# Single-Minimum LDPC Decoding Offset Optimization Methods

Daniel B. Dermont, Jérémy Nadal, François Leduc-Primeau
Department of Electrical Engineering, Polytechnique Montréal, QC, Canada.

***Abstract*—Low-density parity-check codes are widely used in communication systems. To meet the high throughput and energy efficiency requirements of current and future systems, it is desirable to further simplify the decoder. Quantized min-sum (MS) decoders are of particular interest for their low implementation complexity, which can be further reduced by computing a single minimum (SM) during check node update, instead of two. However, this simplification can lead to poor decoding performance unless it is carefully incorporated. In this paper, we formalize a general optimization problem for SM decoding, and propose search heuristics to solve it. In addition, we provide density evolution (DE) equations for the first two decoding iterations that properly take into account the lack of extrinsic update rule, and show that this DE result can be used to obtain good solutions to the SM optimization problem with low computational complexity.**

## I. Introduction

As communication speeds increase and data transmission channels become more complex, encoding transmissions with error correction codes becomes increasingly important. Low-density parity-check (LDPC) codes [1] are used in many current communication networks, including Ethernet, WiFi and 5G, and are being considered for use in impending 6G networks [2]. There already exist limitations in hardware in reaching higher 5G transmission speeds [3]. The peak data rate proposed for 6G networks will be 1 Tbps, 50 times that of existing 5G networks [4]. Thus, there is a need for more efficient algorithms and hardware architectures.

A possible way to improve the efficiency of a min-sum (MS) LDPC decoder, first proposed in [5], is to introduce an approximation in the check node update that consists of computing only a single global minimum value, instead of two minimum values as required by the extrinsic message passing rule of the sum-product algorithm. The results presented in [5] suggest a single-minimum (SM) check node is approximately half the size of a standard min-sum check node, and implementations of single-min approaches are becoming more visible in literature [6]. However, these improvements come at the cost of performance, as the emulation of the second minimum potentially introduces errors.

An MS decoder will find both the global minimum of incoming messages and a second minimum excluding the global minimum message, while a single-min decoder emulates this second minimum using a correction factor and the first minimum. This approach was implemented in [7]–[9], with a configurable SM offset which was shown to outperform the algorithm introduced in [5] and approximate the performance of a min-sum decoder. The work presented in [10] further improves on the performance of SM approaches by incorporating a scaled SM offset dependant on iteration number. Reference [11] applies these ideas to offset min-sum decoding, with a similar approach to weighting the SM offsets as compared to [10]. The work in [11] notably includes offsets varying not only by iteration number, but also by check node degree.

While the methods presented in previous work consider SM emulation using offset values, their offset optimizations are done over a more limited solution space, and they do not systematically optimize the message quantization parameters. In this paper, we introduce new methods of SM offset optimization for single-min decoders. We begin by formulating a general problem for optimizing SM offsets, to which we apply stochastic search methods. A new top-level heuristic for applying these search methods, referred to as "Windowed Search," is outlined. The algorithm attempts to account for dependencies by optimizing SM offsets within a window of several iterations, then advancing the window through successive iterations. In parallel, we propose an analytical method for optimizing the single-min offset for the first two decoding iterations using density evolution. To the best of our knowledge, density evolution has not been applied to single-min decoding schemes. The main advantage of this method is the improved computational efficiency when compared to Monte Carlo (MC)-based methods.

The rest of this paper is organized as follows: Section II reviews min-sum and SM decoding. Section III presents novel SM offset optimization methods. Section IV presents an overview of density evolution (DE) analysis for MS and SM decoding, and proposes optimization methods for SM decoders using DE. Section V presents the performance analyses of fixed and variable-offset SM decoders compared to conventional MS decoders, including the DE-based optimized SM decoder. Section VI concludes the paper.

## II. System description

We consider binary phase-shift keying (BPSK) modulated data transmitted over an additive white Gaussian noise channel. After encoding, the $i^{\text{th}}$ bit is modulated and symbol $x_i$ is transmitted. The received signal is $y_i = x_i + w_i$, where $w_i$ is the noise term of variance $\sigma^2$. The decoder receives a quantized channel log-likelihood ratio (LLR) $L_i$ of the $i^{\text{th}}$ coded bit, expressed as $L_i = \text{sat}_Q \lfloor 2\alpha y_i/\sigma^2 + 1/2 \rfloor$ , where $\alpha$ is a constant scaling factor (referred to as LLR scaling factor),

$\lfloor . \rfloor$ is the floor operator and $\text{sat}_Q$ is a saturation operator that ensures that $L_i \in [-Q, Q]$, $Q \in \mathbb{N}^*$ being the maximum LLR magnitude.

