maximum column weight of 4.

Layered scheduling design for WD

Fig. 1 depicts quasi-cyclic (QC) LDPC-CCs denoted by a protograph (J,K,L,M), where J is maximum column weight, K is maximum row weight, L is termination length of spacial coupling, and M is the QC lifting size. We use low-latency WD with window size W, for which WM consecutive check nodes (CNs) in total can be activated for every sliding window. Inside the window, we split the CNs into multiple layers for scheduling. Both the size of layers and stride of window sliding are set to equal the QC lifting parameter M.

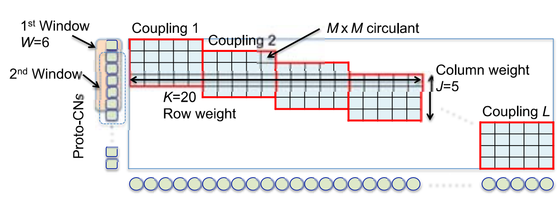

Fig. 1: WD with $W = 6$ for QC-LDPC-CC(J,K,L,M).

For comparison, we consider two conventional schedules: flooding and round-robin scheduling. For the flooding schedule in WD, all WM CNs inside the window propagate the belief messages in parallel, whereas every layer of M CNs is sequentially updated in a circular manner from the

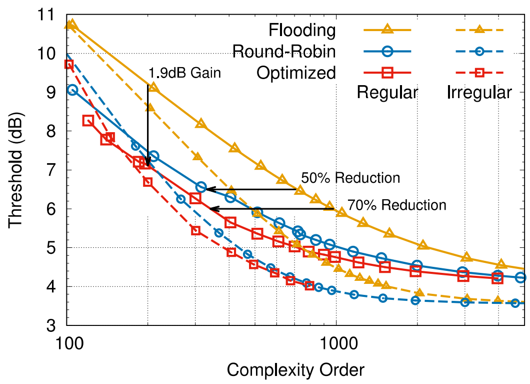

Fig. 2: Threshold vs. complexity for LDPC-CC$(4,20,6)$.

top to the bottom for the round-robin schedule. We optimize the irregular decoding order of the W layers inside the window to improve the decoding convergence speed without requiring any additional complexity. To do so, we use protograph-based extrinsic information transfer (P-EXIT)[13] chart analysis. With a P-EXIT chart, we can understand how much the reliability (i.e., mutual information) of the belief messages can be increased on average by updating each proto-CN. Since the search space of all possible scheduling orders is exponentially large, we use the M-algorithm to prune less promising candidates in the tree search. We use the averaged increase of extrinsic mutual information at proto-CNs as the optimization metric in the M-algorithm.

Threshold vs. complexity

We first discuss the threshold analysis via P-EXIT for a short-coupling LDPC-CC$(4,20,6,M)$. P-EXIT gives the asymptotic threshold assuming $M = \infty$. We consider both regular and irregular LDPC-CCs, whose degree distribution was designed to achieve the best threshold, assuming infinite-iteration non-windowed decoding.

Fig. 2 shows the achievable threshold as a function of the decoding complexity order, defined as the total number of updated belief messages, while varying the number of (sub-)iterations in layered scheduling. We used 500 survivors when optimizing the scheduling. Here, we consider the window size W equal to the total number of proto-CNs to analyze the impact of scheduling rather than the window size. We see from Fig. 2 that layered scheduling (i.e., round-robin and optimized) reduces the required complexity while achieving the same performance as the conventional flooding scheduling. For example, our optimized scheduling can reduce complexity by 70% over a wide range of target thresholds. Alternatively, up to a 1.9 dB threshold improvement can be achieved by the optimized scheduling at the same complexity.

Note that the scheduling gain is more significant in the low-complexity regime because the order of decoding layers does not matter if many iterations are available. It is interesting to note that irregular LDPC codes cannot always outperform regular LDPC codes. This may be because the irregular layered scheduling can work in an analogous fashion to irregular weight distribution even for regular codes, especially when the number of iterations is limited.

