

## 两种降低复杂度的符号翻转多元LDPC译码算法

陈海强 王瑶玲 韦文娟 蒋炳旭 孙友明 黎相成\* 覃团发

(广西大学计算机与电子信息学院 南宁 530004)

(广西多媒体通信与网络技术重点实验室 南宁 530004)

**摘 要:** 该文提出两种低复杂度的基于符号翻转的多元低密度奇偶校验码(LDPC)译码算法:改进型多元加权译码算法(Iwtd-AlgB)和基于截断型预测机制的符号翻转(TD-SFDP)算法。Iwtd-AlgB算法利用外信息频率和距离系数的简单求和取代了迭代过程中的乘性运算操作;TD-SFDP算法结合外信息频率和翻转函数特性,对译码节点和有限域符号进行截断与划分,使得只有满足条件的节点和符号参与运算与翻转预测。仿真和数值结果显示,该文提出的两种算法在性能损失可控的前提下,可减少每次迭代的运算操作数,实现性能和复杂度之间的折中。

**关键词:** 多元LDPC码;符号翻转;迭代译码;译码算法

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2021)01-0051-09

DOI: [10.11999/JEIT191008](https://doi.org/10.11999/JEIT191008)

## Two Low-complexity Symbol Flipping Decoding Algorithms for Non-binary LDPC Codes

CHEN Haiqiang WANG Yaoling WEI Wenjuan JIANG Bingxu

SUN Youming LI Xiangcheng QIN Tuanfa

(School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

(Guangxi Key Laboratory of Multimedia Communication and Network Technology,

Guangxi University, Nanning 530004, China)

**Abstract:** Two low-complexity symbol flipping decoding algorithms, the Improved weighted-Algorithm B algorithm (Iwtd-AlgB) and the Truncation-based Distance-Symbol-Flipping-Decoding with Prediction (TD-SFDP) algorithm, are presented for non-binary Low Density Parity Check (LDPC) codes. For the Iwtd-AlgB algorithm, the scaling factor of the flipping metric can be replaced by the simple sums of the extrinsic information and the distance-based parameter, which can avoid the multiplication operations in the iterations and thus can reduce the decoding complexity. For the presented TD-SFDP algorithm, the variable nodes and the finite field symbols are truncated and classified based on the extrinsic information frequency and the flipping function. Only those nodes/symbols that satisfy the designed conditions can be involved in the message updating process. Simulations and numeric results show that, the presented two decoding algorithms can reduce the computational complexity at each iteration with a controllable performance degradation, thus can make efficient trade-offs between performance and complexity.

**Key words:** Non-binary Low Density Parity Check (LDPC) code; Symbol flipping; Iterative decoding; Decoding algorithm

收稿日期: 2019-12-18; 改回日期: 2020-11-23; 网络出版: 2020-11-26

\*通信作者: 黎相成 xcli@gxu.edu.cn

基金项目: 国家自然科学基金(61761006, 61961004, 61662004), 广西自然科学基金(2017GXNSFAA198263, 2017GXNSFAA198276, 2018GXNSFAA138079)

Foundation Items: The National Natural Science Foundation of China (61761006, 61961004, 61662004), The Natural Science Foundation of Guangxi (2017GXNSFAA198263, 2017GXNSFAA198276, 2018GXNSFAA138079)

## 1 引言

与2元低密度奇偶校验码(Low Density Parity Check code, LDPC)相比较, 构建在 $q$ 阶有限域上的多元LDPC码<sup>[1]</sup>具有更加优秀的译码性能, 特别在码长较短、码率较大时, 其优势更加明显。然而, 多元LDPC码的性能增益往往是以高译码复杂度作为代价换取的。降低多元LDPC译码复杂度的主要思路是降低Tanner图上参与运算的节点数量以及每个节点的运算操作。经典的降低复杂度的多元LDPC译码算法包括基于概率域的和积(Q-ary Sum Product Algorithm, QSPA)算法<sup>[2,3]</sup>以及基于扩展最小和(Extended Min-Sum, EMS)的译码算法<sup>[4]</sup>及其改进版本<sup>[5-8]</sup>。此外, 基于大数逻辑(Majority-LoGic Decoding, MLGD)的简化译码算法<sup>[9-12]</sup>, 也能达到降低译码复杂度的目的。

基于符号翻转的译码算法(Symbol Flipping Decoding, SFD)是另一类重要的简化译码算法, 能在性能和复杂度之间进行有效的折中。最初的SFD算法是文献<sup>[13]</sup>提出的广义Gallager算法B(Gallager's Algorithm B, AlgB)及其改进版算法(weighted-Algorithm B, wtd-AlgB)。AlgB算法具有很低的译码复杂度, 但性能较差; wtd-AlgB算法结合了汉明距离和多数逻辑(plurality-logic)准则, 获得一定的性能增益。其他传统的SFD算法还包括文献<sup>[14]</sup>的并行符号翻转算法(Parallel Symbol Flipping Decoding algorithm, PSFD)以及基于投票机制的符号翻转算法<sup>[15,16]</sup>等。2017年, Wang等人<sup>[17]</sup>提出了基于距离和预测机制的符号翻转(Distance-Symbol Flipping Decoding with Prediction, D-SFDP)算法, 与传统的SFD算法不一样, D-SFDP算法不仅考虑了符号翻转之前的信息, 还把翻转后引起的目标函数变化考虑进来, 以此预测和翻转迭代过程中的硬判决符号。与非预测的符号翻转算法相比, D-SFDP算法能获得明显的性能提升。2018年, Mu等人<sup>[18]</sup>提出了一种基于迭代停止准则的加权符号翻转译码算法, 提高了译码算法的收敛速度。2019年, Dai等人<sup>[19]</sup>对D-SFDP算法进行了改进, 修正了算法的本地循环震荡问题, Oh等人<sup>[20]</sup>提出了一种基于判决符号可靠性的符号翻转译码算法。