### A. Message passing algorithm

Decoding is based on the message-passing algorithm [12], where messages are iteratively passed between variable node (VN) and check node (CN) neighbors. The message value sent from the $i^{\text{th}}$ variable node to the $j^{\text{th}}$ check node at iteration $\ell$ is here formalized as $\lambda_{i\to j}^{(\ell)}$, and $\gamma_{j\to i}^{(\ell)}$ is the message sent from the CN at index $j$ to the VN at index $i$ at decoding iteration $\ell$. In addition, $\mathcal{V}_j$ is the set of VN indices connected to CN $j$ and $C_i$ is the set of CN indices connected to VN $i$. The variable-to-check (V2C) $\lambda$-update rule for a given $i \to j$ edge is formulated as

$$\lambda_{i\to j}^{(\ell)} = L_i + \sum_{k\in C_i\setminus\{k\}} \gamma_{j\to k}^{(\ell)} . \tag{1}$$

The initial check-to-variable (C2V) message is given by $\gamma_{j\to i}^{(0)} = 0$. Defining $f(.)$ as the C2V $\gamma$-update rule function, we have $\gamma_{j\to i}^{(\ell)} = f(\boldsymbol{\lambda}_{\bar{e},i\to j}^{(\ell-1)})$, with $\boldsymbol{\lambda}_{\bar{e},i\to j}^{(\ell-1)} = [\lambda_{i\to k}^{(\ell-1)}]_{\forall k\in\mathcal{V}_j\setminus\{j\}}$ corresponding to the vector composed of all $\lambda$-messages connected to CN index $j$, excluding the extrinsic message (edge $i \to j$). Note that the subscript $\bar{e}$ here refers to extrinsic message exclusion. Finally, $i^{\text{th}}$ bit-decision is performed by evaluating $\mathfrak{s}(\lambda_{i\to j}^{(\ell)} + \gamma_{j\to i}^{(\ell)})$, $\mathfrak{s}(.)$ being the signum function.

### B. Min-sum approximation

The $\gamma$-update rule $f(.)$ can be simplified for resource-efficient implementations. The $\gamma$-message magnitude is approximated by taking the minimum magnitude from the incoming $\lambda$-messages. Decoders that use this approximation are referred to as min-sum (MS) decoders [13]. A correction factor can be applied to the message magnitude, such as an offset $\beta \in \mathbb{N}$ or normalization term $\eta \in \mathbb{R}^*+$, for a non-negligible performance improvement. Applied to a given message vector $\boldsymbol{\lambda}$, the $\gamma$-update function takes the form

$$f(\boldsymbol{\lambda}) = \eta \times \left\lfloor \max\left(\min\ |\boldsymbol{\lambda}| - \beta, 0\right) \times \prod_{\lambda\in\boldsymbol{\lambda}} \mathfrak{s}(\lambda) \right\rfloor \tag{2}$$

Most hardware architectures update messages CN-by-CN, using a dedicated CN processor. In this processor, the $\gamma$-update operation is efficiently implemented using a component that computes the 2 minimum values among $\lambda$-message inputs using several cascaded comparator blocks. This remains the most computationally complex block in an MS decoder.

### C. Single-Min approximation

The $\gamma$-update function can be further simplified by only computing one minimum magnitude per check node, as proposed in [5]. This is equivalent to violating the extrinsic exclusion rule during the message update: $\gamma_{e,j\to i}^{(\ell)} = \mathfrak{s}(\lambda_{i\to j}^{(\ell-1)}) f(\boldsymbol{\lambda}_{e,i\to j}^{(\ell-1)})$, with $\boldsymbol{\lambda}_{e,i\to j}^{(\ell-1)} = [\lambda_{i\to k}^{(\ell-1)}]_{\forall k\in\mathcal{V}_j}$. The subscript $e$ here refers to the inclusion of extrinsic information in the message update function. As a result, the number of cascaded comparators is divided by 2 in the check node processing unit, reducing its complexity and propagation delay by half. However, these improvements come at the cost of degraded error-correction performance. To reduce the performance gap, [7], [8], [10] propose an emulation of the second minimum magnitude by adding a correction factor in the form of an offset that may vary during the decoding procedure. The most general approach is to affect an offset $\omega_{i,j}^{(\ell)}$, referred to as an SM offset, for an edge $i \to j$ and iteration $\ell$. The emulation of the second minimum occurs when the minimum V2C message magnitude is extrinsic: $|\lambda_{i\to j}^{(\ell-1)}| < |\gamma_{\bar{e},j\to i}^{(\ell)}|$, $\gamma_{\bar{e},j\to i}^{(\ell)}$ being the message obtained using (2) without applying correction factors. Thus, we have

$$|\gamma_{e,j\to i}^{(\ell)}| = \begin{cases} \min\left(\eta\left\lfloor|\lambda_{i\to j}^{(\ell-1)}| + \omega_{i,j}^{(\ell)}\right\rfloor, Q\right) & |\lambda_{i\to j}^{(\ell-1)}| < |\gamma_{\bar{e},j\to i}^{(\ell)}| \\ \max\left(\eta\left\lfloor|\gamma_{\bar{e},j\to i}^{(\ell)}| - \beta\right\rfloor, 0\right) & \text{otherwise} \end{cases}, \tag{3}$$

and $\gamma_{e,j\to i}^{(\ell)} = \mathfrak{s}(\gamma_{\bar{e},j\to i}^{(\ell)}) \times |\gamma_{e,j\to i}^{(\ell)}|$. The resulting message magnitude needs to be saturated to the value $Q$ if it exceed this value.

