Variable-to-check residual belief propagation for LDPC codes

J.-H. Kim, M.-Y. Nam and H.-Y. Song

Variable-to-check residual belief propagation for LDPC decoding for faster convergence, better error performance and lower complexity is proposed. It is similar to residual belief propagation (RBP) that was recently applied to LDPC decoding by Vila Casado *et al.* because it is also a dynamic scheduling belief propagation using residuals, but it is different because the residuals are computed from a variable-to-check message. Simulation shows that it outperforms with only a maximum of eight iterations by about 0.3 dB compared with RBP at an FER of 10^{-4}.

Introduction: In various communication systems including wireless communication, the system needs a faster and more accurate decoder than before, allowing very high data rates. For that reason, a number of decoding algorithms for fast convergence especially for low density parity check (LDPC) codes have been studied and presented in the literature. The most typical scheme is the serial scheduled decoding. The serial scheduled decoding updates towards variable nodes or check nodes in a serial manner. Shuffled belief propagation (SBP) [1] updates towards variable node and layered belief propagation (LBP)[2] updates towards check node. Theoretical tools and simulation results show that serial scheduling converges twice faster than so called flooding, which updates all nodes simultaneously. Since SBP and LBP were introduced, various improvements on them have been tried, some of which are replica SBP [3], row-column message passing scheduling decoding [4], joint row-column decoding [5] and edge-based scheduled BP [6].

Recently, a more effective serial scheduling scheme was introduced using the dynamic scheduling method. This dynamic scheduling method, called residual belief propagation (RBP) [7], accelerates the convergence faster than non-dynamic serial scheduling in terms of the number of iterations [8, 9]. However, RBP has some shortcomings both in performance (because of the RBP's greediness [8, 9]) and in complexity.

In this Letter, we propose a less greedy algorithm, named variable-to-check residual belief propagation (VC-RBP). VC-RBP calculates the residual from the difference of variable-to-check message values before and after update, while the original RBP in [8] and [9] calculates the residual from the difference of check-to-variable message values before and after update. It is interesting to note that only the above difference in the original RBP and the one proposed here results in some non-trivial improvement on error performance with only a small number of iterations. Reducing complexity follows easily by reducing the number of ordering processes of the residuals and by avoiding unnecessary computations.

Residual belief propagation for LDPC codes: Original BP decoding for LDPC codes is achieved by exchange of messages between variable nodes and check nodes. For any check node c_i and variable node v_j that are neighbours, the two message generating functions are defined as [8, 9]:

$$m_{v_j \to c_i} = \sum_{c_a \in N(v_j) \backslash c_i} m_{c_a \to v_j} + C_{v_j} \tag{1}$$

$$m_{c_i \to v_j} = 2 \operatorname{arctanh}\left(\prod_{v_b \in N(c_i) \backslash v_j} \tanh\left(\frac{m_{v_b \to c_i}}{2}\right)\right) \tag{2}$$

where the channel information of v_j is given as $C_{v_j} = \log(p(y_j|v_j = 0)/p(y_j|v_j = 1))$ and y_j is the received signal.

RBP is an informed dynamic scheduling strategy that updates first the message that maximises an ordering metric called the residual. The residual is the difference between the values of a message before and after an update. The intuitive justification of this approach is that the differences between messages before and after an update go to zero as loopy BP converges. Therefore, if a message has a large residual, it means that the message is located in a part of the graph that has not yet converged. Therefore, propagating the message having the largest residual first should speed up the process [7, 8].

The decoding procedure of the RBP algorithm can be explained shortly in three steps (see Fig. 1). Following the notations in [8], let $m_{c_i \to v_j}$ be the message having the largest residual so be chosen for the update. In the first step, RBP updates $m_{c_i \to v_j}$, then discards the residual (otherwise set $r(m_{c_i \to v_j}) = 0$) (see Fig. 1*a*). In the second step, it computes $m_{v_j \to c_a}$ with $c_a \in N(v_j) \backslash c_i$ (see Fig. 1*b*). In the third step, it computes $r(m_{c_a \to v_b})$ with $v_b \in N(c_a) \backslash v_j$ and reorders the messages that are not yet updated in the order of the magnitude of residuals by the difference between the messages updated before and after (see Fig. 1*c*). Notice that in this process, updated messages are discarded, and hence all check-to-variable messages are eventually chosen once to be updated. An exemplary algorithm of the RBP in pseudo-codes is stated in Algorithm 1.