虽然D-SFDP算法具有优秀的译码性能, 但仍是牺牲一定的复杂度换来的。特别地, 由于D-SFDP算法每次只能翻转1个符号, 这导致其平均迭代次数远远高于其他同类算法。因此, 降低该算法每次迭代的复杂度很有必要。本文首先提出一种基于加性参数的改进型wtd-AlgB算法, 通过调整距离参数避免了乘性操作, 从而降低复杂度; 其次, 本文

提出一种基于截断型的预测机制符号翻转译码算法(Truncation-based Distance-Symbol-Flipping-Decoding with Prediction, TD-SFDP), 结合翻转函数和变量节点参数特征对节点进行截断与划分, 使得只有满足条件的节点参与迭代运算。此外, 基于外信息频率对预测符号进行截断, 只选取最可能的有限域符号进行翻转和预测。仿真和数值结果显示, TD-SFDP算法能够在性能损失可控前提下, 减少每次迭代的运算操作数, 从而降低算法译码复杂度。

## 2 系统模型和符号定义

令 $\mathbf{H} = [h_{i,j}]_{m \times n}$ 是多元LDPC码 $m$ 行 $n$ 列的校验矩阵, 矩阵行重为 $\rho$ , 列重为 $\gamma$ 。令 $\mathbf{c} = (c_0, c_1, \dots, c_{n-1}) \in F_q^n$ 表示发送端需要传送的LDPC码字, 其中,  $q = 2^r$ 。码字 $\mathbf{c}$ 中的每一个符号 $c_j$ 所对应的二进制向量为 $\mathbf{c}_j = (c_{j,0}, c_{j,1}, \dots, c_{j,r-1})$ , 其中,  $c_{j,t} \in F_2$ ,  $0 \leq j \leq n-1$ ,  $0 \leq t \leq r-1$ 。符号 $c_j$ 中的每一个比特 $c_{j,t}$ 经调制变换后得到一个实数序列 $\mathbf{x}_j = (x_{j,0}, x_{j,1}, \dots, x_{j,r-1})$ , 其中 $x_{j,t} = \phi(c_{j,t})$ ,  $\phi(\cdot)$ 是系统的星座映射规则, 可以是简单的BPSK调制, 例如 $\phi(c_{j,t}) = 1 - 2c_{j,t}$ 。调制后的序列经加性高斯白噪声信道(AWGN)传输, 接收端的信号可表示为 $\mathbf{y}_j = (y_{j,0}, y_{j,1}, \dots, y_{j,r-1})$ , 其中,  $y_{j,t} = x_{j,t} + n_{j,t}$ ,  $n_{j,t}$ 是服从均值为0, 方差为 $\sigma^2$ 的高斯分布, 即 $n_{j,t} \sim N(0, \sigma^2)$ 。令初始信道硬判决序列为 $\mathbf{z}^{(0)} = (z_0^{(0)}, z_1^{(0)}, \dots, z_{n-1}^{(0)})$ , 其中, 当 $y_{j,t}^{(0)} \geq 0$ 时 $z_{j,t}^{(0)} = 0$ ; 当 $y_{j,t}^{(0)} < 0$ 时 $z_{j,t}^{(0)} = 1$ ,  $0 \leq j \leq n-1$ ,  $0 \leq t \leq r-1$ 。

## 3 低复杂度符号翻转多元LDPC译码

### 3.1 基于加性参数的改进型wtd-AlgB算法

文献<sup>[13]</sup>提出了一种基于符号翻转的加权多元LDPC译码算法(wtd-AlgB): 把汉明距离和多数逻辑准则进行结合, 统一形成硬判决符号的可靠度量。但该算法性能受门限值和加权系数的影响较大, 并且每次判决都需大量的实数域乘法操作。为降低算法复杂度和硬件实现能耗, 在此对原算法进行修正和改进, 描述如下:

(1) 校验节点处理: 假设算法第 $k$ 次迭代的硬判决信息为 $\mathbf{z}^{(k)} = (z_0^{(k)}, z_1^{(k)}, \dots, z_{n-1}^{(k)})$ , 定义校验和向量为

$$\mathbf{s}^{(k)} = (s_0^{(k)}, s_1^{(k)}, \dots, s_{m-1}^{(k)}) \quad (1)$$

其中,  $s_i^{(k)}$ 表示第 $i$ 个校验节点的伴随式信息

$$s_i^{(k)} = \sum_{j \in N_i} h_{i,j} z_j^{(k)} \quad (2)$$

其中, 集合 $N_i = \{j | 0 \leq j \leq n-1, h_{i,j} \neq 0\}$ 为第 $i$ 行非0列的序号。定义第 $i$ 个校验节点传递给第 $j$ 个变量节点的外信息为

$$\sigma_{i,j}^{(k)} = h_{i,j}^{-1} \sum_{j' \in N_i \setminus j} h_{i,j'} z_{j'}^{(k)} \quad (3)$$

其中,  $0 \leq i \leq m-1$ ,  $0 \leq j \leq n-1$ 。外信息 $\sigma_{i,j}^{(k)}$ 可认为是译码校验过程中, 相邻节点联合起来对符号 $z_j^{(k)}$ 的一个判决。令 $n(\sigma_{i,j}^{(k)})$ 表示外信息取值为有限域符号 $\sigma_{i,j}^{(k)}$ 的次数, 即出现频率。该频率越大, 表示 $z_j^{(k)}$ 判决为 $\sigma_{i,j}^{(k)}$ 的可能性越高(多数逻辑准则)。