## III. SM Offset Optimization

Little attention has been paid in the literature as to how SM offsets may be systematically optimized. We propose a more general formulation of the offset optimization problem, as well as solutions using novel techniques.

### A. Problem formulation

Let there be a matrix of positive integer SM offsets $\boldsymbol{\Omega}$ of size $n_{\text{edge}} \times \ell_{\max}$, where $n_{\text{edge}} = \sum_j |\mathcal{V}_j|$ is the number of edges in the Tanner graph and $\ell_{\max}$ is the maximum number of iterations supported by the decoder. Each matrix entry corresponds to an SM offset $\omega_{i,j}^{(\ell)}$ applied at a specific decoder iteration $\ell \in [1, \ell_{\max}]$ and edge $j \to i, \forall j, i \in \mathcal{V}_j$. We wish to find the matrix $\boldsymbol{\Omega}$ such that bit-error rate (BER) is minimized. Additionally, both the normalization factor $\eta$ and LLR scaling factor $\alpha$, real-valued parameters, need to be optimized. Let the $(\eta, \alpha, \boldsymbol{\Omega})$ tuple represent a particular choice of decoder parameters. Correction factors $\eta, \alpha$ and offset matrix entries $\omega_{i,j}^{(\ell)}$ have for maximum values $\eta_{\max}, \alpha_{\max}$ and $\omega_{\max}$. The optimal parameters $(\eta^\star, \alpha^\star, \boldsymbol{\Omega}^\star)$ that minimize the BER performance $B(\eta, \alpha, \boldsymbol{\Omega}, \zeta)$ of the SM decoder at $E_b/N_o = \zeta$ are given by solving

$$(\eta^\star, \alpha^\star, \boldsymbol{\Omega}^\star) = \underset{(\eta,\alpha,\boldsymbol{\Omega})}{\arg\min}\, B(\eta, \alpha, \boldsymbol{\Omega}, \zeta). \tag{4}$$

The BER $B(\eta, \alpha, \boldsymbol{\Omega}, \zeta)$ can be estimated through MC simulation. However, problem (4) is difficult to solve with reasonable computational complexity since it is non-convex and the solution space is extremely large. In addition, MC simulation requires extensive simulation times. Hence, the search space must be reduced. One solution is to constrain the offset matrix $\boldsymbol{\Omega}$ using one of three approaches: i) similarly to [7], [8], all offsets can be set to a single scalar value ($\omega_{i,j}^{(\ell)} = \omega$), ii) the offset values only vary with the iteration index ($\omega_{i,j}^{(\ell)} = \omega^{(\ell)}$) or iii) with the edge index ($\omega_{i,j}^{(\ell)} = \omega_{i,j}$).

Approach (iii) can be simplified by grouping edges into $N_G$ groups, then assigning the same offset value $\omega_u$ to all edges belonging to the $u^{\text{th}}$ group for a given iteration $\ell$. It is useful in practice to design these groups based on the CN degree $\text{dc}_j = |\mathcal{V}_j|$ of each message, as proposed in [11]. If $G_u$ is the set of CN degrees belonging to group $u$ ($G_u \cap G_{u'} = \emptyset$ if $u \neq u'$), then $\omega_{i,j}^{(\ell)} = \omega_u, \forall i$ if $\text{dc}_j \in G_u$. Note that the approach described in this paragraph can be combined with (ii), i.e. offset values vary by both edge-group and iteration index. Then, $\boldsymbol{\Omega}$ can be equivalently represented as an $N_G \times \ell_{\max}$ matrix denoted $\boldsymbol{\Omega_G} = [\omega_u^{(\ell)}]$. For the rest of this section, we consider $\boldsymbol{\Omega_G}$ as the offset matrix to be optimized.

It is worth noting that it is not necessary to perform a full search over the maximum iteration count intended for the SM decoder. Instead, the findings gathered from a search over fewer iterations can be extrapolated to design for the intended iteration limit $\ell_{\max}$. For approach ii) and iii), extrapolation can be performed using linear regression, with saturation at zero if extrapolated offsets take negative values. To design $\boldsymbol{\Omega_G}$, two-dimensional extrapolation can be performed using a modified Akima method [14], derived from cubic interpolation.

### B. Top-level optimization using coordinate descent

To further simplify the search space for optimizing $(\eta, \alpha, \boldsymbol{\Omega_G})$, we propose optimizing these parameters through coordinate descent (CD) [15]. CD allows optimization to target each parameter separately; the remaining parameters are fixed while a search is performed to find the best BER-performing value for the targeted parameter, as shown in the equations below

$$\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star} = \arg\min_{\boldsymbol{\Omega_G}} B(\eta^{\star}, \alpha^{\star}, \boldsymbol{\Omega_G}), \tag{5}$$

$$\eta^{\star} = \arg\min_{\eta} B(\eta, \alpha^{\star}, \boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}), \tag{6}$$

$$\alpha^{\star} = \arg\min_{\alpha} B(\eta^{\star}, \alpha, \boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}). \tag{7}$$

The above equations are applied in a loop over $I$ iterations to improve the solution. For (6) and (7), a single-variable minimizer tool may be used, as these values are scalars with definite bounds. The minimization algorithm is based on a golden section search and parabolic interpolation, with tolerance value $10^{-4}$ [16], [17]. Problem (5) is more complex; methods to solve it will be described in further detail in Section III-C and Section IV.

