

王坤, 王娟, 陈锋. 一种引入信息量演变的偏移最小和译码算法[J]. 智能计算机与应用, 2025, 15(10): 96-102. DOI: 10.20169/j. issn. 2095-2163. 251014

# 一种引入信息量演变的偏移最小和译码算法

王 坤, 王 娟, 陈 锋

(西安建筑科技大学 机电工程学院, 西安 710055)

**摘要:** 为解决传统偏移最小和(Offset Min-Sum, OMS)译码算法中偏移因子计算不准确和硬件计算复杂度高的问题, 提出了一种基于信息量演变理论的偏移最小和(Information Evolution-based Offset Min Sum, IE-OMS)译码算法。IE-OMS 算法采用信息量演变理论建立了计算偏移因子的数学模型, 通过概率集中度函数对节点间更新的信息均值进行计算, 实现了在迭代过程中对偏移因子的精确调整。此外, IE-OMS 算法引入了加权平均法对偏移因子的动态变化进行处理, 确保硬件在每次迭代过程中只需保存一个唯一的偏移因子。与传统 OMS 算法相比, IE-OMS 算法不仅降低了误码率和平均迭代次数, 而且在硬件实现上减少了约 30.1% 的逻辑元件和 33.33% 的内存资源。仿真结果表明, IE-OMS 算法在提高译码性能的同时降低了硬件计算复杂度, 具有更好的性能优势和应用潜力。

**关键词:** 低密度奇偶校验码; 偏移最小和算法; 信息量演变理论

中图分类号: TN911.22 文献标志码: A 文章编号: 2095-2163(2025)10-0096-07

## An Offset Minimum-Sum decoding algorithm introducing the evolution of information content

WANG Kun, WANG Juan, CHEN Feng

(School of Mechanical and Electrical Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

**Abstract:** In order to solve the problems of inaccurate offset factor calculation and high hardware computational complexity in the traditional Offset Min-Sum (OMS) decoding algorithm, an Information Evolution-based Offset Min-Sum (IE-OMS) decoding algorithm based on information evolution theory is proposed. IE-OMS algorithm adopts the information evolution theory to establish a mathematical model for calculating the offset factor, and calculates the information mean value of the update between nodes through the probability concentration function, which realizes the accurate adjustment of the offset factor in the iterative process. In addition, the IE-OMS algorithm introduces a weighted average method to deal with the dynamic change of the offset factor, which ensures that the hardware only needs to save a unique offset factor during each iteration. Compared with the traditional OMS algorithm, the IE-OMS algorithm not only reduces the BER and the average number of iterations, but also reduces about 30.1% of logic components and 33.33% of memory resources in the hardware implementation. Simulation results show that the IE-OMS algorithm has better performance advantages and application potentials by improving the decoding performance while reducing the hardware computational complexity.

**Key words:** low-density parity check code; Offset Minimum-Sum algorithm; information evolution theory

## 0 引言

低密度奇偶校验码(Low-Density Parity-Check codes, LDPC)是由Gallager<sup>[1]</sup>在其论文中首次提出的。LDPC码不仅具有接近香农极限的优异解码性能, 还具有并行的解码结构、突出的纠错能力和适中

的解码复杂度等优点。因此, LDPC码在WiMax、IEEE 802.11n等通信系统中得到了大范围的应用<sup>[2-3]</sup>, 成为通信领域中备受关注的一种编码方案。

LDPC码的置信传播(Belief Propagation, BP)译码算法<sup>[4]</sup>通过校验节点和变量节点之间的信息更新实现了高效的译码性能, 但存在硬件资源需求较