(2) 变量节点处理: 对于第 $j$ 个变量节点, 在第 $k$ 次迭代中, 用 $d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ 表示硬判决符号 $z_j^{(k)}$ 与外信息 $\sigma_{i,j}^{(k)}$ 之间的汉明距离, 用 $\theta_d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ 表示一个 $r+1$ 维的实数参数向量, 由 $d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ 确定对应的参数下标。显然,  $d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ 越小, 表示当前外信息 $\sigma_{i,j}^{(k)}$ 距离 $z_j^{(k)}$ 越近, 则其可靠度越大, 对应的 $\theta_d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ 应取得较大值。因此, 距离参数的选取需遵循这样的原则:  $(\theta_0, \theta_1, \dots, \theta_r)$ 是一个非递增的实数序列, 且 $\theta_0$ 取得最大值。本文依据该原则, 以译码性能为优化目标, 利用计算机搜索得到具体的距离参数。同时, 由于译码性能主要取决于最前面的3个距离参数 $(\theta_0, \theta_1, \theta_2)$ , 当 $r \geq 3$ 时系数相差不大, 甚至可设成一致<sup>[21]</sup>。这样可加快搜索速度。

在原wtd-AlgB算法中, 判决符号的可靠度用距离参数和外信息出现频率的乘积来衡量。这样, 在每次判决时, 最多需要计算 $n\gamma$ 次浮点实数的乘法和 $n\gamma r$ 次异或(XOR)操作, 当码长 $n$ 较大时, 每次迭代会产生大量乘法操作, 从而增加了译码复杂度。实际上, 由于迭代过程中的距离参数是固定的, 并且独立于外信息出现频率, 因此它们的乘积(即判决符号可靠度量)完全可以通过简单的加性操作来完成。相应地, 只需将距离参数按比例增大即可。令 $I'(\sigma_{i,j}^{(k)})$ 表示 $\theta_d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ 与外信息出现次数 $n(\sigma_{i,j}^{(k)})$ 之和, 即

$$I'(\sigma_{i,j}^{(k)}) = \theta_d(z_j^{(k)}, \sigma_{i,j}^{(k)}) + n(\sigma_{i,j}^{(k)}) \quad (4)$$

于是, 改进算法的变量节点更新规则为

$$z_j^{(k+1)} = \begin{cases} \arg \max_{\sigma_{i,j}^{(k)}, i \in M_j} I'(\sigma_{i,j}^{(k)}), & \max_{i \in M_j} I'(\sigma_{i,j}^{(k)}) \geq T \\ z_j^{(k)}, & \max_{i \in M_j} I'(\sigma_{i,j}^{(k)}) < T \end{cases} \quad (5)$$

其中, 集合 $M_j = \{i | 0 \leq i \leq m-1, h_{i,j} \neq 0\}$ 为第 $j$ 列非0行的序号,  $T$ 为设定的门限值。对于阈值参数 $T$ , 在此分析以下两个极端的情况: (1) 令 $T=0$ , 则 $\max_{i \in M_j} I'(\sigma_{i,j}^{(k)}) < T$ 总不会成立, 而 $\max_{i \in M_j} I'(\sigma_{i,j}^{(k)}) \geq T$

总是满足, 这时每个码字符号都需要翻转。这种情况是最“冒进”的; (2) 令 $T=\infty$ , 则 $\max_{i \in M_j} I'(\sigma_{i,j}^{(k)}) \geq T$ 总不会成立, 而 $\max_{i \in M_j} I'(\sigma_{i,j}^{(k)}) < T$ 总是满足, 这时每个码字符号都不需要翻转。这种情况是最“保守”的。

显然, 这两种情况都不可取, 会造成很大的译码性能损失。因此, 参数 $T$ 需要精心选择。在实际仿真中以性能为目标, 通过计算机搜索进行选取。实际上, 由 $I'(\sigma_{i,j}^{(k)})$ 的定义可知, 其最大值为 $\theta_0 + \gamma$ , 最小值为 $\theta_r$ 。因此, 参数 $T$ 可在 $[\theta_r, \theta_0 + \gamma]$ 范围内选取, 这样可减小计算机搜索范围, 快速得到逼近最优的参数 $T$ 。

基于上述信息处理的算法简称为Iwtd-AlgB, 描述如表1所示。

本文后续的仿真实验显示, 当把距离参数调整到合适范围, 改进的Iwtd-AlgB算法在译码性能上与原算法相当, 但改进算法避免了浮点乘法运算, 降低了每次迭代的计算能耗。

表 1 Iwtd-AlgB算法

|                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 初始化: 令迭代次数 $k=0$ , 最大迭代次数为 $I_{\max}$ , 设置门限值 $T$ 和汉明距离系数 $\theta_d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ ;                                                                                                |
| 译码迭代: 当 $k < I_{\max}$ 时, 执行以下步骤                                                                                                                                                                     |
| 步骤 1: 计算硬判决序列 $\underline{z}^{(k)} = (z_0^{(k)}, z_1^{(k)}, \dots, z_{n-1}^{(k)})$ ;                                                                                                                 |
| 步骤 2: 计算伴随式 $\underline{s}^{(k)}$ , 如果 $\underline{s}^{(k)} = \underline{z}^{(k)} \mathbf{H}^T = \underline{0}$ , 所有校验方程满足, 退出迭代;                                                                    |
| 步骤 3: 对 $0 \leq j \leq n-1$ 和 $i \in M_j$ , 统计 $n(\sigma_{i,j}^{(k)})$ , 计算 $d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ , 根据汉明距离选择确定系数 $\theta_d(z_j^{(k)}, \sigma_{i,j}^{(k)})$ , 根据式(5)更新得到 $z_j^{(k+1)}$ ; |
| 步骤 4: 执行 $k \leftarrow k+1$ ;                                                                                                                                                                        |
| 输出: 迭代译码结束, 最终译码输出为 $\underline{z}^{(k)} = (z_0^{(k)}, z_1^{(k)}, \dots, z_{n-1}^{(k)})$ 。                                                                                                           |

