Haddadi *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:180
https://doi.org/10.1186/s13638-020-01791-5

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Low-complexity decoding of LDPC codes using reduced-set WBF-based algorithms

Sadjad Haddadi, Mahmoud Farhang* and Mostafa Derakhtian

*Correspondence:
mfarhang@shirazu.ac.ir
Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

Abstract

We propose a method to substantially reduce the computational complexity of iterative decoders of low-density parity-check (LDPC) codes which are based on the weighted bit-flipping (WBF) algorithm. In this method, the WBF-based decoders are modified so that the flipping function is calculated only over a reduced set of variable nodes. An explicit expression for the achieved complexity gain is provided and it is shown that for a code of block length N, the decoding complexity is reduced from $O\left(N^2\right)$ to $O(N)$. Moreover, we derive an upper bound for the difference in the frame error rate of the reduced-set decoders and the original WBF-based decoders, and it is shown that the error performances of the two decoders are essentially the same.

Keywords: Low-density parity-check (LDPC) codes, Iterative decoding, Weighted bit-flipping (WBF)

1 Introduction

The iterative decoding schemes for low-density parity-check (LDPC) codes fall into three main categories: soft-decision methods such as belief propagation (BP) algorithm, hard-decision methods such as bit-flipping (BF) algorithm, and hybrid methods such as weighted bit-flipping (WBF) algorithm [1, 2], with soft-decision and hard-decision methods having the highest and the lowest complexity, respectively. The BP algorithm provides the best performance at the cost of a high implementation complexity [3]. The error performance of BF decoding is inferior to that of the BP, but it is faster and much easier to implement [1, 4, 5]. Moreover, hard-decoding methods like BF are the only option in some applications, such as high-throughput power fiber-optic communications [6, 7], NAND storage systems [8, 9], and McEliece cryptosystem [10],due to hardware limitations.

The WBF decoding algorithm offers a good error performance/decoding complexity trade-off and enjoys an improved performance gained by introducing some measure of reliability (soft information) in the BF decoding algorithm [2]. The WBF algorithm flips some bits in each iteration based on the value of a flipping function and repeats the algorithm until all the parity-check equations are satisfied or the maximum number of

iterations is reached. The performance of the WBF method can be further improved by modifying the flipping function [11–19], and flipping several bits in each iteration can speed up the convergence of decoding [20–23]. However, calculating the flipping function for each variable node requires real-number arithmetic and its computational complexity is much higher than the hard-decision BF decoding.

In this paper, we propose a method to significantly reduce the computational complexity of WBF-based decoders with a negligible loss in the error performance. Our proposed method, named reduced-set (RS) WBF-based decoding, reduces the complexity of obtaining the flipping function to a great extent and can be applied to all WBF-based decoders. Although simulation results do not show any loss in the error performance, we present an upper bound for the difference between the frame error rate (FER) of WBF-based decoders and their RS counterparts.

The rest of this paper is organized as follows. In the next section, some preliminaries about LDPC codes and WBF-based decodings are reviewed. In Section 3, we present the proposed algorithm to reduce the decoding complexity, followed by complexity and error performance analysis. Simulation results are presented in Section 4, and Section 5 concludes the paper.

1.1 Methods/experimental

The research content of this paper is mainly theoretical derivation and analysis, and specific experimental verification will be carried out in a future research.

2 WBF-based algorithms

In this section, we briefly review some preliminaries about LDPC codes and WBF-based decoders.

2.1 Preliminaries

A (d_v, d_c)-regular LDPC code has a sparse parity-check matrix whose column and row weights are exactly d_v and d_c, respectively. An LDPC code is irregular if its rows and/or columns have different weights. An LDPC code can be represented by a bipartite (Tanner) graph which consists of two subsets of nodes, namely, variable nodes (or bit nodes) and check nodes. Variable nodes represent the bits of the codeword and check nodes correspond to the parity-check equations. An edge connects the nth variable node to check node m if and only if bit n is checked by the check-sum m. The set of bits participating in the mth check (i.e., the set of variable nodes connected to the check node m in the Tanner graph) is denoted by $\mathcal{N}(m)$. Similarly, $\mathcal{M}(n)$ denotes the set of checks involving the nth bit. Hence, for an LDPC code with an $M \times N$ parity-check matrix $\mathbf{H} = [h_{mn}]$, we have $\mathcal{N}(m) = \{n : h_{mn} = 1\}$ and $\mathcal{M}(n) = \{m : h_{mn} = 1\}$.

Let $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ be a codeword of a binary LDPC code C of block length N. After BPSK modulation, the transmitted sequence will be $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, with $x_i = 2c_i - 1, i = 1, 2, \ldots, N$. Assuming an additive white Gaussian noise (AWGN) channel, $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ is the real-valued sequence at the output of the receiver matched filter, where $y_i = x_i + n_i$, with n_i's being independent zero-mean Gaussian random variables with variance σ^2. Let $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ be the binary hard-decision sequence obtained from $\mathbf{y}$ (i.e., $z_i = 1$ if $y_i > 0$ and $z_i = 0$ if $y_i \leq 0$). The syndrome vector $\mathbf{s} = (s_1, s_2, \ldots, s_M)$

is then given by $\mathbf{s} = \mathbf{z}\mathbf{H}^T$, i.e., the syndrome component s_m is computed by the check-sum

$$s_m = \sum_{n \in \mathcal{N}(m)} z_n. \tag{1}$$

Vector $\mathbf{s}$ is zero if and only if all parity-check equations are satisfied and $\mathbf{z}$ is a codeword in C.

2.2 WBF-based decoding algorithms

The bit-flipping (BF) algorithm is an iterative hard-decision decoding algorithm that computes all the parity-check equations and then flips a group of bits per iteration that is contained in a preset number of unsatisfied check-sums. The weighted bit-flipping (WBF) algorithm improves the performance of the BF decoding by including some reliability measures of the received symbols in their decoding decisions [2]. Reliability of all the parity-check equations are computed via

$$w_m = \min_{n \in \mathcal{N}(m)} |y_n|, \tag{2}$$

and the flipping function is defined as

$$E_n = \sum_{m \in \mathcal{M}(n)} (2s_m - 1)\, w_m. \tag{3}$$

The WBF decoder first computes the reliability of all the parity-check equations from (2). Next, the decoding algorithm is carried out as follows.

Step 1) For $m = 1, 2, \ldots, M$, compute the syndrome components from (1). Break the algorithm if all the parity-check equations are satisfied ($\mathbf{s} = 0$) or a preset maximum number of iterations is reached. Otherwise, continue.

Step 2) For $n = 1, 2, \ldots, N$, compute the flipping function E_n.

Step 3) Flip the bit z_n for $n = \text{argmax}_{1 \le n \le N} E_n$, and go to Step 1.

In what follows, we review several WBF-based methods that improve the standard algorithm. In [12], the modified WBF (MWBF) is proposed, considering not only the reliability of the syndrome sequence for computing the flipping function, but also the reliability information of the received symbol. The flipping function in the MWBF is modified as

$$E_n = \sum_{m \in \mathcal{M}(n)} (2s_m - 1)\, w_m - a|y_n|, \tag{4}$$

where the weighting factor a can be determined via Monte-Carlo simulation at different SNRs. Reliability-ratio based WBF (RRWBF) proposed in [13] introduces a new quantity called the reliability ratio $R_{m,n}$ and modifies the flipping function as

$$E_n = \sum_{m \in \mathcal{M}(n)} \frac{(2s_m - 1)w_m}{R_{m,n}}. \tag{5}$$

Lee et al. [14] proposed a new version of the RRWBF algorithm which simplifies the calculation. The flipping function in improved RRWBF (IRRWBF) is given by

$$E_n = \frac{1}{|y_n|} \sum_{m \in \mathcal{M}(n)} (2s_m - 1)\, T_m, \tag{6}$$

where $T_m = \sum_{n\in\mathcal{N}(m)} |y_n|$. In [15], Jiang et al. proposed the improved MWBF (IMWBF) algorithm where in computing the flipping function, the reliability of check-sums involving a given bit should exclude that bit, and the reliability computation in (2) should be revised as $w'_{n,m} = \min_{i\in\mathcal{N}(m)/n} |y_i|$, $n \in \mathcal{N}(m)$, and the flipping function as

$$E_n = \frac{1}{a} \sum_{m\in\mathcal{M}(n)} (2s_m - 1)\, \frac{2w'_{n,m}}{\sigma^2} - \left|\frac{2y_n}{\sigma^2}\right|. \tag{7}$$

For a special class of high-rate quasi-cyclic LDPC codes, Liu-Pados WBF (LP-WBF) [16] and its improved version Shan-Zhao-Jiang LP-WBF (SZJLP-WBF) [17] improve the computation of syndrome reliability and perform even better than the IMWBF algorithm at the high SNR regime.