基金项目: 陕西省教育厅产业化项目(21JC018); 陕西省重点产业链项目(2023-ZDLGY-24)。

作者简介: 王 坤(1997—), 女, 硕士研究生, 主要研究方向: 激光通信; 陈 锋(1999—), 男, 硕士研究生, 主要研究方向: 激光通信。

通信作者: 王 娟(1978—), 女, 博士, 教授, 主要研究方向: 大气激光通信。Email: juanwang168@gmail.com。

收稿日期: 2024-01-09

哈尔滨工业大学主办 ◆ 学术研究与应用

高的问题。与 BP 算法相比, 最小和 (Minimum Sum, MS) 译码算法<sup>[5]</sup> 在节点信息更新过程中采用了绝对值和最小值的计算操作, 减小了计算难度, 但同时也带来性能下降的问题。为了弥补 MS 算法的性能损失, 归一化最小和 (Normalized Minimum Sum, NMS) 译码算法<sup>[6]</sup> 和偏移最小和 (Offset Minimum Sum, OMS) 译码算法<sup>[7]</sup> 对 MS 算法进行了改进。具体就是分别在检验节点信息更新过程中引入了尺度因子和偏移因子, 从而实现了译码性能的提升。文献[8]通过引入加权平均系数对 NMS 算法迭代过程中的尺度因子进行自适应修正, 降低了硬件存储需求。文献[9]中 BP 译码算法在分层译码架构的基础上引入了信息量演变 (Information Evolution, IE) 理论<sup>[10]</sup> 对算法进行优化, 相较于分层最小和 (Layered Min Sum, Layered-MS) 译码算法<sup>[11]</sup>, 该算法的译码性能更卓越, 但硬件计算复杂度较高。

上述算法在考虑提高译码性能的同时, 不能满足硬件低计算复杂度的要求。为了同时满足高译码性能和低计算复杂度的需求, 提出一种基于信息量演变理论的偏移最小和 (Information Evolution based Offset Min Sum, IE-OMS) 译码算法。IE-OMS 算法利用信息量演变理论对迭代过程中的偏移因子进行了精确调整, 并采用了加权平均法, 使得硬件在每次迭代过程中只需保存一个唯一的偏移因子, 在提高译码性能的同时降低了硬件计算复杂度。仿真结果表明, 相较于传统的译码算法, IE-OMS 算法不仅降低了误码率和平均迭代次数, 而且在硬件实现上减少了资源消耗。

## 1 LDPC 码的译码算法

### 1.1 对数似然比置信传播算法

BP 算法是一种基于图模型的迭代算法<sup>[12]</sup>, 通过将节点间的条件概率转化为相应的信息概率来进行信息传递, 并不断迭代更新节点的状态, 直到达到收敛条件为止。BP 算法在处理大规模图模型时需进行复杂的乘法计算, 导致硬件计算复杂度高, 并且消耗大量的时间和资源。为解决这一问题, 对数似然比置信传播 (Log Likelihood Ratio – Belief Propagation, LLR-BP) 译码算法<sup>[13]</sup> 被提出。

LLR-BP 算法使用简单的加法运算来代替 BP 算法中的乘法计算, 显著降低了硬件计算复杂度, 缩短了解码时间, 使得 LLR-BP 算法在实际应用中具有更好的可行性。LLR-BP 译码算法的具体步骤<sup>[14]</sup> 如下:

(1) 初始化。经编码和 BPSK 调制后, 在 AWGN 信道下所得信息  $y_i$  的对数似然比 (Log-Likelihood Ratio, LLR) 为:

$$L^{(l)}(v_i) = L(y_i) = \ln \frac{1 + \exp\left(\frac{2y_i}{\delta^2}\right)}{1 + \exp\left(-\frac{2y_i}{\delta^2}\right)} = \frac{2y_i}{\delta^2} \quad (1)$$

其中,  $L(v_i)$  表示变量节点  $v_i$  向校验节点  $c_j$  传递的边信息;  $l$  表示第  $l$  次迭代;  $y_i (i = 1, 2, \dots, n)$  表示经过 AWGN 信道后的信息序列;  $\delta^2$  表示噪声功率。

#### (2) 迭代过程

##### ① 校验节点更新。推得的公式为:

$$L^{(l)}(c_j) = 2 \tanh^{-1} \left( \prod_{i' \in N(j)/i} \tanh \left( \frac{1}{2} L^{(l)}(v_{i'}) \right) \right) \quad (2)$$

其中,  $L(c_j)$  表示校验节点  $c_j$  向变量节点  $v_i$  传递的边信息;  $N(j)/i$  表示除了变量节点  $v_i$  外, 校验节点  $c_j$  的全部相邻变量节点的集合。

##### ② 变量节点更新。推得的公式为:

$$L^{(l)}(v_i) = L(y_i) + \sum_{j' \in N(i)/j} L^{(l)}(c_{j'}) \quad (3)$$

其中,  $N(i)/j$  表示除了校验节点  $c_j$  外, 变量节点  $v_i$  的全部相邻校验节点的集合。

##### ③ 后验概率。推得的公式为:

$$L^{(l)}(q_i) = L(y_i) + \sum_{j \in N(i)} L^{(l)}(c_j) \quad (4)$$

其中,  $N(i)$  表示变量节点  $v_i$  的全部相邻校验节点的集合。

##### ④ 判决。用到的公式为:

$$m_i = \begin{cases} 0, & L^{(l)}(q_i) > 0 \\ 1, & L^{(l)}(q_i) \leq 0 \end{cases} \quad (5)$$