### 3.2 基于截断型的预测机制符号翻转译码算法(TD-SFDP)

文献<sup>[17]</sup>提出一种结合了汉明距离参数和预测机制的多元LDPC符号翻转译码算法(D-SFDP)。与传统的广义Gallager算法B(AlgB)及其基于距离的改进版(wtd-AlgB)不一样, D-SFDP译码算法有两个显著特征。首先, D-SFDP算法不仅考虑了符号翻转之前的距离度量, 还把符号翻转后引起的目标函数的变化考虑进来, 使得译码器能够根据这种变化估计和预测最可能翻转的符号。其次, D-SFDP算法不使用简单的伴随式而是基于外信息 $\sigma_{i,j}^{(k)}$ 来计算距离参数。这样, 每个校验节点到变量节点的可靠

度量信息即可区分开来。同时, 由于汉明距离源于有限域符号的二进制表示, D-SFDP算法实际上也结合了有限域的结构特征。仿真显示, D-SFDP算法能够获得比传统符号翻转算法更加优秀的译码性能。

D-SFDP算法在译码性能上比传统的AlgB等算法有较大提升, 但它仍然是以牺牲一定的复杂度换来的。由于D-SFDP算法每次只能翻转1个符号, 在相同的误比特率下, 其平均迭代次数远远高于其他同类的算法。例如, 文献[17]的表3显示, 在BER =  $10^{-3}$ 时, wtd-AlgB算法的平均迭代次数仅为5.8, 但D-SFDP算法的平均迭代次数却达到了67.2, 差距非常明显。此外, 在每次迭代时, D-SFDP算法都需要计算受翻转符号影响的所有变量节点的翻转度量; 同时, 对每个节点, 还需要进一步计算最多  $q-1$  个可能的翻转符号。在码长较长、有限域较大时, 这必然极大地增加算法平均每次迭代的计算复杂度。

针对D-SFDP算法的上述特点, 本文提出一种基于截断型的预测机制符号翻转译码算法(TD-SFDP), 通过以下思路和技术方法降低每次迭代的计算复杂度: (1) 结合翻转函数和变量节点特征进行节点的截断与划分, 只有满足条件的节点参与迭代运算; (2) 基于外信息出现的频率对翻转后的  $q-1$  种有限域符号进行截断, 只选取最可能的有限域符号进行翻转度量预测。下面对TD-SFDP译码算法进行描述。

首先, 结合第  $k$  次迭代的硬判决符号  $z_j^{(k)}$  和来自信道的接收信号  $y_j$ , 计算它们之间的相关可靠度量

$$\rho(z_{j,t}^{(k)}, y_{j,t}) = \sum_{t=0}^{r-1} \phi(z_{j,t}^{(k)}) y_{j,t} \quad (6)$$

其中,  $0 \leq j \leq n-1$ 。该度量反映了硬判决符号与初始信道信息的相关度。一般而言, 其值越大, 表示翻转为该硬判决符号的可能性越大。

其次, 译码相邻节点之间的信息处理和符号翻转过程如下:

(1) 校验节点处理: 校验节点的处理操作保持不变, 即对  $0 \leq i \leq m-1$  和  $0 \leq j \leq n-1$ , 利用式(3)计算每条边上的外信息符号  $\sigma_{i,j}^{(k)}$ 。同时, 使用  $n(\sigma_{i,j}^{(k)})$  表示外信息取值为有限域符号  $\sigma_{i,j}^{(k)}$  的次数, 即外信息出现频率。

(2) 变量节点处理: 受AlgB和wtd-AlgB算法启发, 在第  $k$  次迭代时, 对于那些参数特征变化不明显的变量节点, 其符号保持不变, 即不需要执行翻转操作。这些参数特征可以是外信息出现的频率或者频率和距离的乘性/加性参数。基于此, 本文对变量节点进行如下的截断和划分, 定义节点下标集合  $\mathbf{J}^{(k)}$

$$\mathbf{J}^{(k)} = \left\{ j \mid \max_{i \in M_j} n(\sigma_{i,j}^{(k)}) \geq T_1 \right\} \quad (7)$$

其中  $0 \leq j \leq n-1$ ,  $T_1$  是一个预先设置的门限值。对于该划分的一个直观理解是, 外信息  $\sigma_{i,j}^{(k)}$  指示了当前硬判决符号的取值, 起到一个类似“裁判”的作用。对于某个变量节点  $j$ , 只有达到一定数量 ( $T_1$ ) 的裁判声明当前判决为某个符号时, 该变量节点才进入集合  $\mathbf{J}^{(k)}$ , 继续进入后续翻转预测处理; 反之, 如果当前节点的“裁判”意见不统一, 没有明显的取值倾向, 则该节点的硬判决符号不考虑翻转, 可进行“截断”处理。显然, 参数  $T_1$  的选取与具体LDPC码的列重有关, 其取值最小为0, 最大为列重  $\gamma$ 。因此, 参数  $T_1$  在此范围内选取, 即  $T_1 \in [0, \gamma]$ 。本文的仿真实验显示, 这种处理方法可方便地在性能和复杂度之间进行折中。

对于进入集合  $\mathbf{J}^{(k)}$  的变量节点, 还可以进一步通过以下策略降低复杂度。对于第  $j$  个变量节点, 定义  $z_j^{*(k)}$  为有限域上除了  $z_j^{(k)}$  以外的其他符号, 共有  $q-1$  种可能的取值。原D-SFDP算法为了估计符号翻转后的目标函数变化值, 对两个集合  $\gamma+r$  个可能的  $z_j^{*(k)}$  都进行了计算。实际上, 也可以根据外信息出现的频率, 只选取那些被“裁判”声明过 (取值与  $\sigma_{i,j}^{(k)}$  相同), 并且出现次数超过某个门限的符号进行符号翻转预测。基于此, 定义集合  $\mathbf{S}_j^{(k)}$  为

$$\mathbf{S}_j^{(k)} = \left\{ z_j^{*(k)} \mid n(z_j^{*(k)}) \geq T_2, z_j^{*(k)} \neq z_j^{(k)} \right\} \quad (8)$$