The standard WBF algorithm selects and flips one bit in each iteration. However, to increase the speed of decoding, it can select and flip multiple bits in each iteration. In [20], a threshold adaptation scheme is applied to multi-bit flipping decoding algorithm, where in each iteration, variable nodes with flipping function greater than a pre-defined threshold are selected and flipped. If no flipping occurs, the threshold is reduced and the algorithm continues. A parallel version of IMWBF (PIMWBF) algorithm is proposed in [21] that converges significantly faster and often performs better than IMWBF. The threshold for PIMWBF must be optimized by simulation in each iteration. The proposed multi-bit algorithm in [22] flips multiple bits in each iteration based on a certain threshold that should be optimized by simulation, but the maximum number of bits that are to be flipped in an iteration is restricted. The adaptive-weighted multibit-flipping (AWMBF) algorithm proposed in [23] adjusts the threshold in each iteration as

$$E_{th} = E_{\max} - |E_{\max}| \left(1 - \frac{w_H(\mathbf{s})}{M}\right), \tag{8}$$

where $w_H(\mathbf{s})$ denotes the Hamming weight of the syndrome vector $\mathbf{s}$ and $E_{\max} = \max E_n, n = 1,\ldots,N$. The flipping function used in AWMBF is the same as the flipping function proposed for MWBF (i.e., Eq. (4)). In AWMBF, the threshold in each iteration has a closed-form expression and there is no need for time-consuming simulations to determine the optimum thresholds. In this paper, we will use the AWMBF algorithm in simulations for multi-bit flipping decoders.

Recently, a two-bit WBF (TBWBF) decoder was proposed in [24] for the binary symmetric channel (BSC) that produces reliability bits for both the bit-decision results at variable nodes and the syndrome values at check nodes and exchanges the reliability bits between variable and check nodes as the decoding proceeds.

3 Reduced-set low-complexity decoders

In this section, we propose a method to significantly reduce the computational complexity of all WBF-based algorithms. The complexity of the decoder is also analyzed and an upper bound for its FER is presented.

3.1 Proposed algorithm

All of the WBF-based decoders use a flipping function E_n to select the bits to be flipped. These decoders compute the flipping function for *all* variable nodes in each iteration to detect the erroneous bits in the received sequence. As the flipping function calculation requires real-number arithmetic, the computational complexity of WBF-based

algorithms is essentially due to this part. The main idea behind our proposed algorithm is to reduce the number of flipping function calculations in each iteration by considering only those variable nodes which are likely to be in error. Denote this set of variable nodes in the lth iteration by $\mathcal{A}_l$. In the first iteration, $\mathcal{A}_1$ contains only the variable nodes that are connected to the unsatisfied check nodes. In the next iterations, $\mathcal{A}_l$ contains the variable nodes that participate in the parity-check equations involving the flipped bits in the last iteration. $\mathcal{A}_l$ can thus be written as:

$$\mathcal{A}_l = \{n : n \in \mathcal{N}(m), m \in \mathcal{B}_l\}, \tag{9}$$

where $\mathcal{B}_1 = \{m : s_m \neq 0\}$ and $\mathcal{B}_l = \left\{m : m \in \mathcal{M}(n_{l-1})\right\}$ for $l \geq 2$. n_{l-1} is the index of the flipped bit in the $(l-1)$th iteration. Note that a variable node might appear in several iterations of the decoding process, and variable nodes in the $(l-1)$th iteration are not excluded in the lth iteration.

A reduced-set (RS) WBF-based algorithm is summarized below.

Step 1) For $m = 1, 2, \ldots, M$, compute the syndrome components in (1). Break the algorithm if all the parity-check equations are satisfied ($\mathbf{s} = 0$) or a preset maximum number of iterations is reached. Otherwise, continue.

Step 2) Compute the flipping function E_n for $n \in \mathcal{A}_l$ where $\mathcal{A}_l = \{n : n \in \mathcal{N}(m), m \in \mathcal{B}_l\}$. If $l = 1$, $\mathcal{B}_1 = \{m : s_m \neq 0\}$, otherwise for $l \geq 2$, $\mathcal{B}_l = \{m : m \in \mathcal{M}(n_{l-1})\}$. Update $\mathbb{A}$ as $\mathbb{A} \triangleq \bigcup_{i=1}^{l} \mathcal{A}_i$.

Step 3) Flip bit z_{n_l} for $n_l = \text{argmax}_{n \in \mathbb{A}} E_n$. Increase the iteration number l by one and go to Step 1.

The standard WBF algorithm flips one bit in each iteration. In the following remark, the reduced-set single-bit WBF-based algorithm is extended to reduced-set multi-bit WBF-based algorithm.

Remark 1 *In multi-bit WBF-based algorithms, the decoder selects and flips multiple bits in each iteration. In the first iteration, the set of variable nodes which are likely to be in error, i.e., the set of variable nodes that are connected to the unsatisfied check nodes, is the same for single-bit and multi-bit WBF-based decoders. Let γ_l denote the number of the flipped bits in the lth iteration and $n_{i,l}$, $i = 1, \ldots, \gamma_l$, denote the index of the flipped bits in the lth iteration. In multi-bit WBF-based algorithms, for $l \geq 2$ the set $\mathcal{B}_l$ is modified as* $\mathcal{B}_l = \left\{m : m \in \bigcup_{i=1}^{\gamma_{l-1}} \mathcal{M}\left(n_{i,l-1}\right)\right\}$.

Due to the sparsity of the LDPC parity-check matrix $\mathbf{H}$, the number of bits that participate in each check is small compared to N. So, each erroneous bit causes a small number of unsatisfied check-sums, and for each unsatisfied check-sum, there is a small number of bits that the decoder must decide whether to flip or not. Therefore, even for moderate values of SNR, the set of candidate variable nodes in each iteration constitutes a very small subset of all variable nodes which, in turn, leads to a substantial reduction in the computational complexity of step 2 of the WBF-based decoding algorithms. In the following subsections, we derive explicit expressions for this reduction in complexity and show that the incurred loss in the performance is indeed intangible.

3.2 Computational complexity analysis

In this subsection, we obtain the average number of flipping function calculations as a complexity measure of the RS decoding algorithms and show how the computational complexity of any of the WBF-based decoders is substantially reduced using the proposed algorithm.[1]

We now present a theorem.

Theorem 1 *Consider a (d_v, d_c)-regular LDPC code. For any of the single-bit and the multi-bit RS decoders, the average number of flipping function calculations in the first iteration (i.e., the average cardinality of $\mathcal{A}_1$) is*

$$L_1 = N\left(1 - \left(p_0\beta^{d_v} + (1-p_0)\,(1-\beta)^{d_v}\right)\right), \tag{10}$$

where p_0 is the probability that a bit is received in error and $\beta = \frac{1}{2}\left(1-(1-2p_0)^{d_c-1}\right)$. For the next iterations, i.e., $l \geq 2$, the average number of flipping function calculations for the single-bit RS decoders is given by

$$L_l = d_v\,(d_c - 1) + 1, \qquad l \geq 2 \tag{11}$$

and for the multi-bit RS decoders it is upper bounded as

$$L_l \leq (d_v\,(d_c-1)+1) \times \gamma_{l-1}, \qquad l \geq 2 \tag{12}$$

where γ_l is the number of flipped bits in the lth iteration.