Scheduling gain in finite-length LDPC-CC

We now evaluate the optimized scheduling gain in BER performance for a finite-length girth-8 LDPC-CC$(4,20,20,384)$, with codeword length $KLM/J = 38{,}400$ bits, which is identical to a state-of-the-art practical LDPC code[3]. We optimized the QC permutations to maximize the girth by an algebraic method[14].

Fig. 3 shows the impact of the number of iterations per window, $I = 1, 2, 3$, for a relatively small window size of $W = 6$. Although the gain of optimized scheduling is marginal for $I = 3$ iterations, significant gains of more than 0.3 dB and 0.9 dB at a BER of 10^{-8} can be achieved, respectively, over the round-robin and flooding scheduling, for 1-iteration WD. Since the optimized schedule can selectively correct errors for less likely variable nodes, steeper BER curves are observed when compared to conventional scheduling. Consequently, the performance improvement is expected to be even greater at typical target BERs which are below 10^{-15}.

We can also reduce complexity by using smaller window sizes as the complexity is roughly proportional to W in WD. Fig. 4 plots BER performance for 1-iteration WD with different window sizes of $W = 3, 6, 9$. This figure shows that the optimized schedule offers a considerable gain irrespective of the window size. Note that the optimized schedule is more advantageous for larger window sizes because we can exploit more degrees of freedom in designing the decoding order to correct unreliable variable nodes.

Steady-state scheduling

Table 1 lists the optimized schedule for LDPC-CC$(4,20,20,M)$ with $W = 6$ and $I = 1$. The threshold of optimized scheduling in P-EXIT is 0.5 dB and 1.1 dB better than the round-robin and flooding scheduling, respectively. We found that the optimized schedule converges to a steady-

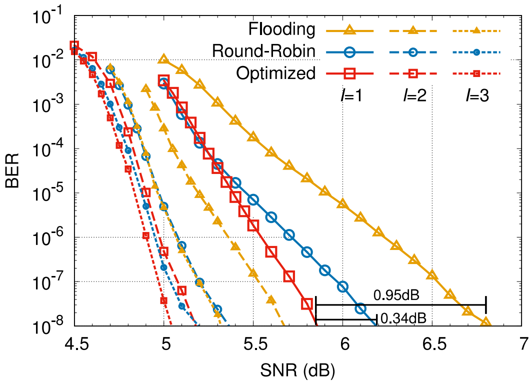

Fig. 3: Iteration count impact on BER with window size of $W = 6$ for LDPC-CC(4,20,20,384).

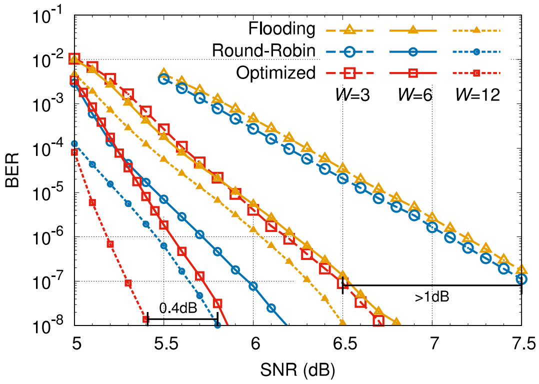

Fig. 4: Window size impact on BER with 1-iteration WD for LDPC-CC(4,20,20,384).

state decoding order (i.e., $6, 5, 4, 3, 6, 5$-th layers inside the window) in the 7–15th windows. Note that the first two layers in the window are not activated, while the last two layers are instead updated twice, in the steady-state scheduling. This irregular schedule leads to the increase in BER slope. In addition, such a steady-state scheduling benefits hardware implementation, in particular for a longer (or even infinite[4]) coupling parameter L.

For our cases with $J = 4$, $L = 20$, $W = 6$, and $I = 1$, there were no better irregular component matrices than the regular one. This result may potentially change if we consider larger J (such as 6 which has been used in the literature[4]), due to increased degrees of freedom in designing the weight irregularity.

Conclusions

We have proposed hardware-efficient and high-performance WD for LDPC-CC using irregular layered scheduling. A significant performance improvement of up to a 1 dB gain was achieved by our optimized scheduling for limited iterations. The required complexity to achieve the same threshold was also significantly reduced by up to 70%, when compared to conventional scheduling.