其中,  $0 \leq j \leq n-1$ ,  $T_2$  是一个预先设置的门限值, 其取值范围与参数  $T_1$  类似, 即  $T_2 \in [0, \gamma]$ 。本文利用计算机搜索, 以性能为优化目标, 确定参数  $T_1, T_2$  的具体取值。由于  $n(\sigma_{i,j}^{(k)})$  表示有限域符号的出现频率, 由此推知,  $T_1, T_2$  的取值应为  $[0, \gamma]$  范围内的某个整数, 这可进一步加快搜索速度。此外, 需要指出的是, 文献[17]也采用了一些措施来降低复杂度, 例如计算  $z_j^{*(k)}$  时, 只考虑与  $z_j^{(k)}$  距离为1且  $n(z_j^{*(k)}) \neq 0$  的有限域符号, 因为这些符号发生的概率相对更大。

本文提出的TD-SFDP算法将基于上述定义的两个截断集合进行后续的符号翻转处理, 先计算目标函数的变化量  $\Delta E_j^{(k)}(z_j^{*(k)}, z_j^{(k)})$ , 计算公式为

$$\begin{aligned} \Delta E_j^{(k)}(z_j^{*(k)}, z_j^{(k)}) = & \rho(z_{j,t}^{*(k)}, y_{j,t}) + \sum_{i \in M_j} \theta_{\bar{d}}(z_j^{*(k)}, \sigma_{i,j}^{(k)}) \\ & - \rho(z_{j,t}^{(k)}, y_{j,t}) - \sum_{i \in M_j} \theta_{\bar{d}}(z_j^{(k)}, \sigma_{i,j}^{(k)}) \end{aligned} \quad (9)$$

其中,  $z_j^{*(k)} \in S_j^{(k)}$ ,  $j \in J^{(k)}$ 。与原D-SFDP算法类似, 距离参数 $\theta_{d(z_j^{*(k)}, \sigma_{i,j}^{(k)})}$ 的选取也需遵循“非递增性”原则, 以性能为优化目标, 通过计算机搜索得到。

对于第 $j$ 个变量节点, 经 $|S_j^{(k)}|$ 个符号预测后, 从中选出一个最大值作为该节点的翻转度量变化值, 即

$$\Delta E_{\max\_j}^{(k)} = \max_{z_j^{*(k)} \in S_j^{(k)}} \Delta E_j^{(k)} \left( z_j^{*(k)}, z_j^{(k)} \right) \quad (10)$$

该数值表征了第 $k$ 次迭代中, 第 $j$ 个变量节点的硬判决符号“被翻转”的趋势:  $\Delta E_{\max\_j}^{(k)}$ 的值越大, 表示该节点的符号越趋向于执行翻转操作。本文算法每次迭代同样只翻转1个符号, 因此需要在 $|J^{(k)}|$ 个变量节点中, 寻找一个最大值 $\Delta E_{\max\_p^{(k)}}^{(k)}$ 及其对应的变量节点序号 $p^{(k)}$ , 计算如式(11)和式(12)

$$\Delta E_{\max\_p^{(k)}}^{(k)} = \max_{j \in J^{(k)}} \Delta E_{\max\_j}^{(k)} \quad (11)$$

$$p^{(k)} = \arg \max_{j \in J^{(k)}} \Delta E_{\max\_j}^{(k)} \quad (12)$$

找到需要执行翻转操作的变量节点序号 $p^{(k)}$ 以后, 即可根据式(13)对该节点的符号执行翻转操作。假设翻转后的符号为 $v_{p^{(k)}}^{(k)}$ , 则

$$v_{p^{(k)}}^{(k)} = \arg \max_{z_j^{*(k)} \in S_j^{(k)}} E_{\max\_p^{(k)}}^{(k)} \quad (13)$$

利用该翻转后的符号重新计算伴随式 $\mathbf{s}^{(k)}$ , 若 $\mathbf{s}^{(k)} = \mathbf{z}^{(k)} \mathbf{H}^T = \mathbf{0}$ , 则输出译码结果; 反之进入下一轮迭代。

基于上述截断信息处理的符号翻转译码算法简称为TD-SFDP算法, 描述如表2。

需要指出的是, 上述算法设计方案也可应用于基于多数逻辑(Plurality)的符号翻转算法<sup>[16]</sup>(P-SFDP)上, 并通过调整门限值 $T_1$ ,  $T_2$ 进行性能和复

表2 TD-SFDP算法

|                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 初始化: 令迭代次数 $k = 0$ , 最大迭代次数为 $I_{\max}$ , 设置门限值 $T_1$ 和 $T_2$ 以及汉明距离系数 $\theta_{d(z_j^{(k)}, \sigma_{i,j}^{(k)})}$ ;                                                                                                       |
| 译码迭代: 当 $k < I_{\max}$ 时, 执行以下步骤                                                                                                                                                                                           |
| 步骤 1: 计算硬判决序列 $\mathbf{z}^{(k)} = (z_0^{(k)}, z_1^{(k)}, \dots, z_{n-1}^{(k)})$ ;                                                                                                                                          |
| 步骤 2: 计算伴随式 $\mathbf{s}^{(k)}$ , 如果 $\mathbf{s}^{(k)} = \mathbf{z}^{(k)} \mathbf{H}^T = \mathbf{0}$ , 所有校验方程满足, 退出迭代;                                                                                                      |
| 步骤 3: 计算外信息频率 $n(\sigma_{i,j}^{(k)})$ , 确定截断集合 $J^{(k)}$ 和 $S_j^{(k)}$ , 计算 $\Delta E_j^{(k)}(z_j^{*(k)}, z_j^{(k)})$ , $\Delta E_{\max\_j}^{(k)}$ , $\Delta E_{\max\_p^{(k)}}^{(k)}$ , $p^{(k)}$ 以及 $v_{p^{(k)}}^{(k)}$ ; |
| 步骤 4: 执行 $k \leftarrow k + 1$ ;                                                                                                                                                                                            |
| 输出: 迭代译码结束, 最终译码输出为 $\mathbf{z}^{(k)} = (z_0^{(k)}, z_1^{(k)}, \dots, z_{n-1}^{(k)})$ 。                                                                                                                                    |