Proof We first obtain the cardinality of $\mathcal{A}_1$, the selected set in the first iteration. As noted in Remark 1, the set of variable nodes that are connected to the unsatisfied check nodes in the first iteration is the same for both single-bit and multi-bit RS decoders. So, the cardinality of set $\mathcal{A}_1$ (i.e., L_1) is the same for both single-bit and multi-bit RS decoders.

We define the indicator function $\mathcal{I}_i$ of the ith variable node as

$$\mathcal{I}_i = \begin{cases} 1, & i \in \mathcal{A}_1 \\ 0, & i \notin \mathcal{A}_1 \end{cases} \tag{13}$$

for $1 \leq i \leq N$. The cardinality of $\mathcal{A}_1$, denoted by l_1, is a random variable and can be written as $l_1 = \sum_{i=1}^{N} \mathcal{I}_i$. So, the average number of variable nodes in set $\mathcal{A}_1$ is obtained as

$$L_1 = \sum_{i=1}^{N} E\left\{\mathcal{I}_i\right\}, \tag{14}$$

and we have

$$E\left\{\mathcal{I}_i\right\} = 1 - \Pr\left\{i \notin \mathcal{A}_1\right\}. \tag{15}$$

The event $i \notin \mathcal{A}_1$ occurs when all checks involving the ith bit are satisfied. Let μ_m be the event that the mth check involving the ith bit is satisfied. The ith bit participates in d_v checks, hence

$$\begin{aligned} \Pr\left\{i \notin \mathcal{A}_1\right\} &= \Pr\left\{\mu_1, \mu_2, \ldots, \mu_{d_v}\right\} \\ &= \Pr\left\{\mu_1, \mu_2, \ldots, \mu_{d_v} | i \in \mathcal{E}\right\} \Pr\left\{i \in \mathcal{E}\right\} \\ &\quad + \Pr\left\{\mu_1, \mu_2, \ldots, \mu_{d_v} | i \notin \mathcal{E}\right\} \Pr\left\{i \notin \mathcal{E}\right\}, \end{aligned} \tag{16}$$

[1] The number of iterations in the original and the proposed algorithms are the same, but the number of calculations required to obtain the flipping function in a WBF-based decoder has been sharply decreased in the proposed method.

where $\mathcal{E}$ denotes the set of all erroneous bits in the received sequence. We assume that the code is 4-cycle free, i.e., no two code bits are checked by the same two parity constraints. This structural property is imposed on almost all LDPC code constructions and is very important to achieve good error performance with iterative decoding [5, 25, 26]. If there are no cycles of length 4 in the Tanner graph, no two checks share more than one variable node. In other words, if more than one variable node appear at two different checksums, there will be at least one cycle of length 4 in the Tanner graph. On the other hand, in the first iteration, values of variable nodes are received directly from the channel output. Thus, in the *first iteration*, all variable nodes are independent (as the noise was assumed to be white). Therefore, assuming a 4-cycle free graph, all checks involving the ith bit do not share any other bits, and conditioned on the ith bit all these checks will be independent in the first iteration. Thus,

$$\Pr\{i \notin \mathcal{A}_1\} = p_0 \prod_{m=1}^{d_v} \Pr\{\mu_m | i \in \mathcal{E}\} + (1-p_0) \prod_{m=1}^{d_v} \Pr\{\mu_m | i \notin \mathcal{E}\}. \tag{17}$$

$\Pr\{\mu_m | i \in \mathcal{E}\}$ is the probability that the number of erroneous bits participating in the mth check (except the ith bit) is an odd number and is given by [4]

$$\beta = \frac{1}{2}\left(1 - (1-2p_0)^{d_c-1}\right). \tag{18}$$

Similarly, $\Pr\{\mu_m | i \notin \mathcal{E}\} = 1-\beta$. Therefore,

$$\Pr\{i \notin \mathcal{A}_1\} = p_0\beta^{d_v} + (1-p_0)(1-\beta)^{d_v}. \tag{19}$$

Using equations (14), (15) and (19), we have

$$\begin{aligned} L_1 &= \sum_{i=1}^{N} (1 - \Pr\{i \notin \mathcal{A}_1\}) \\ &= N\left(1 - \left(p_0\beta^{d_v} + (1-p_0)(1-\beta)^{d_v}\right)\right). \end{aligned} \tag{20}$$

For $l \geq 2$, $\mathcal{A}_l$ contains all the variable nodes that participate in the parity-check equations involving the flipped bit in the last iteration. The number of variable nodes that participate in the parity-check equations involving a given variable node is $d_v(d_c-1)$ (see Fig. 1). Single-bit RS decoders flip only one bit in each iteration. Therefore, in this case, the cardinality of set $\mathcal{A}_l$ for $l \geq 2$, will be

$$L_l = d_v(d_c-1) + 1. \tag{21}$$

In multi-bit RS decoders, γ_l bits are flipped in the lth iteration, and for each flipped bit in the last iteration, the RS decoder must update $d_v(d_c-1)+1$ flipping functions. In general, parity-check equations involving flipped bits in the last iteration may have some bits in common. So, the cardinality of the set $\mathcal{A}_l$, $l \geq 2$, in multi-bit RS decoders is upper bounded as

$$L_l \leq (d_v(d_c-1)+1) \times \gamma_{l-1}. \tag{22}$$

□

End of Proof.

Plotted in Fig. 2 is L_1 versus SNR for (3,6) and (4,32)-regular codes. It is seen that the result of (20) matches the average number of variable nodes in $\mathcal{A}_1$ obtained from Monte-Carlo simulation.

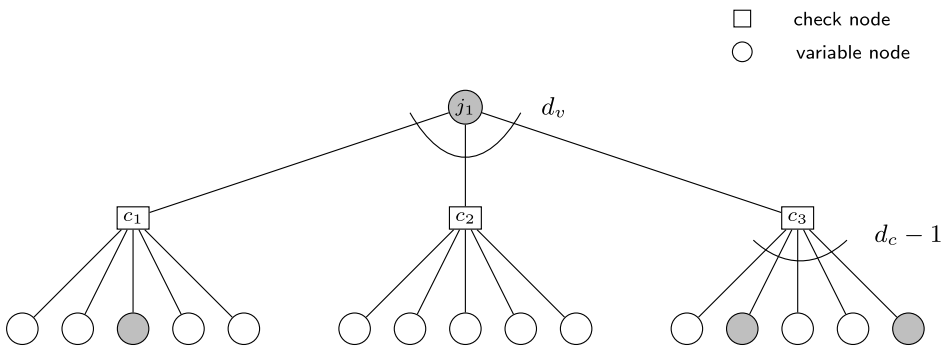

Fig. 1 A subgraph spreading from variable node j_1 associated with a (3,6)-regular LDPC code. The erroneous variable nodes are painted gray

If k is the number of iterations required in the decoding process, by using (22), the number of flipping function calculations in multi-bit RS decoders can be upper bounded as[2]

$$\begin{aligned} L =& L_1 + \sum_{l=2}^{k} L_l \\ \leq& L_1 + (d_v(d_c - 1) + 1) \times \sum_{l=2}^{k} \gamma_{l-1} \end{aligned} \tag{23}$$

Assume that the decoder is in the waterfall region and is able to detect and correct some erroneous bits in each iteration and eventually corrects all of them. Therefore, $\sum_{l=2}^{k} \gamma_{l-1}$ is equal to the number of erroneous bits in the received sequence. For large block sizes, the number of erroneous bits is approximately Np_0 and it can be easily verified that for $p_0 \ll 1$ and large N, we have

$$L \leq 2Np_0 \left(d_v \left(d_c - 1\right) + 1\right). \tag{24}$$

For single-bit RS decoder, the inequality in Eq. (24) becomes equality (cf. (21) and (22)). From Eq. (24), it can be seen that the computational complexity is linear in the codeword length. This fact was checked by simulation and the results are presented in Table 1. The simulation results for several (3,6)-regular LDPC codes of different codeword lengths are tabulated along with the theoretical results. The parity check matrices of the codes are given in [27], and the SNR is considered to be 6 dB. We observe that both the single-bit and multi-bit RS decoders need essentially the same average number of flipping function calculations, and the derived upper bound for L in (24) is quite tight. As expected, as N increases, the upper bound obtained from Eq. (24) get closer to the simulation results.