其中,  $L^{(l)}(q_i) > 0$  时,  $m_i = 0$ ;  $L^{(l)}(q_i) \leq 0$  时,  $m_i = 1$ , 进而得到码字  $m(i) = (m_1, m_2, \dots, m_n)$ 。 $\mathbf{H}$  表示校验矩阵,  $\mathbf{m}^T$  表示校验矩阵的第  $m$  列, 当  $\mathbf{Hm}^T = 0$  时, 停止迭代过程, 将  $m(i)$  作为译码结果输出; 否则  $l \leftarrow l + 1$  继续迭代过程, 直到  $l$  达到最大迭代次数, 并给出译码失败标志。

### 1.2 最小和译码算法及其简化算法

LLR-BP 算法在计算校验节点信息时使用了  $\tanh$  函数, 该运算在硬件实现上需进行传统的查表方式, 会消耗大量的逻辑资源和存储空间。为了解决这一问题, MS 译码算法被提出, 其核心思想是利用辅助函数最小和运算来简化 LLR-BP 算法中  $\tanh$  函数的运算<sup>[15]</sup>。MS 算法的校验节点更新公式为:

$$L^{(l)}(c_{ji}) = \prod_{i' \in N(j)/i} \operatorname{sgn}(L^{(l)}(v_{i'j})) \times \min_{i' \in N(j)/i} (|L^{(l)}(v_{i'j})|) \quad (6)$$

MS 算法虽然降低了校验节点更新公式计算的复杂度,但也带来了译码性能下降的问题。LLR-BP 算法的校验节点信息计算中使用了  $\tanh$  函数和  $\operatorname{arctanh}$  函数,其中  $\tanh$  函数输入范围为  $(-1, 1)$ ,  $\operatorname{arctanh}$  函数输入范围为  $(-\infty, +\infty)$ 。经过  $\tanh$  函数运算,校验节点信息取值范围限制在  $(-1, 1)$  之间,多次乘法运算后,其绝对值会进一步减小。再经过  $\operatorname{arctanh}$  函数进行恢复时,其结果可能会偏向于较大的值,从而使 MS 算法中校验节点信息值偏高<sup>[16]</sup>。

为解决 MS 算法中信息值过高估计的问题,OMS 译码算法被提出。在 OMS 算法的校验节点更新公式中,通过减去偏移因子  $\beta$  来减少 MS 算法中过高估计的问题。OMS 算法相对于 MS 算法的校验节点更新公式为:

$$L^{(l)}(c_{ji}) = \prod_{i' \in N(j)/i} \operatorname{sgn}(L^{(l)}(v_{i'j})) \times \max\left\{\left(\min_{i' \in N(j)/i} (|L^{(l)}(v_{i'j})|)\right) - \beta, 0\right\} \quad (7)$$

在 OMS 算法的迭代过程中,偏移因子  $\beta$  保持固定不变会引入误差导致译码性能的损失。为了解决这个问题,IE-OMS 算法利用信息量演变理论对迭代过程中的偏移因子  $\beta$  进行精确调整,并通过加权平均法实现了硬件在每次迭代过程中只需保存一个唯一的偏移因子  $\beta$  的设计,从而在提高译码性能的同时降低硬件计算复杂度。

## 2 改进的偏移最小和译码算法

### 2.1 信息量演变理论

信息量演变理论是基于概率图模型对纠错码在迭代译码过程中的性能进行优化。通过在概率图模型上进行迭代来分析节点信息的更新和传递,可计算出每个节点信息的集中度以及概率空间的演变过程。通过分析信息的演变过程,可发现节点信息集中度过低或者过高的情况,从而为纠错码的优化提供理论基础<sup>[17]</sup>。可设:

$$\begin{aligned} v_i^{(l)} &= \lg \frac{p(x=+1|y)}{p(x=-1|y)} \\ c_j^{(l)} &= \lg \frac{p(x=+1|z)}{p(x=-1|z)} \end{aligned} \quad (8)$$