Tab. 1: Scheduling ($W = 6$, $I = 1$) of LDPC-CC(4,20,20,M)

Scheduling	Optimized	Round-Robin	Flooding
Threshold	5.59 dB	6.09 dB	6.69 dB
1st Window	1, 1, 1, 3, 2, 4	1, 2, 3, 4, 5, 6	1–6
2nd Window	4, 5, 2, 3, 1, 6	1, 2, 3, 4, 5, 6	1–6
3rd Window	3, 6, 4, 2, 1, 5	1, 2, 3, 4, 5, 6	1–6
4th Window	6, 2, 5, 3, 4, 1	1, 2, 3, 4, 5, 6	1–6
5th Window	6, 5, 4, 3, 6, 1	1, 2, 3, 4, 5, 6	1–6
6th Window	6, 4, 5, 3, 1, 4	1, 2, 3, 4, 5, 6	1–6
7th Window	6, 5, 4, 3, 6, 5	1, 2, 3, 4, 5, 6	1–6
8th Window	6, 5, 4, 3, 6, 5	1, 2, 3, 4, 5, 6	1–6
⋮	⋮	⋮	⋮
15th Window	6, 5, 4, 3, 6, 5	1, 2, 3, 4, 5, 6	1–6
16th Window	6, 5, 4, 6, 3, 5	1, 2, 3, 4, 5, 6	1–6
17th Window	6, 5, 4, 3, 1, 6	1, 2, 3, 4, 5, 6	1–6
18th Window	6, 1, 4, 1, 1, 1	1, 2, 3, 4, 5, 6	1–6

References

[1] I. B. Djordjevic, “Advanced coded-modulation for ultra-high-speed optical transmission,” in *OFC* (2014), paper W3J-4.

[2] S. Kudekar, T. Richardson, and R. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” *IEEE TIT* **57** (2011), pp. 803–834.

[3] K. Sugihara et al., “A spatially-coupled type LDPC code with an NCG of 12 dB for optical transmission beyond 100 Gb/s,” in *OFC* (2013), paper OM2B.4.

[4] L. Schmalen et al., “Spatially coupled soft-decision error correction for future lightwave systems,” *JLT* **33** 5 (2015), pp. 1109–1116.

[5] L. Schmalen, V. Aref, J. Cho, and K. Mahdaviani, “Next generation error correcting codes for lightwave systems,” in *ECOC* (2014), paper Th.1.3.3.

[6] L. Schmalen, D. Suikat, D. Rosener, and A. Leven, “Evaluation of left-terminated spatially coupled LDPC codes for optical communications,” in *ECOC* (2014), paper Th.2.3.4.

[7] F. Buchali, L. Schmalen, A. Klekamp, K. Schuh, and A. Leven, “5 × 50 Gb/s WDM Transmission of 32 Gbaud DP-3-PSK over 36,000 km fiber with spatially coupled LDPC coding,” in *OFC* (2014), paper W1A-1.

[8] A. Leven and L. Schmalen, “Status and recent advances on forward error correction technologies for lightwave systems,” *JLT* **32** 16 (2014), pp. 2735–2750.

[9] D. Chang et al., “LDPC convolutional codes using layered decoding algorithm for high speed coherent optical transmission,” in *OFC* (2012), paper OW1H-4.

[10] C.-K. Lian, “A partially parallel low-density parity check code decoder with reduced memory for long code-length,” *Ph.D thesis* (2007).

[11] J. Zhang and M. P. C. Fossorier, “Shuffled iterative decoding,” *IEEE TCOM* **53** 2 (2005), pp. 209–213.

[12] A. I. Casado, M. Griot, and R. D. Wesel, “LDPC decoders with informed dynamic scheduling,” *IEEE TCOM* **58** 12 (2010), pp. 3470–3479.

[13] G. Liva and M. Chiani, “Protograph LDPC codes design based on EXIT analysis,” in *GLOBECOM* (2007), pp. 3250–3254.

[14] Y. Wang, S. C. Draper, and J. S. Yedidia, “Hierarchical and high-girth QC LDPC codes,” *IEEE TIT* **59** 7 (2013), pp. 4553–4583.