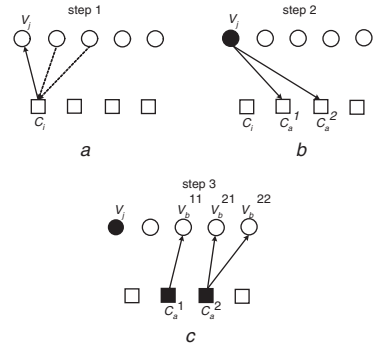

Fig. 1 *Procedure of RBP decoding for LDPC codes*

Even though the original RBP is an effective dynamic scheduling scheme for fast convergence, its application to LDPC codes by Vila Casado *et al.* [8] has some margin of improvement both in error performance and complexity. From a performance aspect, RBP generates new errors which do not appear in non-dynamic scheduled decoding because of its greediness. From a complexity aspect, we specifically try to concentrate on lines 5 and 10. When a check-to-variable message is updated, $m_{c \to v}$ is unnecessarily recomputed because it has already been calculated when $r(m_{c \to v})$ is determined in the preceding step. Moreover, in line 10, Q is reordered whenever the residual value of each edge is calculated. Q denotes a set containing the information on the magnitude of all the remaining residual values.

Algorithm 1 RBP for LDPC codes [8]
1: Initialise all $m_{c \to v} = 0$
2: Initialise all $m_{v_n \to c} = C_n$
3: Compute all $r(m_{c \to v})$ and generate Q
4: Let $m_{c_i \to v_j}$ be the first message in Q
5: Generate and propagate $m_{c_i \to v_j}$
6: Set $r(m_{v_i \to c_j}) = 0$ and reorder Q
7: **for** every $c_a \in N(v_j) \backslash c_i$ **do**
8: Generate and propagate $m_{v_j \to c_a}$
9: **for** every $v_b \in N(c_a) \backslash v_j$ **do**
10: Compute $r(m_{c_a \to v_b})$ and reorder Q
11: end **for**
12: end **for**
13: **if** Stopping rule is not satisfied **then**
14: Go back to line 4;
15: end **if**

Variable-to-check residual belief propagation for LDPC codes: VC-RBP is a less greedy algorithm for overcoming the negative effect of the greediness of RBP. This method not only has better performance but also lower complexity than RBP in [8] and [9]. The main difference between VC-RBP and RBP is in the calculation of residuals: VC-RBP sequentially updates the check node corresponding to the chosen edge, in other words, it calculates the residual from the difference of variable-to-check message values before and after update. On the other hand, RBP sequentially updates the variable node corresponding to the chosen edge, in other words, it calculates the residual from the difference of check-to-variable message values before and after update.

Fig. 2 shows the one-step reduced (compared to RBP) decoding procedure of VC-RBP. In the first step, VC-RBP finds the largest $r(m_{v \to c})$ for choosing the correspondent edge. If the edge corresponding to $m_{v_i \to c_j}$ is chosen, it sets $r(m_{v_i \to c_j}) = 0$, and then updates the check node connected by the chosen edge. In the second step, it updates $m_{c_j \to v_a}$ for

all $v_a \in N(c_j)\backslash v_i$ and then updates $m_{v_a \to c_b}$ for all $c_b \in N(v_a)\backslash c_j$ for calculating the correspondent residual $r(m_{v_a \to c_b})$. This change can reduce the complexity compared with the original RBP [8]. The exact process is explained in algorithm A.

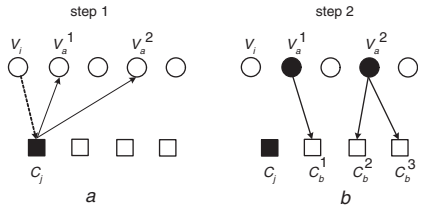

Fig. 2 *Procedure of VC-RBP decoding for LDPC codes*

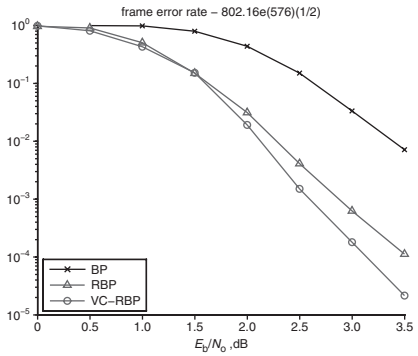

Fig. 3 *FER performance comparison of BP, RBP, VC-RBP decoding with at most eight iterations*

From the performance aspect of error correction, VC-RBP guarantees better performance than RBP. RBP tends to propagate first the message to the less reliable variable nodes because the message which has the largest residual is based on only one check equation. The greediness of RBP can generate new errors that need a large number of message updates to be corrected. However, VC-RBP propagates first the message which has the largest residual based on check equations. So, VC-RBP is a less greedy algorithm compared with RBP. Moreover, VC-RBP solves the trapping set more effectively by updating the check node (based on the largest residual) and then simultaneously all the variable nodes representing the check equation. These facts result in much faster convergence as well as better performance than RBP.