其中,  $x$  表示节点信息的值,  $y$  和  $z$  表示迭代更新后该节点所有信息。

IE-OMS 算法的信息量演变是通过校验节点信

息的概率集中度函数发生变化来实现的。该函数描述了校验节点周围变量节点的概率分布情况,反映了校验节点的信息集中度。概率集中度函数的演变表达式如下:

$$F(p_{v_i}) = \exp(\beta \cdot \sum_{j \in N(i)} p_{c_j} H_{ji}) \quad (9)$$

其中,  $p_{v_i}$  表示变量节点  $v_i$  的概率值;  $\beta$  表示可调节的偏移因子;  $p_{c_j}$  表示校验节点  $c_j$  的概率值;  $H_{ji}$  表示校验矩阵  $\mathbf{H}$  中第  $j$  个校验节点与第  $i$  个变量节点之间的连接权重。

由式(9)可知,偏移因子  $\beta$  被用来调整概率集中度函数的变量节点和所有与其相邻的校验节点之间的依赖关系强度。当偏移因子  $\beta$  较小时,各节点信息的概率分布较均匀,有利于快速收敛;当偏移因子  $\beta$  较大时,各节点信息的概率分布会更加集中,有利于算法准确度的提高。然而,如果偏移因子  $\beta$  过小,会导致节点的信息集中程度不够,影响算法的译码性能;相反,如果偏移因子  $\beta$  过大,会导致算法迭代次数增多、收敛速度变慢,出现震荡或发散等问题。因此,在 IE-OMS 算法的迭代过程中,选择合适的偏移因子  $\beta$  至关重要,可确保算法具有较好的收敛速度和译码性能。

### 2.2 首次迭代中偏移因子的计算模型

在首次迭代中,校验节点到变量节点的信息更新是基于初始化信道参数进行的。在 LLR-BP 算法和 MS 算法中,校验节点到变量节点的信息更新方程是利用初始化信道参数进行计算的。利用 LLR-BP 算法和 MS 算法的校验节点更新方程建立模型,可准确计算出首次迭代中的偏移因子  $\beta$ 。

令  $h_1$  和  $h_2$  分别表示 LLR-BP 算法和 MS 算法的校验节点到变量节点的信息更新方程,如下所示:

$$h_1 = 2 \tanh^{-1} \left( \prod_{i' \in N(j)/i} \tanh \left( \frac{1}{2} L^{(l)}(v_{i'j}) \right) \right) \quad (10)$$

$$h_2 = \prod_{i' \in N(j)/i} \operatorname{sgn}(L^{(l)}(v_{i'j})) \cdot \min_{i' \in N(j)/i} (|L^{(l)}(v_{i'j})|) \quad (11)$$

LLR-BP 算法和 MS 算法的概率集中度函数表达式分别如下所示:

$$F(h_1) = \frac{1}{\sqrt{2\pi}\sigma} \left( \int_{-\infty}^{+\infty} h_1 dL^{(l)}(v_{ji}) \right)^{d_c-1} \quad (12)$$

$$F(h_2) = \frac{1}{\sqrt{2\pi}\sigma} \left( \int_{-\infty}^{+\infty} h_2 dL^{(l)}(v_{ji}) \right)^{d_c-1} \quad (13)$$

其中,  $\sigma = \sqrt{2N_0}$ ,  $d_c$  表示校验节点信息的度。

结合  $F(h_1)$  和  $F(h_2)$ , LLR-BP 算法和 MS 算法中校验节点到变量节点更新的信息均值定义如下:

$$E(|h_1|) = \int_{-\infty}^{+\infty} |h_1| \cdot F(h_1) dh_1 \quad (14)$$

$$E(|h_2|) = \int_{-\infty}^{+\infty} |h_2| \cdot F(h_2) dh_2 \quad (15)$$

根据  $h_1$  和  $h_2$ , 通过概率集中度函数可计算出 LLR-BP 算法和 MS 算法在首次迭代中校验节点到变量节点更新的信息均值表达式, 具体如下:

$$E(|h_1|) = \sqrt{\frac{2}{\pi\sigma}} \sum_{n=1}^{\infty} \frac{E\left(\tanh\left(\frac{L^{(l)}(v_{ij})}{2}\right)^{2n-1}\right)^{d_c-1}}{2n-2} \quad (16)$$

$$E(|h_2|) \approx \frac{1}{\sqrt{2\pi\sigma}} \int_0^{\mu} \left[ 1 - F\left(\frac{\mu-y}{\sigma}\right) + F\left(\frac{\mu+y}{\sigma}\right) \right]^{d_c-1} dy \quad (17)$$

