Simplified sum-product algorithm for decoding LDPC codes with optimal performance

S. Papaharalabos and P.T. Mathiopoulos

A simple, yet effective decoding algorithm is proposed for low-density parity-check (LDPC) codes, which significantly simplifies the check node update computation of the optimal sum-product algorithm. It achieves essentially optimal performance by applying scaling in the decoder's extrinsic information. The proposed algorithm has small performance degradation, e.g. in the order of 0.1 to 0.2 dB, depending on the coded block size.

Introduction: Low-density parity-check (LDPC) codes were proposed initially by Gallager in the early 1960s [1] but they were not deployed for many years mainly because the technology was not mature enough for their practical implementation. Their deployment was not until the late 1990s when they had been firstly rediscovered by MacKay [2]. LDPC codes can be decoded in an iterative fashion by the sum-product algorithm (SPA) based on the message-passing principle. SPA is an optimal decoding algorithm assuming that the code's bipartite graph does not contain any cycles.

The authors in [3] have been among the first to publish relevant work on reduced complexity LDPC decoding algorithms based on SPA. A thorough overview among the most popular reduced complexity decoding algorithms for LDPC codes, such as the min-sum (MS), normalised MS (NMS) and offset MS (OMS), can be found in [4]. Very recently, two low-complexity decoding algorithms for LDPC codes with near optimal performance have been proposed in [5, 6]. In particular, the OMS algorithm was improved in [5] by using a variable, instead of a constant, offset parameter. In [6] a dual min-sum algorithm was proposed, which makes use of, apart from the minimum value, the second least value among the incoming set of message values.

Turbo codes [7] are another family of iterative decoding codes achieving comparable performance with LDPC codes. An improved max-log-MAP decoding algorithm for turbo codes has been recently presented in [8]. It makes use of the first-term MacLaurin series expansion to approximate the correction term of the well-known max* operator, i.e. Jacobian logarithm, used in log-MAP turbo decoding. However, since the application of this approach to LDPC decoding has not been investigated so far, it is the topic of this Letter.

In particular, a novel check node update approximation of SPA is proposed that makes use of such MacLaurin series expansion [8]. Our research has shown that by applying scaling in the decoder's extrinsic information the resulting algorithm not only outperforms the already known NMS and OMS algorithms but achieves essentially identical performance with the optimal SPA. Note that such scaling technique has not been investigated in [8] for turbo decoding, in order to further improve the performance of the presented algorithm.

Check node update approximations in sum-product algorithm: The SPA is composed of three steps: (i) initialisation; (ii) iterative process; and (iii) hard decision. Step (ii) includes the check node and bit node updates, which are the core of the algorithm. The Jacobian approach was proposed in [3, 4] to simplify the check node update computation with negligible bit error rate (BER) performance loss. The check node update can be applied recursively based on the following expression

$$L(U \oplus V) = \log \left( \frac{1 + \exp(L(U) + L(V))}{1 + \exp(L(U) + L(V))} \right)$$

$$= \text{sgn}(L(U)) \text{sgn}(L(V)) \min(|L(U)|, |L(V)|) \log(1 + \exp[-(L(U) + L(V))]) - \log(1 + \exp[-(L(U) - L(V))])$$

where $\oplus$ denotes modulo-2 operation, $\text{sgn}$ denotes absolute value, $U$ and $V$ are two statistically independent binary random variables with log-likelihood ratio (LLR) values of $L(U)$ and $L(V)$, respectively, and $\log$ (·) is the signum function. The two nonlinear logarithmic functions in (1) are implemented with a look-up table (LUT) usually consisting of eight values [3]. Omitting these two nonlinear logarithmic functions, (1) simplifies to

$$L(U \oplus V) \approx \text{sgn}(L(U)) \text{sgn}(L(V)) \min(|L(U)|, |L(V)|) \log(1 + \exp[-(L(U) - L(V))])$$

which is also known as the MS algorithm. To improve performance, scaling is applied in (2), thus obtaining the NMS as

$$L(U \oplus V) \approx \text{sgn}(L(U)) \text{sgn}(L(V)) \min(|L(U)|, |L(V)|) \frac{1}{\alpha}$$

where $\alpha > 1$. Furthermore, the OMS makes use of the following expression

$$L(U \oplus V) \approx \text{sgn}(L(U)) \text{sgn}(L(V)) \max(0, \min(|L(U)|, |L(V)|)) - \beta$$

where $\beta > 0$. In this case, since the incoming messages with magnitude less than $\beta$ are eliminated from the check node update computation, the performance improves.