An initial solution must be identified for all parameters before performing CD. We observed there is little difference in CD-optimized $\alpha$ and $\eta$ between MS decoding and fixed offset SM decoding. Therefore, we propose to first find initial values for $\alpha$ and $\eta$ using CD on (6) and (7), with BER obtained through MS decoding (variable $\boldsymbol{\Omega_G}$ is excluded). We then move to the SM decoder to set the initial value of $\boldsymbol{\Omega_G}$. For simplicity, we consider that all coefficients of $\boldsymbol{\Omega_G}$ are fixed to $\omega$, such that $\omega_u^{(\ell)} = \omega$. This constraint applies only for this initial step. Then, $\omega$ is obtained by performing CD on (5)–(7). This allows $\alpha$ and $\eta$ values to be further optimized.

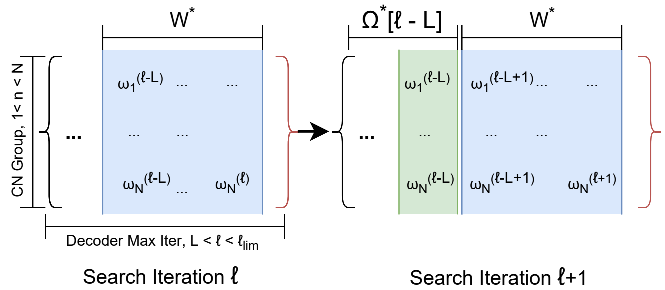


Fig. 1. An example of an iteration of the generic windowed search, evaluated for $p = 1$.

### C. SM offset optimization using windowed search

SM decoding operations involve dependencies that allow the offset at one iteration to change the behavior of later iterations. This can be explained by the fact that messages become correlated over successive iterations due to the extrinsic exclusion principle being violated, as demonstrated later in Section IV-C. Therefore, problem (5) is not convex, and unlike MS decoding, offsets cannot be optimized on a per-iteration basis. We propose introducing a top-level heuristic that incorporates memory from previous decoder iterations to optimize subsequent ones, which we refer to as window search algorithm (WSA).

We denote as $\boldsymbol{\Omega_G}[\ell]$ the offset matrix of the $\ell \leq \ell_{\max}$ first iterations (size $N_G \times \ell$). Defining a window length $L$, we propose finding $\boldsymbol{\Omega_G} = \boldsymbol{\Omega_G}[\ell_{\max}]$ by successively optimizing $\boldsymbol{\Omega_G}[\ell]$ from $\ell = L$ to $\ell = \ell_{\max}$ with step $p \in [1, L]$. During each step, the $L$ last columns of $\boldsymbol{\Omega_G}[\ell]$, corresponding to a submatrix $\boldsymbol{W}$ of size $N_G \times L$, are optimized by stochastic search. Meanwhile, the remaining coefficients are kept constant during the optimization process. The resulting optimized offset matrix is denoted $\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}[\ell]$. By noting that $\boldsymbol{\Omega_G}[\ell] = [\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}[\ell - L], \boldsymbol{W}]$, $\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}[\ell]$ can be obtained by solving the following optimization problem:

$$\boldsymbol{W}^{\star} = \arg\min_{\boldsymbol{W}} B(\eta^{\star}, \alpha^{\star}, [\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}[\ell - L], \boldsymbol{W}]), \tag{8}$$

and setting $\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}[\ell] = [\boldsymbol{\Omega}_{\boldsymbol{G}}^{\star}[\ell - L], \boldsymbol{W}^{\star}]$. The search is performed using a genetic algorithm, or, when $L \times N_G$ is small, through exhaustive search. Then, $\ell$ is incremented by $p$, and the next offset matrix $\boldsymbol{\Omega_G}[\ell]$ is optimized through the same procedure. This process repeats until $\ell = \ell_{\max}$. Figure 1 depicts an iteration of the search for $p = 1$.

## IV. Optimization using density evolution

The main drawback of MC-based searches outlined in the previous section is their long simulation time. In the following section, we investigate a faster, analytical approach for solving (5) using DE.

### A. DE for MS Decoding

DE, first proposed for sum-product decoding in [18], is an analysis tool that uses probabilistic properties of belief propagation to predict the behavior of a decoder at each iteration. If we assume that a code is free of dependencies introduced by cycles in the Tanner graph, we can accurately predict the performance of a decoder at each iteration by evolving the probability density of the channel output. The result given by this analysis assumes that the LDPC code length tends to infinity.