其中,  $\sigma = \sqrt{2N_0}/2$ ,  $\mu = 4/N_0$ 。

基于  $E(|h_1|)$  和  $E(|h_2|)$  来确定首次迭代中偏移因子  $\beta$  值, 如下所示:

$$\beta = E(|h_1|) - E(|h_2|) \quad (18)$$

在首次迭代后, 变量节点信息及其概率集中度函数会随着迭代的进行而逐渐发生变化。传统 OMS 算法中的偏移因子  $\beta$  仅是基于第一次迭代计算得到的, 无法准确应用于后续迭代。为了克服偏移因子  $\beta$  的局限性并提升译码效率, IE-OMS 算法在后续迭代过程中采用信息量演变理论来动态调整偏移因子  $\beta$ , 以更好地适应概率分布的变化, 从而提高 IE-OMS 算法的译码性能。

### 2.3 后续迭代中偏移因子的计算模型

利用信息量演变理论对校验节点和变量节点之间的信息传递进行量化和概率变化分析, 可计算出每个节点状态的概率分布, 进而得到相应的概率集中度函数。通过概率集中度函数可以计算出每个节点的信息均值, 并对 2 次迭代之间的信息均值差异进行比较, 来调整偏移因子  $\beta$  的取值。

令  $h_3$  表示 OMS 算法的校验节点到变量节点的信息更新表达式, 如下所示:

$$h_3 = \prod_{i' \in N(j)/i} \operatorname{sgn}(L^{(l)}(v_{ij})) \times \max\left\{\left(\min_{i' \in N(j)/i} (|L^{(l)}(v_{ij})|)\right) - \beta, 0\right\} \quad (19)$$

为了简化上述表达式, 令上式中:

$$W = \max\left\{\left(\min_{i' \in N(j)/i} (|L^{(l)}(v_{ij})|)\right) - \beta, 0\right\} \quad (20)$$

结合信息演变方程(8), 节点信息的离散概率分布函数的定义如下:

$$P(X=x) = \sum p(x) \cdot \log(p(x)) \quad (21)$$

其中,  $P(X=x)$  表示随机变量  $X$  取值为  $x$  的概

率;  $p(x)$  表示随机变量  $X$  取值为  $x$  的概率质量;  $x$  表示随机变量  $X$  的一个具体取值。

在后续节点信息的传递过程中, 节点信息的概率分布函数表达式如下所示:

$$F'(x) = P(W \leq x) = \{1 - [1 - P(|L^{(l)}(v_{ij})| \leq x + \beta)]\}^{d_c-1} \cdot P(x \geq 0) \quad (22)$$

节点信息的概率集中度函数表达式为:

$$F(x=i) = F'(x=i) - F'(x=i-s) \quad (23)$$

其中,  $s$  表示量化步长。

研究中, 校验节点到变量节点更新的信息均值  $E(|h_3|)$  具体表达式如下所示:

$$E(|h_3|) = E\left[\max\left(\min_{i' \in N(j)/i} (|L^{(l)}(v_{ij})| - \beta, 0)\right)\right] = E(W) = \sum_i i \cdot F(x=i) \quad (24)$$

根据信息量演变模型, 在后续每一次迭代中可通过节点间更新的信息均值差异来选择合适的偏移因子  $\beta_l$ , 可将偏移因子  $\beta_l$  定义为一个关于信息均值的函数, 如下所示:

$$\beta_l = \begin{cases} E(|h_2|) - E(|h_1|), & l = 1 \\ E(|h_3|) - E(|h_1|), & l \geq 2 \end{cases} \quad (25)$$

其中,  $\beta_l$  表示第  $l$  次迭代时的偏移因子值。

在每次迭代中, 通过计算节点间更新的信息均值, 并将其与之前的信息均值进行比较, 来对偏移因子  $\beta$  进行动态调整。如果当前信息均值较大, 可增大偏移因子  $\beta$ ; 如果当前信息均值较小, 可减小偏移因子  $\beta$ 。通过对偏移因子  $\beta$  的调整, 使得在后续每次迭代中的信息均值保持在一个相对稳定的范围内, 从而确保 IE-OMS 算法的收敛速度和稳定性。