On the other hand, original WBF-based decoders compute the flipping function for all N variable nodes in each iteration. So, the number of flipping function calculations for WBF-based decoders is approximately kN. So, the ratio of the average number of flipping function calculations for WBF-based and RS decoders—which can be considered as the complexity gain—is lower bounded as

$$G_c \geq \frac{k}{2p_0\big(d_v(d_c - 1) + 1\big)}. \tag{25}$$

[2]Note that when the codeword is received correctly, i.e., $\mathbf{s} = 0$, there is no need to calculate the flipping function, so $L = 0$.

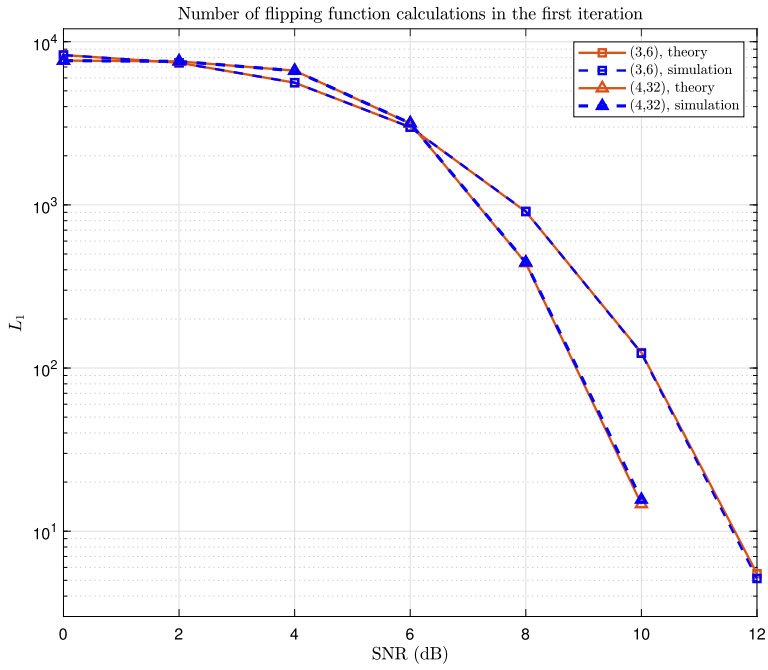

Fig. 2 L_1 versus SNR for the (3,6) and (4,32)-regular LDPC codes

By assuming that the decoder is able to detect and correct one erroneous bit in each iteration, in single-bit decoders, the average number of iterations required to obtain the correct codeword is the same as the number of erroneous bits in the received sequence, i.e., $k = Np_0$, and the inequality in equation (25) becomes equality (cf. (21) and (22)). It should also be noted that the complexity gain is higher for a sparser parity-check matrix.

For example, for a $(3, 6)$-regular code with $N = 10^5$ and at SNR=6 dB, G_c for the single-bit and multi-bit RS decoders is obtained as 3125 and 1279, respectively. Although the complexity gain is smaller for multi-bit RS decoders, it is still significant.

3.3 Performance analysis

To evaluate the performance of the proposed RS algorithm and compare it with the original WBF-based decoder,[3] we first note that if $\mathbb{A} \triangleq \bigcup_i \mathcal{A}_i$, the selected set by RS decoders, which contains *all* erroneous bits, both decoders will have the same performance. However, in general, some erroneous bits may happen not to be in the selected set and thus the RS decoders can never detect and correct them. Specifically, an erroneous bit will not be included in $\mathcal{A}_1$ if all parity-checks in which this bit participates are satisfied (i.e., if these checks involve an even number of errors). This bit may never enter $\mathbb{A}$ in the next iterations, and so the RS decoder will totally miss it. Therefore, the performance of RS-based decoders will generally be inferior to that of the original decoders. However, in the following theorem, we show that the difference between the FER of original WBF-based decoders P_O and RS decoders P_{RS} is indeed negligible.

[3] By "original" WBF-based decoders, we mean all WBF-based decoders previously proposed in the literature (to differentiate them from their RS counterparts).

Table 1 Number of flipping function calculations for several (3,6)-regular LDPC codes over AWGN channel at SNR = 6 dB

N	L (SB)	L (MB)	L (upper bound)	$\frac{L}{N}$ (SB)	$\frac{L}{N}$ (MB)	$\frac{L}{N}$ (upper bound)
1000	651	649	736	0.651	0.649	0.736
2640	1743	1738	1943	0.66122	0.65833	0.735984
4000	2645	2640	2820	0.66125	0.66	0.705
4896	3245	3243	3468	0.66278	0.66237	0.708333
8000	5307	5310	5536	0.66337	0.66375	0.692
10000	6652	6649	6845	0.6652	0.6649	0.6845

SB: single-bit, *MB*: multi-bit

Theorem 2 *The difference between the FER of the original WBF-based and RS decoders for a (d_v, d_c)-regular LDPC code is upper bounded as*

$$\Delta P \leq N \sum_{\varepsilon_0=1}^{N} \sum_{\theta \in T} P_1(\theta) \binom{d_v(d_c-1)}{\theta} \binom{N-d_v(d_c-1)-1}{\varepsilon_0-\theta-1} p_0^{\varepsilon_0} (1-p_0)^{N-\varepsilon_0}, \tag{26}$$

where $T = \left\{ \theta' \mid d_v \leq \theta' \leq \min\{d_v(d_c-1), \varepsilon_0 - 1\},\ \theta' \overset{2}{\equiv} d_v \right\}$,[4] *and*

$$P_1(\theta) = \frac{\sum_{(X_1,X_2,\ldots,X_{d_v}) \in \Psi'_\theta} \binom{d_c-1}{X_1}\binom{d_c-1}{X_2}\cdots\binom{d_c-1}{X_{d_v}}}{\sum_{(X_1,X_2,\ldots,X_{d_v}) \in \Psi_\theta} \binom{d_c-1}{X_1}\binom{d_c-1}{X_2}\cdots\binom{d_c-1}{X_{d_v}}},$$

with X_i's being non-negative integers. The sets Ψ_θ and Ψ'_θ are defined as

$$\Psi_\theta = \left\{ (X_1, X_2, \ldots, X_{d_v}) \mid 0 \leq X_i \leq d_c - 1, \sum_i X_i = \theta \right\},$$

and

$$\Psi'_\theta = \left\{ (X_1, X_2, \ldots, X_{d_v}) \mid 0 \leq X_i \leq d_c - 1, \sum_i X_i = \theta, X_i \ odd \right\}.$$

Proof Let $\mathbf{b}$ and $\hat{\mathbf{b}}_{RS}$ be the transmitted message and the estimated message by the RS decoder, respectively. The FER of the RS decoder can then be written as

$$\begin{aligned} P_{RS} &\triangleq \Pr\left\{ \mathbf{b} \neq \hat{\mathbf{b}}_{RS} \right\} \\ &= \Pr\left\{ \mathbf{b} \neq \hat{\mathbf{b}}_{RS}, \mathcal{E} \subseteq \mathbb{A} \right\} + \Pr\left\{ \mathbf{b} \neq \hat{\mathbf{b}}_{RS}, \mathcal{E} \nsubseteq \mathbb{A} \right\}, \end{aligned}$$

where $\mathcal{E} = \{j_i, i = 1, 2, \ldots, \varepsilon\}$ is the set of indices of erroneous bits in the received sequence and $\mathbb{A} \triangleq \bigcup_i \mathcal{A}_i$ is the selected set of variable nodes in the decoding process. By defining $\hat{\mathbf{b}}_O$ as the estimated sequence by the original WBF-based decoder, we have

$$\begin{aligned} \Pr\left\{ \mathbf{b} \neq \hat{\mathbf{b}}_{RS}, \mathcal{E} \subseteq \mathbb{A} \right\} &= \Pr\left\{ \mathbf{b} \neq \hat{\mathbf{b}}_O, \mathcal{E} \subseteq \mathbb{A} \right\} \\ &\leq \Pr\left\{ \mathbf{b} \neq \hat{\mathbf{b}}_O \right\} \\ &\triangleq P_O. \end{aligned} \tag{27}$$

[4] $\theta' \overset{2}{\equiv} d_v$ means θ' and d_v are congruent modulo 2, i.e., both are even or both are odd.