The channel LLR cumulative distribution function (CDF), denoted $\Phi_\lambda^{(0)}(k), k \in [-Q, Q]$, is given by

$$\Phi_\lambda^{(0)}(k) = \frac{1}{2} + \frac{1}{\sqrt{4\sigma^2}} \mathrm{erf}\left(\left(k + \frac{1}{2}\right)\frac{\sigma^2}{\alpha} - 1\right), \tag{9}$$

where $\mathrm{erf}(x) = 2/\sqrt{\pi}\int_0^x \exp[-t^2]dt$ is the Gauss error function. The resulting probability-mass function (PMF) is denoted $P_\lambda^{(0)}(k)$ or $\boldsymbol{P}_\lambda^{(0)} = [P_\lambda^{(0)}(-Q), ..., P_\lambda^{(0)}(Q)]$ in vector format. The density of V2C messages at iteration $\ell$ is given by

$$P_\lambda^{(\ell)}(k) = \left(\left[\circledast_{d_v-1} \boldsymbol{P}_\gamma^{(\ell)}\right] \circledast \boldsymbol{P}_\lambda^{(0)}\right)(k), \tag{10}$$

where $\boldsymbol{P}_\gamma^{(\ell)} = [P_\gamma^{(\ell)}(-Q), ..., P_\gamma^{(\ell)}(Q)]$ is the C2V message PMF vector and $\circledast_n$ operator is a $n$-fold convolution on a vector $\boldsymbol{X}$ with itself. Finally, the PMF of the C2V message is defined as [19]

$$P_\gamma^{(\ell)}(k) = \begin{cases} \Phi_+(k) - \Phi_+(k+1), & k > 0 \\ 1 - \left(1 - P_\lambda^{(\ell)}(0)\right)^{d_c-1}, & k = 0 \\ \Phi_-(k) - \Phi_-(k-1)), & k < 0 \end{cases} \tag{11}$$

where $\Phi_+(k)$ and $\Phi_-(k)$ are given by

$$\Phi_+(k) = \sum_{p \text{ even}} \binom{d_c - 1}{p} A_-^p(k) A_+^{d_c-1-p}(k), \tag{12}$$

$$\Phi_-(k) = \sum_{p \text{ odd}} \binom{d_c - 1}{p} A_-^p(-k) A_+^{d_c-1-p}(-k), \tag{13}$$

with $A_-$ and $A_+$ are respectively the complementary CDF (CCDF) of the negative and positive values of the saturated V2C messages at iteration $\ell$ [19]. To simplify presentation, we present the equations for single-edge-type codes, but it is also possible to develop equations for multi-edge-type codes.

### *B. Proposed DE optimization method*

Instead of directly solving (5), we propose to find the offset $\omega_{i,j}^{(\ell)}$ that minimizes the mean square error (MSE) $\epsilon(\omega_{i,j}^{(\ell)})$ between the C2V messages obtained respectively through the SM and MS message update rules, for a given edge $i \to j$ and iteration $\ell$. This translates to the following optimization problem:

$$\arg\min_{\omega_{i,j}^{(\ell)}} \epsilon(\omega_{i,j}^{(\ell)}) = \arg\min_{\omega_{i,j}^{(\ell)}} \mathrm{E}\left|\gamma_{e,j\to i}^{(\ell)} - \gamma_{\bar{e},j\to i}^{(\ell)}\right|^2. \tag{14}$$

This method does not guarantee finding the optimal solution of problem (5), but its main advantage is that offsets can be independently optimized for each edge, and, more importantly, analytical methods can be applied to solve this problem. Recalling that the SM C2V message update rule can be formulated as in (3), the MSE term can be expressed as

$$\epsilon(\omega_{i,j}^{(\ell)}) = \sum_{\substack{(q,v)\\|q|<|v|}} \left(\omega_{i,j}^{(\ell)} + |q| - |v|\right)^2 \underbrace{P\left(\lambda_{i\to j}^{(\ell-1)} = q \,\cap\, \gamma_{\bar{e},j\to i}^{(\ell)} = v\right)}_{\theta_{i,j}^{(\ell)}(q,v)} \tag{15}$$

with $(q, v) \in [-Q, Q]^2$ and $\theta_{i,j}^{(\ell)}(q, v)$ being the joint probability mass function of the $(\lambda_{i\to j}^{(\ell-1)}, \gamma_{\bar{e},j\to i}^{(\ell)})$ random variables. If these variables are independent, we have $\theta_{i,j}^{(\ell)}(q,v) = P_\lambda^{(\ell-1)}(q) \times P_{\gamma_{\bar{e}}}^{(\ell)}(v)$. The message dependencies will be further discussed in the next sub-section. Problem (14) is convex, having solution:

$$\frac{d\epsilon(\bar{\omega}_{i,j}^{(\ell)})}{d\bar{\omega}_{i,j}^{(\ell)}} = 2\sum_{\substack{(q,v)\\|q|<|v|}} \left(\bar{\omega}_{i,j}^{(\ell)} + |q| - |v|\right)\theta_{i,j}^{(\ell)}(q,v) = 0, \tag{16}$$

$$\implies \bar{\omega}_{i,j}^{(\ell)} = \frac{\sum_{\substack{(q,v)\\|q|<|v|}} \left(|v| - |q|\right)\theta_{i,j}^{(\ell)}(q,v)}{\sum_{\substack{(q,v)\\|q|<|v|}} \theta_{i,j}^{(\ell)}(q,v)}, \tag{17}$$

where $\bar{\omega}_{i,j}^{(\ell)}$ is the optimal real-valued SM offset, and $\omega_{i,j}^{(\ell)} = \lfloor \bar{\omega}_{i,j}^{(\ell)} + 1/2 \rfloor$.