### 2.4 基于加权平均化的偏移因子处理

如果将每次迭代计算得到的偏移因子  $\beta$  存储在寄存器中, 并在相应的迭代次数时检索使用对应的偏移因子  $\beta$ , 可以进一步提高解码性能, 但此方法需要大量存储位。为了确保提高译码性能的同时降低存储需求, 通过使用加权平均化<sup>[18]</sup> 来处理偏移因子  $\beta_l$ , 并将加权平均化后的结果作为每次迭代时所对应的唯一偏移因子  $\beta$ , 可由下式计算求出:

$$\beta = \frac{1}{6} \cdot u_i \cdot \sum_{i=1}^6 \mu_i \beta_i \quad (26)$$

其中,  $\mu_i$  表示第  $i$  个加权平均系数。

随着迭代次数的增加, IE-OMS 算法的修正能力逐渐下降。为更好地适应修正能力的变化, 可依据迭代次数调整加权系数, 具体的加权系数设置为:  $\mu_1 = 0.20$ 、 $\mu_2 = 0.15$ 、 $\mu_3 = 0.10$ 、 $\mu_4 = 0.10$ 、 $\mu_5 = 0.05$ 、 $\mu_6 = 0.01$ 。在初始迭代阶段的加权系数较

大,以使修正能力在开始时得到加强;在后续迭代阶段的加权系数逐渐减小,以适应修正能力的下降。

在每次迭代中,通过加权平均化方法处理偏移因子 $\beta_i$ ,可以更好地捕捉偏移因子 $\beta$ 的动态变化。同时,每次迭代只需一个累加器和一个计数器的存储位来计算加权平均后偏移因子 $\beta$ ,并将其作为每次迭代所对应的唯一偏移因子 $\beta$ ,使得 IE-OMS 算法在提高译码性能的同时更加节省硬件存储资源。

### 3 结果分析

#### 3.1 译码性能仿真分析

在本次仿真中,使用 IEEE 802.11n 标准下的 LDPC 码<sup>[19]</sup>参数为(8448,4224)。采用 Matlab 软件进行二进制相移键控(Binary Phase Shift Keying, BPSK)调制方式下的加性高斯白噪声(AWGN)信道仿真测试,测试了在不同信噪比下算法的误码率和平均迭代次数,每个信噪比下最多进行 20 次迭代,并统计了 2 200 个误帧数。图 1 为 IE-OMS 算法与其它 6 种不同的译码算法在不同 SNR 下的误码率对比。

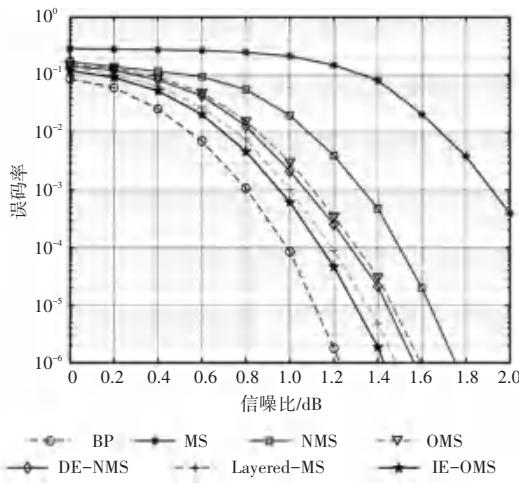


图 1 不同算法的误码率对比图

Fig. 1 Comparison of Bit Error Rates of different algorithms

图 1 中,NMS 算法、OMS 算法、DE-NMS 算法、Layered Min Sum 算法和 IE-OMS 算法都是对 MS 算法的改进。由图 1 可知,改进后的 5 种译码算法相较于 MS 算法有较低的误码率,在译码性能上有明显的提高。Layered Min Sum 算法和 IE-OMS 算法的译码性能十分相似,但 IE-OMS 算法的性能更接近 BP 算法。这是因为在 IE-OMS 算法迭代更新过程中,通过引入精确调整后的偏移因子 $\beta$ 来调整节点信息更新的权重,使得算法能够灵活地适应不同误码率的要求,并提供更接近于理论极限的译码性能,实现高性能译码。图 2 为 IE-OMS 算法与其