Therefore, using the Bayes rule,

$$P_{RS} \leq P_O + \Pr\left\{\mathbf{b} \neq \hat{\mathbf{b}}_{RS} | \mathcal{E} \nsubseteq \mathbb{A}\right\} \Pr\left\{\mathcal{E} \nsubseteq \mathbb{A}\right\}. \tag{28}$$

By defining $\Delta P = P_{RS} - P_O$, we have

$$\begin{aligned}\Delta P &\leq \Pr\left\{\mathbf{b} \neq \hat{\mathbf{b}}_{RS} | \mathcal{E} \nsubseteq \mathbb{A}\right\} \Pr\left\{\mathcal{E} \nsubseteq \mathbb{A}\right\} \\ &\leq \Pr\left\{\mathcal{E} \nsubseteq \mathcal{A}_1\right\}.\end{aligned} \tag{29}$$

The event $\mathcal{E} \nsubseteq \mathcal{A}_1$ is the event that some erroneous variable nodes may not be in the selected set $\mathcal{A}_1$. The number of erroneous bits ε in the received sequence (i.e., the cardinality of set $\mathcal{E}$) is a random variable with binomial distribution $B(\varepsilon, p_0)$, i.e.,

$$\Pr\{\varepsilon = \varepsilon_0\} = \binom{N}{\varepsilon_0} p_0^{\varepsilon_0} (1 - p_0)^{N - \varepsilon_0}. \tag{30}$$

Therefore, we have

$$\begin{aligned}\Pr\left\{\mathcal{E} \nsubseteq \mathcal{A}_1\right\} &= \sum_{\varepsilon_0=0}^{N} \Pr\left\{\mathcal{E} \nsubseteq \mathcal{A}_1, \varepsilon = \varepsilon_0\right\} \\ &= \sum_{\varepsilon_0=1}^{N} \Pr\left\{\bigcup_{i=1}^{\varepsilon_0} \left(j_i \notin \mathcal{A}_1\right), \varepsilon = \varepsilon_0\right\} \\ &\leq \sum_{\varepsilon_0=1}^{N} \varepsilon_0 \Pr\left\{j_1 \notin \mathcal{A}_1, \varepsilon = \varepsilon_0\right\} \\ &= \sum_{\varepsilon_0=1}^{N} \varepsilon_0 \Pr\left\{j_1 \notin \mathcal{A}_1 | \varepsilon = \varepsilon_0\right\} \Pr\{\varepsilon = \varepsilon_0\}.\end{aligned} \tag{31}$$

By defining Θ as the number of erroneous bits participating in checks that involve bit j_1, we have

$$\Pr\left\{j_1 \notin \mathcal{A}_1 | \varepsilon = \varepsilon_0\right\} = \sum_{\theta=0}^{K} \Pr\left\{j_1 \notin \mathcal{A}_1 | \Theta = \theta, \varepsilon = \varepsilon_0\right\} \Pr\{\Theta = \theta | \varepsilon = \varepsilon_0\}, \tag{32}$$

where $K = \min\{d_v(d_c - 1), \varepsilon_0 - 1\}$. For a (d_v, d_c)-regular code

$$\Pr\{\Theta = \theta | \varepsilon = \varepsilon_0\} = \frac{\binom{d_v(d_c-1)}{\theta}\binom{N - d_v(d_c-1) - 1}{\varepsilon_0 - \theta - 1}}{\binom{N-1}{\varepsilon_0 - 1}}. \tag{33}$$

To compute $\Pr\left\{j_1 \notin \mathcal{A}_1 | \Theta = \theta, \varepsilon = \varepsilon_0\right\}$, we define X_i as the number of erroneous bits participating in the ith check that involves bit j_1. Figure 1 shows an example in which $d_v = 3, d_c = 6, \theta = 3$ and the erroneous variable nodes are painted gray. It is seen that $X_1 = 1, X_2 = 0$ and $X_3 = 2$. Noting that a check is satisfied if an even number of erroneous bits are involved in it, and by defining

$$\Psi_\theta \triangleq \left\{\left(X_1, X_2, \ldots, X_{d_v}\right) \mid 0 \leq X_i \leq d_c - 1, \sum_i X_i = \theta\right\},$$

and

$$\Psi'_\theta \triangleq \left\{\left(X_1, X_2, \ldots, X_{d_v}\right) \mid 0 \leq X_i \leq d_c - 1, \sum_i X_i = \theta, X_i \text{ odd}\right\},$$

we have

$$P_1(\theta) \triangleq \Pr\left\{j_1 \notin \mathcal{A}_1 | \Theta = \theta, \varepsilon = \varepsilon_0\right\} = \frac{\sum_{(X_1,X_2,\ldots,X_{d_v})\in\Psi'_\theta} \binom{d_c-1}{X_1}\binom{d_c-1}{X_2}\ldots\binom{d_c-1}{X_{d_v}}}{\sum_{(X_1,X_2,\ldots,X_{d_v})\in\Psi_\theta} \binom{d_c-1}{X_1}\binom{d_c-1}{X_2}\ldots\binom{d_c-1}{X_{d_v}}}. \tag{34}$$

From the definition of set Ψ'_θ, if d_v is an even (odd) number, then $P_1(\theta) = 0$ when θ is odd (even). Moreover, $P_1(\theta) = 0$ for $\theta < d_v$. Therefore, using (29)-(34), the upper bound of ΔP is obtained as (26), and from (28) the FER of the RS decoders can be upper bounded as

$$P_{RS} \leq P_O + N \sum_{\varepsilon_0=1}^{N} \sum_{\theta\in T} P_1(\theta) \binom{d_v\,(d_c-1)}{\theta} \binom{N - d_v(d_c-1) - 1}{\varepsilon_0 - \theta - 1} p_0^{\varepsilon_0} (1-p_0)^{N-\varepsilon_0}. \tag{35}$$

□

End of Proof.

The upper bound presented in Theorem 2 is general and is applicable to both single-bit and multi-bit WBF-based decoders. Indeed, as shown above, the difference between the FER of the original WBF-based decoders and their RS counterparts (ΔP) is upper bounded by the probability that some erroneous variable nodes may not be in the selected set $\mathcal{A}_1$ in the first iteration (see Eq. (29)), and the set $\mathcal{A}_1$ is the same in single-bit and multi-bit WBF-based decoders.

Remark 2 *Noting that $P_1(\theta) < 1$, from (26) we have*

$$\Delta P < N \sum_{\varepsilon_0=1}^{N} \sum_{\theta\in T} \binom{d_v(d_c-1)}{\theta} \binom{N - d_v(d_c-1) - 1}{\varepsilon_0 - \theta - 1} p_0^{\varepsilon_0} (1-p_0)^{N-\varepsilon_0}. \tag{36}$$

By changing the order of the summations and modifying their bounds, we have

$$\Delta P < N \sum_{\theta=d_v,\theta\overset{2}{\equiv}d_v}^{d_v(d_c-1)} \left(\binom{d_v(d_c-1)}{\theta} \times \sum_{\varepsilon_0=\theta+1}^{N-d_v(d_c-1)+\theta} \binom{N - d_v(d_c-1) - 1}{\varepsilon_0 - \theta - 1} p_0^{\varepsilon_0} (1-p_0)^{N-\varepsilon_0} \right). \tag{37}$$

Making the substitution $\varepsilon'_0 = \varepsilon_0 - \theta - 1$ and using $\sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} = 1$, after some simplification, (37) becomes

$$\Delta P < N \sum_{\theta=d_v,\theta\overset{2}{\equiv}d_v}^{d_v(d_c-1)} \binom{d_v(d_c-1)}{\theta} p_0^{\theta+1} (1-p_0)^{d_v(d_c-1)-\theta}. \tag{38}$$

From the above inequality, it is clear that in the high SNR regime ΔP tends to zero at least as $p_0^{d_v+1}$, and the upper bound will be tighter for a code with a larger degree of the variable nodes.