Algorithm A VC-RBP for LDPC codes

1: Initialise all $m_{c \to v} = 0$
2: Initialise all $m_{v_n \to c} = C_n$
3: Find the largest $r(m_{v \to c})$
4: Let $m_{v_i \to c_j}$ be chosen
5: Set $r(m_{v_i \to c_j}) = 0$
6: **for** every $v_a \in N(c_j)\backslash v_i$ **do**
7: Generate and propagate $m_{c_j \to v_a}$
8: **for** every $c_b \in N(v_a)\backslash c_j$ **do**
9: Generate and propagate $m_{v_a \to c_b}$
10: Compute $r(m_{v_a \to c_b})$
11: end **for**
12: end **for**
13: **if** Stopping rule is not satisfied then
14: Go back to line 3;
15: end **if**

Simulation results: This section presents the performance of VC-RBP for various types of LDPC codes. The QC-LDPC code mother matrices designed in the IEEE 802.16e standard [10] is used for code construction for the simulation. We consider an AWGN channel for all the simulations. The code lengths are 576, 1152, 2304, and the code rate 1/2 and 3/4. Because of space limitation, we only show the result of length 576 of rate 1/2.

Fig. 3 shows the performance difference between VC-RBP and RBP with only eight iterations. We note that VC-RBP has a gain of about 0.3 dB over RBP at FER of 10^{-4}.

We have also checked the error performances of these codes with a maximum of 50 iterations at 2.5 dB.

Conclusion: We propose VC-RBP for LDPC codes. Simulation shows that: 1. VC-RBP makes LDPC decoding converge very fast in terms of the number of iterations; 2. it guarantees better performance than RBP in only eight iterations; and 3. it performs similarly better after many sufficient iterations. The complexity reduction can be seen easily from comparison of the two algorithms: algorithm 1 and algorithm A.

Acknowledgment: This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (no. R01-2008-000-20578-0).

28 August 2008
Electronics Letters online no: 20092505
doi: 10.1049/el:20092505

J.-H. Kim, M.-Y. Nam and H.-Y. Song (*School of Electrical & Electronic Engineering, Yonsei University, Seoul 120-749, Korea*)

E-mail: my.nam@yonsei.ac.kr

References

1 Zhang, J., and Fossorier, M.: 'Shuffled belief propagation decoding', *IEEE Trans. Commun.*, 2005, **53**, pp. 209–213
2 Rovini, M., Rossi, F., Ciao, P., Linsalata, N., and Fanucci, L.: 'Layered decoding of non-layered LDPC codes'. Proc. 9th EUROMICRO Conf. on Digital System Design, August 2006, Cavtat, Croatia, pp. 537–544
3 Zhang, J., Wang, Y., Fossorier, M., and Yedidia, J.S.: 'Replica Shuffled Iterative decoding'. IEEE Int. Symp. on Information Theory, Adelaide, Australia, September 2005, pp. 454–458
4 Radosavljevic, P., de Baynast, A., and Cavallaro, J.R.: 'Optimized message passing schedules for LDPC decoding'. 39th Asilomar Conf. on Signals, Systems and Computers, 2005, Pacific Grove, CA, USA, November 2005, pp. 591–595
5 He, Z., Roy, S., and Fortier, P.: 'Lowering error floor of LDPC codes using a joint row-column decoding algorithm'. ICC-2007, Glasgow, Scotland, June 2007
6 Golov, O., and Amrani, O.: 'Edge-based scheduled BP in LDPC codes'. ISIT2007, Nice, France, June 2007
7 Elidan, G., McGraw, I., and Koller, D.: 'Residual belief propagation: informed scheduling for asynchronous message passing'. Proc. 22nd Conf. on Uncertainty in Artificial Intelligence, MIT, Cambridge, MA, USA, July 2006
8 Vila Casado, A.I., Griot, M., and Wesel, R.D.: 'Informed dynamic scheduling for belief-propagation decoding of LDPC codes'. Proc. IEEE ICC 2007, Glasgow, Scotland, June 2007
9 Vila Casado, A.I., Griot, M., and Wesel, R.D.: 'Improving LDPC decoders via informed dynamic scheduling'. IEEE Information Theory Workshop 2007, Lake Tahoe, CA, USA, September 2007
10 IEEE C802.16e-05/0066r3 'LDPC coding for OFDMA PHY'