它 6 种不同的译码算法在不同 SNR 下的平均迭代次数对比。

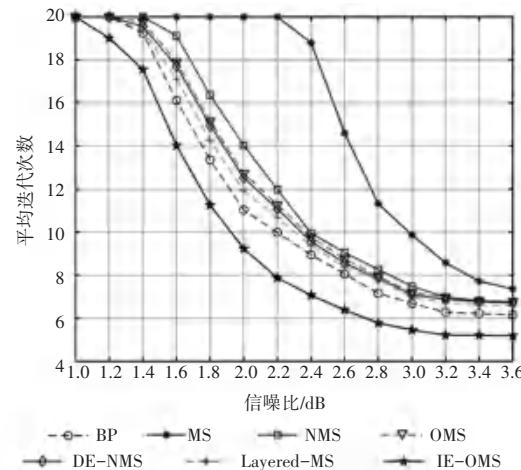


图 2 不同算法的平均迭代次数对比图

Fig. 2 Comparison of the average number of iterations of different algorithms

图 2 中,MS 算法的收敛速度最慢,所需的迭代次数更多。对 MS 算法改进后的 5 种译码算法收敛性能近似,相比之下,IE-OMS 算法表现出最为优越的收敛性,其收敛速度更快,所需的迭代次数更少。这是因为,IE-OMS 算法通过引入信息量演变理论,对节点信息的状态和分布进行更加有效且准确的处理,使得算法具有更高的收敛速度和较少的迭代次数,从而提高了译码的效率和性能。

#### 3.2 复杂度分析

NMS 算法、OMS 算法、DE-NMS 算法、Layered Min Sum 算法和 IE-OMS 算法都是对 MS 算法的改进,但在本质上仍是 MS 算法。这 7 种不同的算法在一次迭代过程中处理校验节点信息更新所需的硬件操作见表 1。在表 1 中, $N_b$  表示校验矩阵 $H$ 中元素“1”数量, $R$  表示校验矩阵 $H$ 的行重。

表 1 校验节点信息更新硬件复杂度

Table 1 Hardware complexity of verification node information update

算法	操作			
	加法	乘法	异或	比较
MS	0	0	$2N_b - R$	$2N_b$
NMS	0	$2N_b$	$2N_b - R$	$3N_b$
OMS	$2N_b$	0	$2N_b - R$	$3N_b$
DE-NMS	0	$N_b$	$2N_b - R$	$3N_b$
Layered Min Sum	$N_b$	$N_b$	$2N_b - R$	$3N_b$
IE-OMS	$N_b$	0	$2N_b - R$	$3N_b$
BP	$6N_b - R$	0	$6N_b - R$	0

表 1 中,MS 算法的计算复杂度最低,Layered Min

Sum 算法计算复杂度略高, IE-OMS 算法相对于 Layered Min Sum 算法去除了乘法运算, 计算复杂度适中, 在保持较高译码性能的同时节省了硬件资源。

硬件选用 Altera Cyclone FPGA。IE-OMS 算法和 Layered Min Sum 算法在硬件实现时, 硬件资源需求情况见表 2。

表 2 中, Layered Min Sum 算法需要更多的硬件资源, 这是因为该算法引入了层次结构, 所以需要更

多的逻辑元件和存储位来支持每一层的信息传递和更新。IE-OMS 算法需要较少的硬件资源, 这是因为该算法去除了乘法运算等复杂操作, 并简化了信息传递策略, 使得 IE-OMS 算法在保持较高译码性能的同时节省硬件资源。相对于 Layered Min Sum 算法, IE-OMS 算法可降低逻辑元件数量约 30.10% 以及存储位需求约 33.33%。

表 2 IE-OMS 算法和 Layered Min Sum 算法硬件资源需求情况

Table 2 Hardware resource requirements of IE-OMS algorithm and Layered Min Sum algorithm

硬件资源	算法	操作			总计
		加法	比较	乘法	
逻辑元件	IE-OMS	$2268N_b - 756R$	$240N_b$	0	$2508N_b - 756R$
	Layered Min Sum	$2268N_b - 756R$	$240N_b$	$1080N_b$	$3588N_b - 756R$
存储位	IE-OMS	$80N_b - 40R$	0	0	$80N_b - 40R$
	Layered Min Sum	$120N_b - 40R$	0	0	$120N_b - 40R$