4 Results and discussion

In this section, we compare WBF-based and reduced-set (RS) decoders in terms of computational complexity and the probability of error. In the simulations, we use $(3, 6)$ and $(4, 32)$-regular LDPC codes with rates $\frac{1}{2}$ and $\frac{7}{8}$, respectively. The parity-check matrix for the $(3, 6)$-regular code is constructed with the progressive edge growth (PEG) method [28]. For the $(4, 32)$-regular code, we use the LDPC code considered in [29] for near earth

applications which is a quasi-cyclic code. The maximum number of iterations is set to 100 in all simulations.

First, an analysis of the computational complexity of the decoders based on the average number of flipping function calculations (L) is presented. Plotted in Fig. 3 is L in the RS decoder versus SNR for the $(3, 6)$ and $(4, 32)$-regular LDPC codes with codeword length 10000 and 8176, respectively. Average number of flipping function calculations obtained by Monte-Carlo simulation for single-bit and multi-bit WBF-based decoders, along with the upper bound of (24) are shown in this figure. As expected, in the high SNR regime, the upper bound becomes quite tight for both single-bit and multi-bit decoders.

In Fig. 4, the average number of flipping function calculations is plotted versus SNR for the RS and original WBF-based decoders. Both single-bit and multi-bit decoders are considered in this figure. As discussed in Section 3.2, the average number of flipping function calculations in single-bit RS and multi-bit RS decoders are almost the same, and this is confirmed by the results obtained by simulation in Fig. 4. It is clearly seen that using the RS algorithm results in about three orders of magnitude decrease in the decoding complexity in single-bit WBF-based decoders and at least two orders of magnitude decrease in the decoding complexity in multi-bit WBF-based decoders. Moreover, this reduction in the complexity is higher for the sparser codes (cf. (25)). It should also be noted that the number of flipping function calculations required in original (non-RS) multi-bit decoders at the medium SNR regime is less than those required in the single-bit decoders, while in the low and high SNR regimes the number of flipping function calculations required in the two decoding algorithms are the same. This behavior can be explained as follows. At low SNRs, neither decoding algorithms are able to correct the errors, so the decoding process continues until the predefined maximum number of iterations is reached,

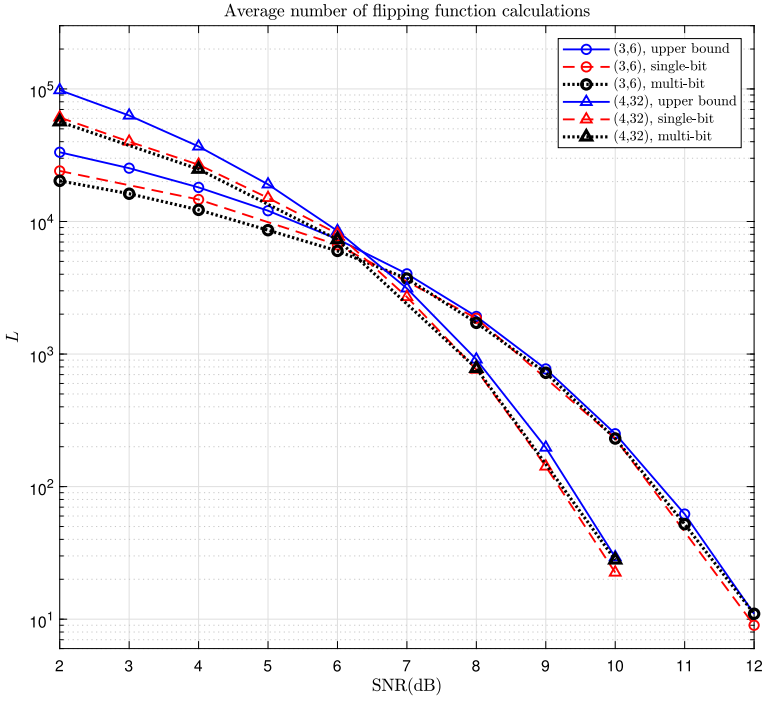

Fig. 3 L versus SNR for the (3,6) and (4,32)-regular LDPC codes with $N = 10000$ and 8176, respectively

and thus the average number of flipping function calculations is the same for single-bit and multi-bit WBF decoders. At intermediate SNRs, the convergence speed of the multi-bit decoding algorithm is higher (i.e., the average number of required iterations is smaller), and therefore, the average number of flipping function calculations for the multi-bit decoder is lower. At the high SNR regime, either the received sequence is error-free or the number of erroneous bits is very small. In this case, the number of required iterations in the decoding process in the single-bit and multi-bit decoders are almost equal. These results are shown in Fig. 5. In this figure, the average number of required iterations versus SNR is plotted to evaluate the convergence of the original and proposed RS single-bit and multi-bit decoders. As expected, the average number of iterations of the original and RS decoders are nearly identical, i.e., both decoders have similar convergence speeds.

To evaluate the probable performance loss incurred by using the RS decoders (compared to their original WBF-based counterparts), the FER and BER for both the RS and original WBF-based decoders are plotted in Figs. 6, 7, 8, and 9. In these figures, regular $(3, 6)$ and $(4, 32)$ LDPC codes with codeword length 10000 and 8176 are employed. In Fig. 6, the simulation results for the FER of the $(3, 6)$ and $(4, 32)$-regular codes for both the RS and original WBF-based decoders, along with an upper bound for the FER of the RS decoder are plotted. In this figure, P_O is obtained by Mont-Carlo simulations for both single-bit standard WBF decoder [2] and multi-bit AWMBF decoder [23], and the upper bound is given by equation (35). We observe that both the RS and the original WBF-based decoders have essentially the same performance, and the derived upper bound for

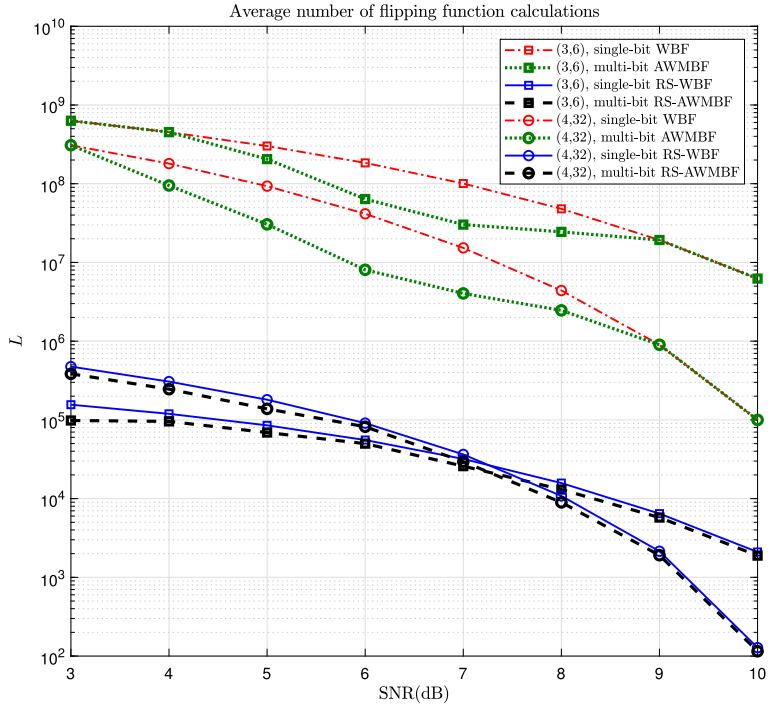

Fig. 4 L versus SNR for the (3,6) and (4,32)-regular codes with $N = 10^5$ and 81760, respectively