杂度之间的折中。此外, 算法在迭代过程中, 有可能会出现震荡现象, 即目标函数最大值的序号在连续迭代中保持不变, 使得算法陷入“死循环”, 导致译码性能受损, 收敛速度降低。因此, 需要对算法进行一些特别的处理, 更多的细节可参考文献<sup>[17]</sup>和文献<sup>[19]</sup>。

#### 4 复杂度分析

对Iwtd-AlgB算法, 在第 $k$ 次译码迭代中, 该算法1次迭代的计算复杂度为:  $3\delta - m$ 次有限域加法(Galois field Addition, GA),  $2\delta$ 次有限域乘法(Galois field Multiplication, GM),  $2\delta$ 次实数加法(Real Addition, RA)以及 $\delta$ 次实数比较(Real Comparison, RC)。为了便于比较, 本文也将其他几种经典译码算法的复杂度一起进行分析, 见表3。由表3数据可见, 与原算法相比, Iwtd-AlgB算法的主要差别在于计算度量 $I'(\sigma_{i,j}^{(k)})$ 时, 将 $\delta$ 次实数乘法换成了实数加法操作, 在具体硬件实现时, 可获得更低的能耗。

对TD-SFDP算法, 在第 $k$ 次迭代时, 对于 $0 \leq j \leq n-1$ , 假设外信息最大出现次数大于门限值 $T_1$ 的变量节点个数为 $n_{T_1} = |J^{(k)}|$ , 每个测试符号与外信息取得相同值的计数 $n(z_j^{*(k)})$ 大于门限值 $T_2$ 的预测值个数为 $n_{T_2} = |S_j^{(k)}|$ 。由于算法每次只翻转1个符号, 则容易推知, 该符号翻转后, 将会对 $\gamma$ 个伴随式信息和 $\gamma(\rho-1)$ 个外信息造成影响, 这些信息需要更新。另外, 由于校验矩阵无环4的设计, 需要更新的外信息都分布在不同的变量节点上。由此可知, 需要更新的变量节点数量为 $\gamma(\rho-1)$ 个; 结合翻转位, 则共有 $\gamma\rho - \gamma + 1$ 个变量节点需要更新。本文提出的TD-SFDP算法将根据截断集合 $J^{(k)}$ , 进一步只选择那些满足条件的 $n_{T_1}$ 个变量节点进行更新计算; 同时, 每个变量节点需要预测的翻转符号个数也由原算法的 $\gamma + r$ 个, 减少到 $n_{T_2}$ 个。仿真数值显示, 在第 $k$ 次迭代时, 一般都有 $n_{T_1} < \gamma\rho - \gamma + 1$ 以及 $n_{T_2} < \gamma + r$ 。因此, 算法每次迭代的复杂度得到了降低。复杂度分析的具体步骤可根据本文算法的描述并参考文献<sup>[17]</sup>得到, 如表3所示。

为了对不同译码算法的计算复杂度给出一个直观数据, 本文统计了译码算法在 $F_{16}(255, 175)$ LDPC码下不同运算操作的统计数值, 如表4所示。由表4的具体数据可以看出, 本文提出的TD-SFDP算法, 其有限域加法运算次数约为原算法的35%; 整数/实数加法运算次数约为原算法的40%。可见, 所提出算法的每次迭代的计算复杂度得到了有效的

表3 译码算法每次迭代的计算复杂度比较

| 算法        | 每次迭代的运算操作数量 |                                                                           |                     |                                                                          |                                                                |
|-----------|-------------|---------------------------------------------------------------------------|---------------------|--------------------------------------------------------------------------|----------------------------------------------------------------|
|           | IM//RM      | GA                                                                        | GM                  | IA//RA                                                                   | IC//RC                                                         |
| wBRB      | $\delta$    | $2\delta - m$                                                             | $2\delta$           | $2\delta r$                                                              | $\delta(2r + 1) - 3m$                                          |
| wtd-AlgB  | $\delta$    | $3\delta - m$                                                             | $2\delta$           | $\delta$                                                                 | $\delta$                                                       |
| Iwtd-AlgB | 0           | $3\delta - m$                                                             | $2\delta$           | $2\delta$                                                                | $\delta$                                                       |
| P-SFDP    | 0           | $2\gamma(\rho - 1)$                                                       | $\gamma(2\rho - 1)$ | $(\gamma\rho - \gamma + 1) \times (\gamma r + 3\gamma + 5r + 1)$         | $(\gamma + r - 1) \times (\gamma\rho - \gamma + 1) + 2(n - 1)$ |
| D-SFDP    | 0           | $(\gamma\rho - \gamma + 1) \times \gamma(\gamma + r) + 2\gamma(\rho - 1)$ | $\gamma(2\rho - 1)$ | $(\gamma\rho - \gamma + 1) \times (\gamma^2 + 2\gamma r + 4r + 2\gamma)$ | $(\gamma + r - 1) \times (\gamma\rho - \gamma + 1) + 2(n - 1)$ |
| TD-SFDP   | 0           | $n_{T_1} \gamma n_{T_2} + 2\gamma(\rho - 1)$                              | $\gamma(2\rho - 1)$ | $n_{T_1} n_{T_2} (r + \gamma + 1) + n_{T_1} (r + \gamma) + n\gamma$      | $n_{T_1} (n_{T_2} + \gamma - 2) + 2(n_{T_1} - 1)$              |