## 4 结束语

针对在译码迭代过程中偏移因子  $\beta$  计算不准确的问题, 提出了一种新型的 IE-OMS 译码算法。IE-OMS 算法通过引入信息量演变理论, 先计算出每个节点信息量化点的概率集中度函数, 该函数描述了节点信息在空间中的集中程度。然后用该函数来计算节点间更新的信息均值, 通过对 2 次迭代之间的信息均值差异进行比较来调整偏移因子  $\beta$ 。在确保高译码性能的前提下, 为进一步降低硬件存储需求, 通过对偏移因子  $\beta_i$  进行加权平均化处理, 并将处理后结果作为每次迭代时所对应的唯一偏移因子  $\beta$ , 在提高译码性能的同时降低了硬件资源需求。仿真结果表明, 与传统译码算法相比, IE-OMS 算法具有较低的误码率和较少的平均迭代次数。在硬件资源上, 与 Layered Min Sum 算法相比, IE-OMS 算法可以减少约 30.10% 的逻辑元件数量和 33.33% 的内存资源。

## 参考文献

- [1] GALLAGER R G. Low-density parity-checkcodes [J]. IRE Transactions on Information Theory, 1962, 8(1): 21-28.
- [2] LIU Wenjie, HAN G J, FAN Zhengqin, et al. Threshold-voltage-drift-aware scheduling for belief propagation decoding of LDPC-coded NAND flash memory [J]. IET Communications, 2019, 13(17): 2871-2875.
- [3] LIU Fei, XU Kuiwen, ZHAO Peng, et al. Uniplanar dual-band printed compound loop antenna for WLAN/WiMAX applications

- [4] NGUYEN D T, PARK Y. Performance improvement of optical satellite communications by interleaved IEEE 802.11 LDPC [C]// Proceedings of 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN). Piscataway, NJ: IEEE, 2018: 575-579.
- [5] WANG X M, CAO W L, LI J, et al. Improved minsum algorithm based on density evolution for low-density parity check codes [J]. IET Communications, 2017, 11(10): 1582-1586.
- [6] 陈发堂, 李贺宾, 李平安. 基于偏移最小和的 LDPC 译码改进算法 [J]. 系统工程与电子技术, 2022, 44(7): 2350-2356.
- [7] WU Hao, WANG Huayong. A high throughput implementation of QC-LDPC codes for 5G NR [J]. IEEE Access, 2019, 7: 185373-185384.
- [8] 陈发堂, 王永航, 张翰卿. 多归一化因子的 LDPC 译码算法 [J]. 重庆邮电大学学报(自然科学版), 2022, 34(1): 59-64.
- [9] 陈发堂, 张友寿, 杜铮. 5G 低密度奇偶校验码的低复杂度偏移最小和算法 [J]. 计算机应用, 2020, 40(7): 2028-2032.
- [10] WU Zijing, SU Kaixiong, GUO Liting. A modified min sum decoding algorithm based on LMMSE for LDPC codes [J]. AEU-International Journal of Electronics and Communications, 2014, 68(10): 994-999.
- [11] BONCALO O, ANTAL G, AMARICAL A, et al. Layered LDPC decoders with efficient memory access scheduling and mapping and built-in support for pipeline hazards mitigation [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2019, 66(4): 1643-1656.
- [12] ZHANG P W, JIANG S, LAU F, et al. Hardware architecture of layered decoders for PLDPC-Hadamard codes [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2022, 69(12): 5325-5338.
- [13] 伊晓琦. 基于密度进化算法的正则 LDPC 码噪声门限 [J]. 河北大学学报(自然科学版), 2016, 36(3): 332-336.
- [14] 周帆, 牛琳琳, 田秉禾. 基于分层消息传递机制的低复杂度

- LDPC 译码算法 [J]. 弹箭与制导学报, 2020, 40(4): 110–112.
- [15] MARCHAND C, DORE J B, CONDE L, et al. Conflict resolution for pipelined layered LDPC decoders [C]//Proceedings of 2009 IEEE Workshop Signal Processing System. Piscataway, NJ: IEEE, 2009: 220–225.
- [16] 胡东伟. 5G LDPC 码译码器实现 [J]. 电子与信息学报, 2021, 43(4): 1112–1119.
- [17] WEE K N, CHEE Y L, KEIVAN N, et al. Application of NOMA for cellular – connected UAVS: Opportunities and challenges [J]. Science China (Information Sciences), 2021, 64(4): 1–14.
- [18] 孙大军, 欧阳雨洁, 韩云峰, 等. 基于因子图模型的水下传感器网络时间同步方法 [J]. 哈尔滨工程大学学报, 2023, 44(11): 1996–2004.
- [19] 肖若灵, 尹蕊, 倪碧雪, 等. 基于 LDPC 码校验矩阵的编码模板设计 [J]. 微波学报, 2017, 33(S1): 284–288.