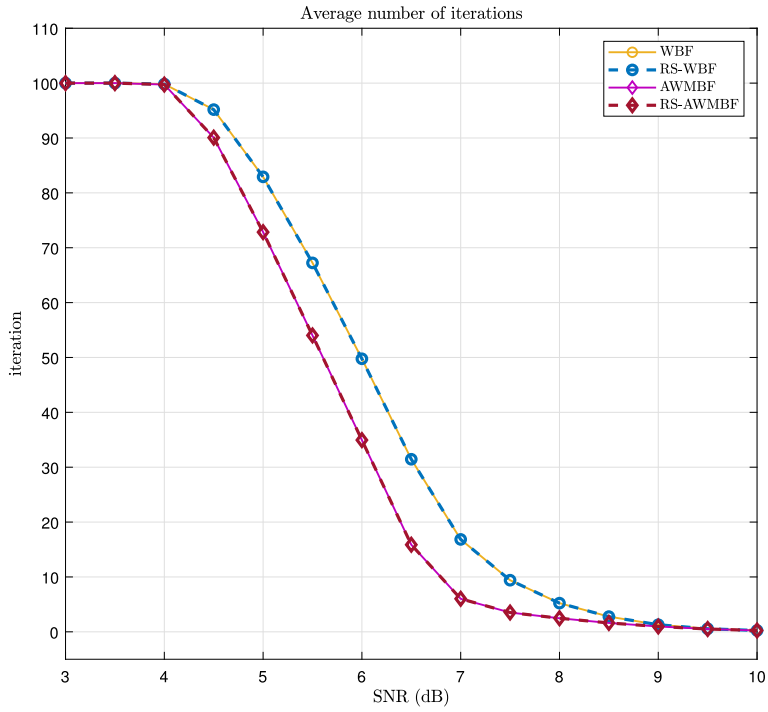

Fig. 5 Number of iterations versus SNR for the (4,32)-regular LDPC code with $N = 8176$

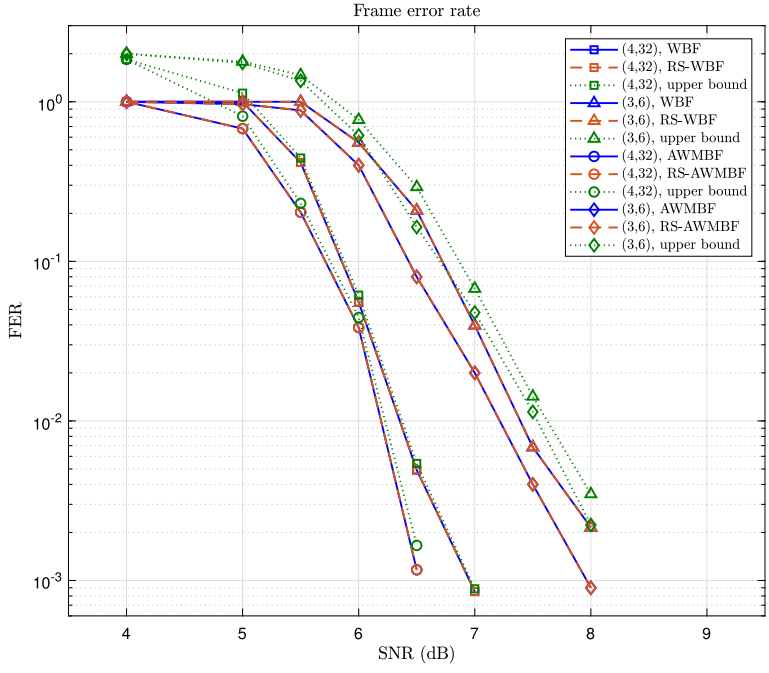

Fig. 6 The FER and the upper bound of ΔP versus SNR for the (3,6) and (4,32)-regular codes

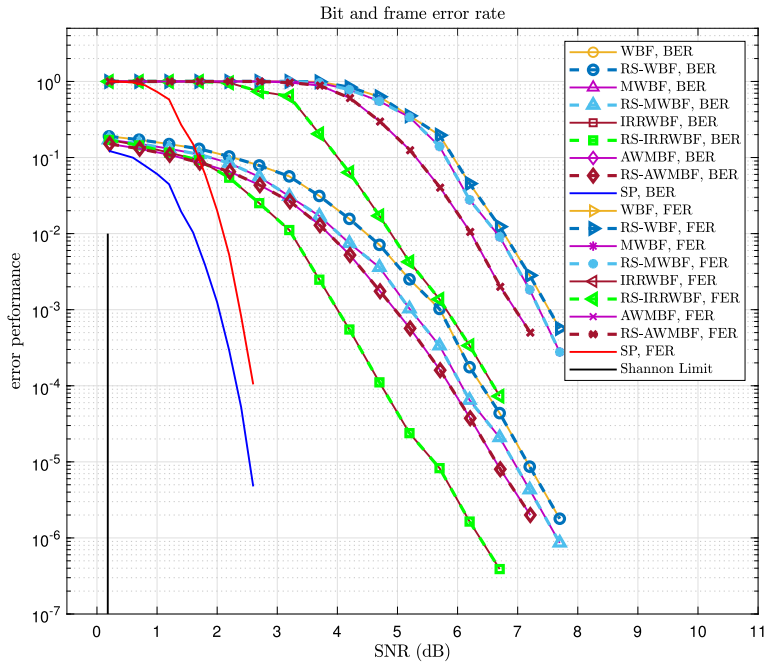

Fig. 7 Performance of (3,6)-regular LDPC code with rate $\frac{1}{2}$ over AWGN channel

the RS decoders are quite tight in both single-bit and multi-bit decoders. As can be seen in Fig. 6, the upper bound of ΔP for $(4, 32)$-regular LDPC code is tighter than the upper bound for $(3, 6)$-regular LDPC code, because as discussed in Section 3.3, the upper bound is tighter for a code with a larger degree of the variable nodes (recall that ΔP tends to zero at least as $p_0^{d_v+1}$).

In Figs. 7, 8, and 9, the error performance of the proposed RS and the original WBF-based decoders are shown. Figures 7 and 8 show the results over the AWGN channel and Fig. 9 over the BSC. In these simulations, we have employed the single-bit WBF, MWBF, IRRWBF, and TBWBF decoders and multi-bit AWMBF decoder and their RS counterparts. As expected, the error performance in terms of BER and FER of the original decoders and RS decoders are very close.

5 Conclusion

We proposed a method to reduce the computational complexity of iterative LDPC decoders based on the WBF algorithm. It was shown that the decoder computational complexity is significantly reduced, especially when the code length is large. Our method performs just as well as the existing WBF-based iterative decoding algorithms and the FER and BER of the two decoders are essentially the same. In the proposed method, instead of all variable nodes, the decoder considers only a subset of variable nodes that are potentially erroneous and thus the complexity of the flipping function calculation is significantly reduced.

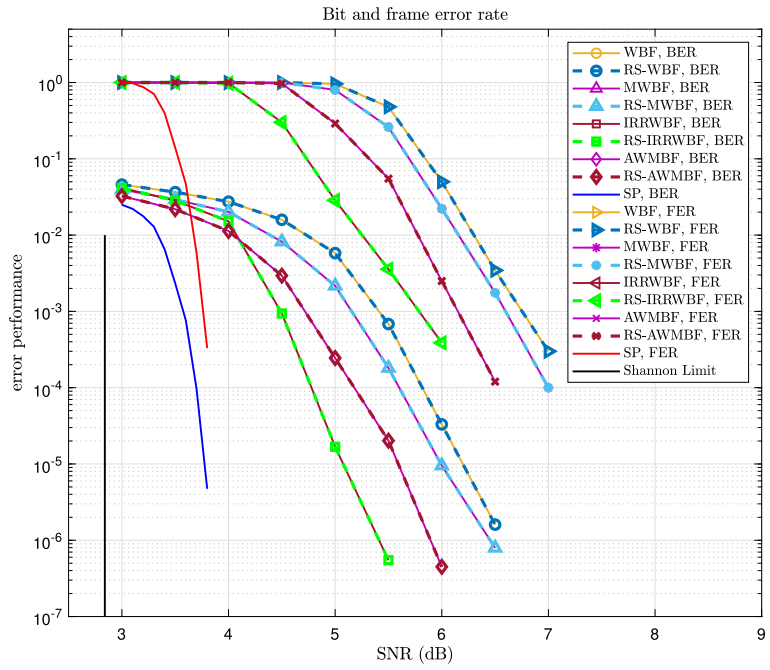

Fig. 8 Performance of (4,32)-regular LDPC code with rate $\frac{7}{8}$ over AWGN channel