### *C. On the message dependencies*

DE equations are derived based on the fact that all messages are independent, a consequence of the extrinsic exclusion rule and the cycle-free assumption. However, when the $\gamma$-update rule includes the extrinsic message, the message independence assumption no longer holds. To illustrate this, we study how messages propagate while being exchanged with the first VN in a cycle-free code. For ease of notation, we consider that this VN is connected to the first $d_v$ CNs in the Tanner graph. By noting that, in a SM decoder without offset compensation, $\gamma_{e,j\to i}^{(\ell)} = \mathfrak{s}(\lambda_{i\to j}^{(\ell-1)}) \times f([\lambda_{i\to j}^{(\ell-1)}, \gamma_{\bar{e},j\to i}^{(\ell)}])$, we have

$$\lambda_{1\to i}^{(\ell)} = L_1 + \sum_{\substack{j=1\\i\neq j}}^{d_v} \underbrace{\mathfrak{s}(\lambda_{1\to j}^{(\ell-1)}) f\left([\lambda_{1\to j}^{(\ell-1)}, \gamma_{\bar{e},j\to 1}^{(\ell)}]\right)}_{\gamma_{e,j\to 1}^{(\ell)}}. \tag{18}$$

Through recursion, all $\lambda_{1\to j}^{(\ell-1)}$ variables are correlated with $\gamma_{\bar{e},j\to 1}^{(\ell-1)}$ and $L_1$. Consequently, the $\gamma_{e,j\to 1}^{(\ell)}$ variables are not independent, and (10) cannot be applied to derive $P_\lambda^{(\ell)}$. Performing DE for the SM decoding algorithm requires conditioning the message PMFs on each message value in the computation tree. Since the number of possibilities grows exponentially with $\ell$, the number of message bits and check node degree, this method is too computationally expensive.

### *D. Optimisation for the first iterations*

When $\ell \in [1, 2]$, (11) is valid and can be used to derive $P_{\gamma_{\bar{e}}}^{(\ell)}$. Furthermore, with $\lambda_{i\to j}^{(1)}$ and $\gamma_{\bar{e},i\to j}^{(2)}$ being independent,

$\theta_{i,j}$ can be easily calculated and the optimal offsets can be deduced for the first two iterations through (16). However, $P_\lambda^{(1)}$ cannot be obtained with (10) since $\gamma_{e,j\to i}^{(1)}$ is correlated with $L_i$, as shown in (18). Instead, we have

$$P_\lambda^{(1)}(k) = \sum_{l=-Q}^{Q} P_\lambda^{(0)}(l)\Big(\underset{d_v-1}{\circledast} \boldsymbol{P}^{(1)}_{\boldsymbol{\gamma_e|L}}\Big)(k-l), \tag{19}$$

where $\boldsymbol{P}^{(1)}_{\boldsymbol{\gamma_e|L}} = [P^{(1)}_{\gamma_e|L}(-Q|l), \dots, P^{(1)}_{\gamma_e|L}(Q|l)]$ is the PMF vector of the SM C2V messages when the extrinsic message value (corresponding to the channel LLR) is $l$:

$$P^{(1)}_{\gamma_e|L}(k|l) = \begin{cases} P^{(1)}_{\gamma_{\bar{e}}}(k) & |k| \le |l|, \\ \sum_{m=|l|+1}^{Q} P^{(1)}_{\gamma_{\bar{e}}}(\mathfrak{s}(k)m) & |k| = \min\left(|l| + \omega^{(\ell)}_{i,j}, Q\right), \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Finally, the offset of the remaining iterations $\ell > 2$ are obtained by linearly extrapolating the real-valued offsets deduced by the method described in this section for the first 2 iterations: $\bar{\omega}^{(\ell)}_{i,j} = \left(\bar{\omega}^{(2)}_{i,j} - \bar{\omega}^{(1)}_{i,j}\right) \times (\ell - 1) + \bar{\omega}^{(1)}_{i,j}$.

## V. Simulation results

### A. Simulation setup

To validate the optimization methods, MC simulation are performed to measure the error correction performance for two different codes: i) a (1723,2048)-regular LDPC code from the 802.3an-2006 10 Gb/s Ethernet (10GE) standard [20]; ii) a 5$^{\text{th}}$ generation cellular networks (5G) code with base graph index 1, code length $N = 16128$, lifting size $Z = 384$ and code rate 0.5238 [21]. Message values for the 10GE decoders are quantized on 7 bits and passed on a flooding schedule for a maximum of 40 iterations. For the 5G decoders, messages are quantized on 6 bits, and passed on a flooding schedule for a maximum of 40 iterations.