备注: IM: 整数乘法; RM: 实数乘法; GA: 有限域加法; GM: 有限域乘法; IA: 整数加法; RA: 实数加法; IC: 整数比较; RC: 实数比较。

表4  $F_{16}(255, 175)$ 多元LDPC码的每次迭代译码算法复杂度

| 算法        | 每次迭代的运算操作数量 |       |      |        |        |
|-----------|-------------|-------|------|--------|--------|
|           | IM//RM      | GA    | GM   | IA//RA | IC//RC |
| wBRB      | 4080        | 7905  | 8160 | 32640  | 35955  |
| wtd-AlgB  | 4080        | 11985 | 8160 | 4080   | 4080   |
| Iwtd-AlgB | 0           | 11985 | 8160 | 8160   | 4080   |
| P-SFDP    | 0           | 480   | 496  | 32053  | 5087   |
| D-SFDP    | 0           | 77600 | 496  | 104112 | 5087   |
| TD-SFDP   | 0           | 26880 | 496  | 42030  | 4288   |

降低。此外,需要注意的是,截断集合的阶数 $n_{T_1}$ 和 $n_{T_2}$ 是动态的,会随着不同的迭代次数而略有变化。本文在仿真时,取其平均值进行计算。在本例中, $n_{T_1} = 165$ , $n_{T_2} = 10$ 。

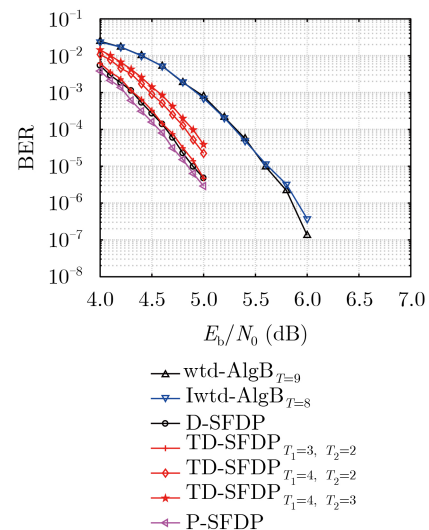
## 5 译码性能仿真实验

本节将基于两种不同构造方法的准循环多元LDPC码,对本文提出的Iwtd-AlgB算法和TD-SFDP算法进行性能仿真。仿真总帧数为 $T_{\text{total}} = 10^7$ ,最大迭代次数为 $I_{\text{max}} = 100$ ,结束条件为当错误帧数大于200帧或者总帧数超过 $T_{\text{total}}$ 。

**实验1** 考虑基于有限域构造的 $F_{16}(225, 147)$ 规则准循环LDPC码<sup>[22]</sup>,该码列重 $\rho = 14$ ,行重 $\gamma = 14$ ,码率 $R = 0.65$ 。参数设置如下:(1)对于wtd-AlgB算法,距离参数 $\theta = (2.1, 2.0, 1.0, 1.0)$ ,对于Iwtd-AlgB算法, $\theta = (4.0, 3.5, 1.0, 1.0)$ ;(2)对D-SFDP算法和TD-SFDP算法,距离参数统一选取为 $\theta = (3.0, 1.4, 1.2, 1.0)$ ;(3)对于P-SFDP算法,外信息加权系数 $\eta = (0, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.4, 1.5, 1.7, 1.8, 2.0, 2.2, 2.4, 2.4)$ 。各算法的门限参数在图上标出。

译码性能如图1所示,由图可见:(1)本文提出

的Iwtd-AlgB算法与wtd-AlgB算法性能相当,但由于避免了译码过程中的乘法操作,因此降低了每次迭代的计算复杂度;(2)D-SFDP算法和P-SFDP算法性能相当(差距在0.1 dB以内),且由于没有进行截断处理,其性能表现最佳;(3)经不同程度截断处理的TD-SFDP算法存在不同程度的性能损失。具体地,当 $T_1 = T_2 = 0$ 时对应原算法;随着 $T_1, T_2$ 的

图1  $F_{16}(225, 147)$ 多元准循环LDPC码译码性能

增大, 算法的复杂度将得到优化(下降), 但也会带来性能上的下降。例如: 在 $\text{BER}=10^{-4}$ 时, 门限值 $T_1=4$ ,  $T_2=2$ 和 $T_1=4$ ,  $T_2=3$ 的TD-SFDP算法, 其性能分别比D-SFDP算法下降了0.18 dB和0.25 dB。

算法平均迭代次数如图2所示, 由图2可见: (1)在中低信噪比区域, D-SFDP算法, P-SFDP算法和小门限值的TD-SFDP算法平均迭代次数相当, 但低于门限值较大的TD-SFDP算法。例如, 在 $\text{SNR}=4.0$  dB时,  $T_1=4$ ,  $T_2=3$ 的TD-SFDP算法的平均迭代次数约为46次; 明显大于原算法的37次; (2)在高信噪比区域, D-SFDP, P-SFDP算法和TD-SFDP算法的迭代次数基本一致; (3)在高信噪比区域, wtd-AlgB算法与本文的Iwtd-AlgB算法的平均迭代次数更小, 这是因为D-SFDP, P-SFDP算法和TD-SFDP算法每次迭代只允许翻转1个符号。