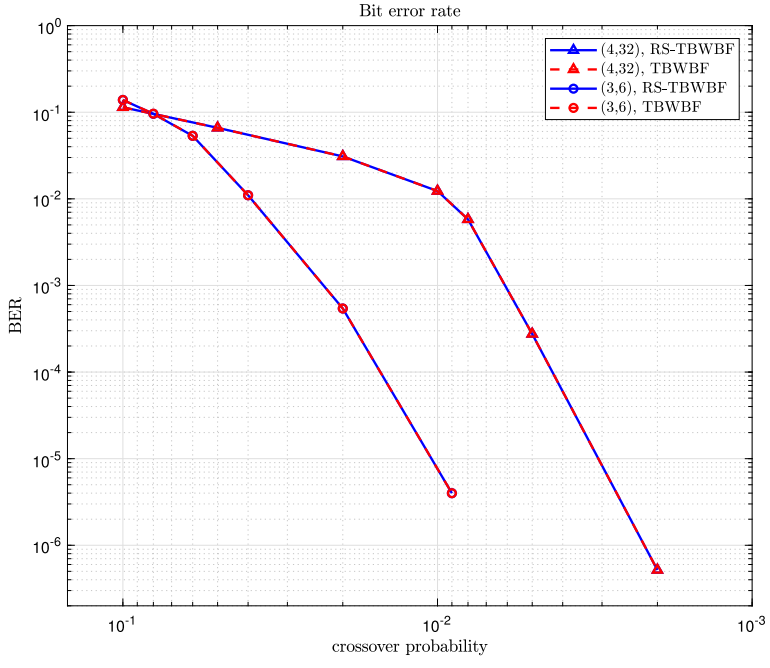

Fig. 9 Performance of regular LDPC codes with rate $\frac{1}{2}$ and $\frac{7}{8}$ over BSC channel

Abbreviations
LDPC: Low-density parity-check; RS: Reduced set; AWMBF: Adaptive-weighted multibit-flipping; BF: Bit flipping; WBF: Weighted bit flipping; MWBF: Modified weighted bit flipping; IMWBF: Improved modified weighted bit flipping; RRWBF: Reliability-ratio weighted bit flipping; IRRWBF: Improved reliability-ratio weighted bit flipping; TBWBF: Two-bit weighted bit flipping; BER: Bit error rate; FER: Frame error rate; AWGN: Additive white Gaussian noise; BSC: Binary symmetric channel; PEG: Progressive edge growth

Authors' contributions
SH contributed to the main idea and performed the numerical simulations. MF and MD contributed to the mathematical analysis. SH and MF wrote the paper. All authors read and approved the final manuscript.

Funding
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 22 December 2019 Accepted: 31 August 2020
Published online: 21 September 2020

References
1. R. Gallager, Low-density parity-check codes. IRE Trans. Inf. Theory. **8**(1), 21–28 (1962)
2. Y. Kou, S. Lin, M. P. Fossorier, Low-density parity-check codes based on finite geometries: a rediscovery and new results. IEEE Trans. Inf. Theory. **47**(7), 2711–2736 (2001)
3. T. Richardson, M. Shokrollahi, R. Urbanke, Design of capacity-approaching irregular low-density parity-check codes. IEEE Trans. Inf. Theory. **47**(2), 619–637 (2001)
4. S. Lin, D. J. Costello, *Error control coding*. (Pearson Education India, Upper Saddle River, 2004)
5. W. Ryan, S. Lin, *Channel codes: classical and modern*. (Cambridge University Press, Cambridge, 2009)
6. A. Sheikh, A. G. i Amat, G. Liva, Achievable information rates for coded modulation with hard decision decoding for coherent fiber-optic systems. J. Light. Technol. **35**(23), 5069–5078 (2017)
7. I. Djordjevic, W. Ryan, B. Vasic, *Coding for optical channels*. (Springer, New York, 2010)
8. F. Ghaffari, B. Vasic, in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Probabilistic gradient descent bit-flipping decoders for flash memory channels (IEEE, Florence, 2018), pp. 1–5
9. K. Le, F. Ghaffari, in *2018 15th International Multi-Conference on Systems, Signals & Devices (SSD)*, On the use of hard-decision LDPC decoders on MLC NAND flash memory (IEEE, Hammamet, 2018), pp. 1453–1458
10. M. Baldi, *QC-LDPC code-based cryptography*. (Springer, Heidelberg, 2014)
11. A. Nouh, A. H. Banihashemi, Bootstrap decoding of low-density parity-check codes. IEEE Commun. Lett. **6**(9), 391–393 (2002)
12. J. Zhang, M. P. Fossorier, A modified weighted bit-flipping decoding of low-density parity-check codes. IEEE Commun. Lett. **8**(3), 165–167 (2004)
13. F. Guo, L. Hanzo, Reliability ratio based weighted bit-flipping decoding for low-density parity-check codes. Electron. Lett. **40**(21), 1356–1358 (2004)
14. C.-H. Lee, W. Wolf, Implementation-efficient reliability ratio based weighted bit-flipping decoding for LDPC codes. Electron. Lett. **41**(13), 755–757 (2005)
15. M. Jiang, C. Zhao, Z. Shi, Y. Chen, An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes. IEEE Commun. Lett. **9**(9), 814–816 (2005)
16. Z. Liu, D. A. Pados, in *Communications, 2003. ICC'03. IEEE International Conference On*, Low complexity decoding of finite geometry LDPC codes, vol. 4 (IEEE, Anchorage, 2003), pp. 2713–2717
17. M. Shan, C. Zhao, M. Jiang, Improved weighted bit-flipping algorithm for decoding LDPC codes. IEE Proc. Commun. **152**(6), 919–922 (2005)
18. T. C.-Y. Chang, Y. T. Su, Dynamic weighted bit-flipping decoding algorithms for LDPC codes. IEEE Trans. Commun. **63**(11), 3950–3963 (2015)
19. N. Miladinovic, M. P. Fossorier, Improved bit-flipping decoding of low-density parity-check codes. IEEE Trans. Inf. Theory. **51**(4), 1594–1606 (2005)
20. J. Cho, W. Sung, Adaptive threshold technique for bit-flipping decoding of low-density parity-check codes. IEEE Commun. Lett. **14**(9), 857–859 (2010)
21. X. Wu, C. Zhao, X. You, Parallel weighted bit-flipping decoding. IEEE Commun. Lett. **11**(8), 671–673 (2007)
22. J. Jung, I.-C. Park, Multi-bit flipping decoding of LDPC codes for NAND storage systems. IEEE Commun. Lett. **21**(5), 979–982 (2017)
23. T.-C. Chen, Adaptive-weighted multibit-flipping decoding of low-density parity-check codes based on ordered statistics. IET Communications. **7**(14), 1517–1521 (2013)
24. J. Oh, J. Ha, A two-bit weighted bit-flipping decoding algorithm for LDPC codes. IEEE Commun. Lett. **22**(5), 874–877 (2018)
25. M. Esmaeili, M. Tadayon, T. Gulliver, Low-complexity girth-8 high-rate moderate length QC-LDPC codes. AEU-Int. J. Electron. Commun. **64**(4), 360–365 (2010)
26. M. Gholami, M. Alinia, Z. Rahimi, An explicit method for construction of CTBC codes with girth 6. AEU-Int. J. Electron. Commun. **74**, 183–191 (2017)
27. D. MacKay, Encyclopedia of sparse graph codes (2020). http://www.inference.phy.cam.ac.uk/mackay/codes/data.html. Accessed May 2020
28. X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, in *Proc. IEEE GLOBECOM Conf.*, Progressive edge-growth tanner graphs (IEEE, San Antonio, TX, 2001), pp. 995–1001
29. CCSDS 131.1-O-2, Low Density Parity Check Codes for Use in Near-Earth and Deep Space Applications. The Consultative Committee for Space Data Systems, Orange book, Issue 2, September 2007 (2007)

Publisher's Note