Proposed simplified check node update computation: Using the MacLaurin series expansion and neglecting orders greater than one [8], one can obtain

$$\log(1 + \exp(-x)) \approx \log 2 - \frac{1}{2} x$$

Furthermore, as the logarithmic term is always greater than zero, the following approximation holds [8]

$$\log(1 + \exp(-x)) \approx \max(0, \log 2 - \frac{1}{2} x)$$

Substituting (6) into (1), a novel check node update approximation is obtained as

$$L(U \oplus V) \approx \text{sgn}(L(U)) \text{sgn}(L(V)) \min(|L(U)|, |L(V)|) \text{sgn}(L(U)) \text{sgn}(L(V)) \text{sgn}(L(U)) \text{sgn}(L(V)) \text{sgn}(L(U)) \text{sgn}(L(V)) \text{sgn}(L(U)) \text{sgn}(L(V)) \text{sgn}(L(U)) \text{sgn}(L(V)) \text{sgn}(L(U)) \text{sgn}(L(V)) \text{sgn}(L(U)) \text{sgn}(L(V))$$

An example of a practical implementation of (6) can be found in [8]. Its advantage is that simple circuits can be used, such as adders and comparators, without the need of a LUT, as (1) requires. Note that further performance improvements can be obtained by applying scaling in (7) in a similar way with the NMS from (3).

Performance evaluation results: Two randomly constructed regular LDPC codes obtained from [9] are considered with block sizes $(N, K) = (504, 252)$ and $(8000, 4000)$. The column weight of these two codes equals to three and the coding rate is $R = 1/2$. The encoded bits are binary phase-shift keying (BPSK) modulated and transmitted with bit energy $E_b$ over the additive white Gaussian noise (AWGN) channel with single-sided power spectral density $N_0$. At the receiver, the LDPC decoder considers the SPA of (1) and its approximations, namely MS from (2), NMS from (3), OMS from (4), as well as the newly proposed algorithm from (7). For a fair comparison with the performance obtained in [4-6], a maximum of 50 decoding iterations are used for the (504, 252) code and 100 decoding iterations are used for the (8000, 4000) code, respectively, and at least 100 bit errors are counted. The scaling factor for NMS is equal to $\alpha = 1.25$, the offset value for OMS is equal to $\beta = 0.15$, and the scaling factor for the proposed algorithm is equal to 0.11.

BER performance evaluation results against $E_b/N_0$ are illustrated in Figs. 1 and 2 for the two LDPC codes, respectively. From Fig. 1 it is noticed that the proposed algorithm is approximately 0.1 to 0.2 dB inferior to both SPA and OMS. Furthermore, the proposed algorithm with scaling and also the NMS outperform SPA by approximately 0.1 dB. This can be explained because SPA is not optimal for the considered block size, owing to the presence of cycles in the message-passing algorithm. A similar observation was reported in [4]. Fig. 2 depicts that the proposed algorithm is approximately 0.1 dB inferior to SPA and less than 0.1 dB inferior to NMS and OMS, respectively.

When scaling is applied, the proposed algorithm outperforms both NMS and OMS, achieving essentially optimal SPA performance.
Conclusion: A novel LDPC decoding algorithm is presented with small performance degradation against the optimal SPA and lower computational complexity. If scaling is additionally applied, then the resulting algorithm outperforms the already known NMS and OMS and achieves essentially optimal SPA performance.

References