We propose comparing four offset optimization methods for the SM decoder. All methods use CD as described in Section III-B to optimize $(\eta, \alpha, \boldsymbol{\Omega_G})$ parameters. These parameters, and subsequently the SM offset vectors determined by the stochastic searches, are optimized for $\alpha_{\max} = 10$, $\omega_{\max} = 10$ and $\eta \in [0.25, 0.5, 0.75, 1]$ to facilitate hardware implementation. The differences between the methods resides in how (5) is solved. The first method (M1), called *fixed offset*, only optimizes a single offset $\omega$ following approach (i) described in Section III-A. The second method (M2) optimizes $\boldsymbol{\Omega_G}$ following approach (iii), where (5) is solved using *genetic algorithm* [22] for the regular 10GE code. For the 5G codes, SM offsets are selected based on the extended variable weight (EVW) method reported in [11], then $(\eta, \alpha)$ are optimized using CD. This method is referred to as CD-EVW in this paper. The third method (M3) is the proposed WSA performed with $L = 3$, $p = 1$, with (8) solved through exhaustive search for the 10GE code and genetic algorithm for the 5G code using the following group setup: $G_1 = \{3\}$, $G_2 = \{6, 7, 8, 9, 10\}$ and $G_3 = \{19\}$. The last method (M4) corresponds to the DE-based optimization proposed in Section IV.

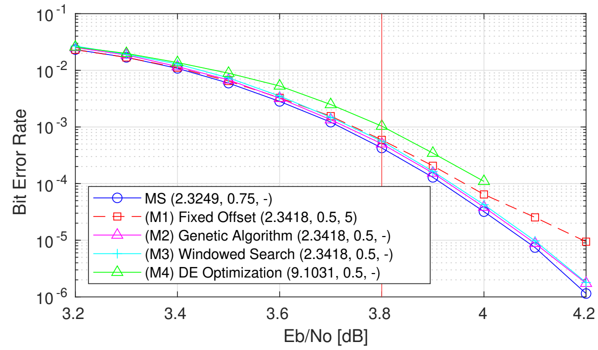


Fig. 2. Comparison of BER performance for SM decoders against optimized MS for the 802.3an-2006 code. Triples in the legend correspond to ($\alpha$, $\eta$, $\omega_{SM}$).

For the 10GE code, problem (5) is solved when considering 25 decoding iterations, for all methods. The remaining iterations use the final SM offset gathered during the optimization procedure. For the 5G code, WSA is applied for 10 decoding iterations, and the modified Akima method was performed to extrapolate SM offsets from 10 to 40. For all instances where the genetic algorithm is used, the parameters are set to a population of 200 candidates, including 10 elites, from which 152 are selected for crossover.

### B. Results for the 10GE code

Figure 2 shows the BER performance for the regular 10GE code when considering the 4 offset optimization methods for the SM decoder. BER results for the MS decoder are also shown as reference. The legend indicates the optimal parameters, optimized at $\zeta = 3.8$ dB for all methods, as indicated by the red vertical line in the figure.

The results for the stochastic searches show that the difference in performance between optimized varying offset methods (M2) and (M3) is negligible. However, for $\zeta > 4$ dB, the fixed offset method is outperformed by both (M2) and (M3), with a gap of 0.1 dB observed at $10^{-5}$ BER. Concerning the DE method (M4), we observe a performance loss compared to other optimization methods. This is likely due to the fact that only SM offsets of the first two iterations are optimized, while we approximate the remaining ones by linear extrapolation.

We measured the execution times of each method on the same computer (at $\zeta = 3.8$ dB), and obtained the following results: (M1) 14min; (M2) 20h; (M3) 314h; (M4) 0.7s. The DE method (M4) is by far the fastest method. Particularly, the compute time is constant for any $\zeta$ value, whereas MC approaches require more simulation trials with decreasing noise power. This opens the possibility to re-design the offsets "on-the-fly", when the decoding parameters change.

### C. Results for the 5G code

Figure 3 depicts the BER of SM decoders for the 5G code. For the MS decoder, we have $\beta = 1$, applied at each iteration. Parameters are optimized at the following $E_b/N_o$ values: $\zeta = 1.1$ dB for the MS decoder (blue vertical line), $\zeta = 2.8$ dB for

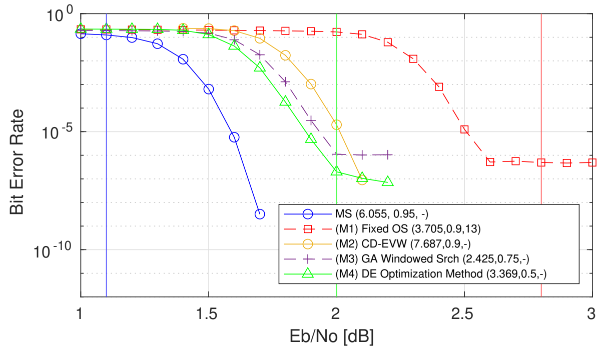


Fig. 3. Comparison of SM decoding methods against an MS decoder; performed using a 5G code BG1 [N=16128, R=0.5238]. Triples in the legend correspond to ($\alpha$, $\eta$, $\omega_{SM}$).

the SM fixed offset method and $\zeta = 2$ dB for the CD-EVW, WSA and DE-based methods.

There is a significant gain in error performance for varying versus fixed offset SM decoding in the 5G code presented. The SM offsets obtained with WSA outperforms the ones obtained using CD-EVW by $0.1 - 0.2$ dB in the Waterfall region. Furthermore, contrary to the 10GE code, the DE analysis method (M4) significantly outperforms both CD-EVW and WSA. It can be explained by the fact that SM offsets are optimized separately for each edge of the base graph. This additional degree of freedom provides gains since the 5G code has several edge types and CN/VN degrees. In addition, the DE method has the lowest execution time.

## VI. Conclusion

In this paper, we tackled the problem of optimizing second-minimum emulation offset values for SM decoders. These decoders require less hardware resources than typical MS decoders, but their performance depends heavily on well-optimized SM offsets. We first presented a general formulation of the problem and proposed a simplification with the coordinate descent algorithm. We then solved for offset values by proposing two methods: 1) a Monte Carlo-based method referred to as WSA 2) A DE-based analytical method. We evaluate these proposed methods using two different codes (5G and 10G Ethernet standards) and compare with existing optimization methods. We show that, for the 5G code, large BER gains are observed, particularly for the DE method, which also requires less computation time. This method optimizes the SM offsets for only the first two iterations, while the remaining ones are extrapolated. Therefore, we believe that the performance gap between MS and SM decoders can be further reduced. This encourages further investigation to improve DE analysis for SM decoders.

## Acknowledgment

This work was supported by grant 2021-NC-286323 of Fonds de recherche du Québec – Nature et technologie.

## References

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

[2] K. Zhu and Z. Wu, "Comprehensive Study on CC-LDPC, BC-LDPC and Polar Codes," in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Apr. 2020, pp. 1–6.

[3] J. Nadal and A. Baghdadi, "FPGA based design and prototyping of efficient 5G QC-LDPC channel decoding," in *2020 International Workshop on Rapid System Prototyping (RSP)*, Sept. 2020.

[4] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, July 2020.

[5] A. Darabiha, A. Carusone, and F. Kschischang, "A bit-serial approximate min-sum LDPC decoder and FPGA implementation," in *2006 IEEE International Symposium on Circuits and Systems*, May 2006.

[6] H. Lopez, H.-W. Chan, K.-L. Chiu, P.-Y. Tsai, and S.-J. J. Jou, "A 75-gb/s/mm2 and energy-efficient LDPC decoder based on a reduced complexity second minimum approximation min-sum algorithm," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 926–939, Apr. 2020.

[7] C. Zhang, Z. Wang, J. Sha, L. Li, and J. Lin, "Flexible LDPC decoder design for multigigabit-per-second applications," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 116–124, Mar. 2009.

[8] F. Yi and P. Wang, "Low complexity decoding algorithm of QC-LDPC code," in *2010 IEEE Asia-Pacific Services Computing Conference*, Dec. 2010, pp. 531–534.

[9] S. Hemati, F. Leduc-Primeau, and W. J. Gross, "A relaxed min-sum LDPC decoder with simplified check nodes," *IEEE Communications Letters*, vol. 20, no. 3, pp. 422–425, Jan. 2016.

[10] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 2150–2158, Feb. 2014.

[11] V. L. Petrović and D. M. El Mezeni, "Reduced-complexity offset min-sum based layered decoding for 5G LDPC codes," in *2020 28th Telecommunications Forum (TELFOR)*, Nov. 2020, pp. 1–4.

[12] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[13] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. on Communications*, vol. 47, no. 5, pp. 673–680, May 1999.

[14] H. Akima, "A new method of interpolation and smooth curve fitting based on local procedures," *J. ACM*, vol. 17, no. 4, p. 589–602, Oct. 1970.

[15] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, Mar. 2015.

[16] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, "Computer methods for mathematical computations," *ZAMM - Journal of Applied Mathematics and Mechanics*, vol. 59, no. 2, pp. 141–142, 1979.

[17] R. P. Brent, *Algorithms for Minimization without Derivatives*, 1st ed. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

[18] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[19] A. Balatsoukas-Stimming and A. Burg, "Density evolution for min-sum decoding of LDPC codes under unreliable message storage," *IEEE Communications Letters*, vol. 18, no. 5, pp. 849–852, Apr. 2014.

[20] IEEE, 802.3an, " Standard for Information Technology – LAN/MAN - CSMA/CD Access Method and Physical Layer Specifications - Parameters for 10 Gb/s Operation, Type 10GBASE-T ," pp. 1–181, Sept. 2006.

[21] 3GPP TS 38.212, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding," Aug. 2021.

[22] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, June 1994.