**实验2** 考虑基于有限几何构造的 $F_{16}(255, 175)$ 规则准循环LDPC码<sup>[23]</sup>, 该码行重 $\rho=16$ , 列重 $\gamma=16$ , 码率 $R=0.68$ 。参数设置如下: (1)对于wtd-AlgB算法, 距离参数 $\theta=(2.1, 2.0, 1.0, 1.0)$ , 对于Iwtd-AlgB算法,  $\theta=(4.0, 3.5, 1.0, 1.0)$ ; (2)对于D-SFDP算法和TD-SFDP算法, 距离参数统一选取为 $\theta=(3.0, 1.4, 1.2, 1.0)$ ; (3)对于P-SFDP算法,  $\eta=(0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2, 1.3, 1.4, 1.5, 2.0, 2.0)$ 为其外信息加权系数。各算法的门限参数在图上标出。

译码性能和迭代次数如图3和图4所示, 由图可观察到类似的结果, 即本文提出的两种降低复杂度的多元LDPC符号翻转译码算法与原算法相比, 在适当的参数设置下, 译码性能保持不变或者以牺牲一定性能为代价, 换取复杂度方面的降低。具体地: (1)结合了外信息和距离参数的Iwtd-AlgB符号翻转算法与wtd-AlgB算法的性能相当; (2) D-SFDP

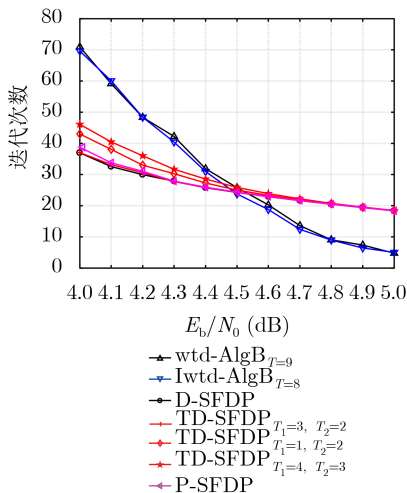


图2  $F_{16}(225, 147)$ 多元准循环LDPC码平均迭代次数

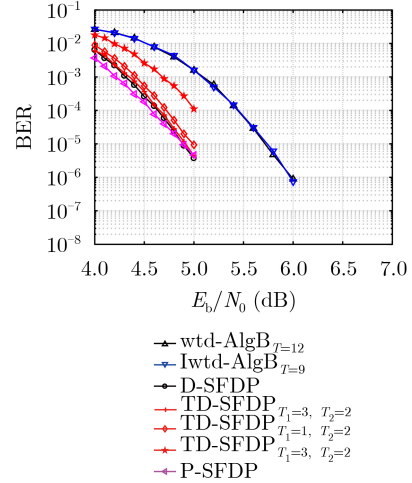


图3  $F_{16}(255, 175)$ 多元准循环LDPC码译码性能

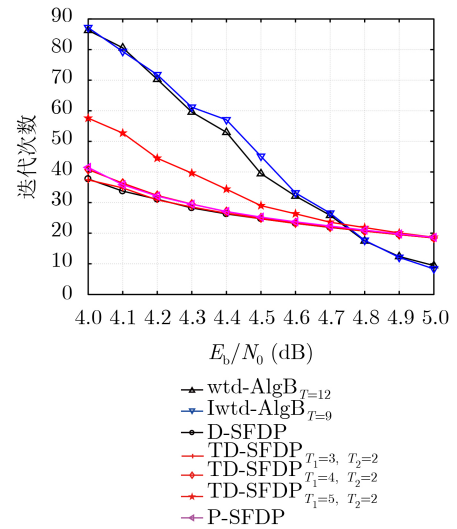


图4  $F_{16}(255, 175)$ 多元准循环LDPC码平均迭代次数

算法和P-SFDP算法性能最佳, 均优于Iwtd-AlgB算法, 在 $\text{BER}=10^{-4}$ 时获得了约0.9 dB的性能增益; (3)经不同程度截断处理的TD-SFDP算法存在不同程度的性能损失, 门限值 $T_1$ ,  $T_2$ 越大, 性能下降越多。在算法的平均迭代次数方面, 也得到与实验1类似的结果。

## 6 结论

本文提出了两种基于符号翻转的低复杂度多元LDPC译码算法, Iwtd-AlgB算法将翻转度量函数修正为外信息频率和距离参数的简单求和, 降低了复杂度; TD-SFDP算法根据参数和外信息特征对节点和有限域符号进行截断与划分, 使得只有部分比例的节点和符号参与迭代过程中的处理和翻转预测, 从而降低每次迭代的译码复杂度。仿真结果表明: 两种改进算法的译码性能与原算法相差不大, 适当调整参数, 可在性能和复杂度之间进行折中。

复杂度数据显示,第1种算法避免了原算法的实数乘法操作;第2种算法的有限域加法运算次数约为原算法的35%,整数/实数加法运算次数约为原算法的40%。此外,本文提出的截断阈值基于外信息频率设计,仍属于多数逻辑范畴,因此对于列重较大LDPC码的复杂度降低效果尤为明显;当列重减少时,截断集合的阶与原算法参与运算的节点/符号数目基本一致,其复杂度降低效果有限。

### 参 考 文 献

- [1] KOU Yu, LIN Shu, and FOSSORIER M P C. Low-density parity-check codes based on finite geometries: A rediscovery and new results[J]. *IEEE Transactions on Information Theory*, 2001, 47(7): 2711–2736. doi: [10.1109/18.959255](#).
- [2] BARNAULT L and DECLERCQ D. Fast decoding algorithm for LDPC over GF(2<sup>n</sup>)[C]. 2013 IEEE Information Theory Workshop, Paris, France, 2013: 70–73. doi: [10.1109/ITW.2003.1216697](#).
- [3] SONG Hongxin and CRUZ J R. Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording[J]. *IEEE Transactions on Magnetics*, 2003, 39(2): 1081–1087. doi: [10.1109/TMAG.2003.808600](#).
- [4] DECLERCQ D and FOSSORIER M. Decoding algorithms for non-binary LDPC codes over GF(q)[J]. *IEEE Transactions on Communications*, 2007, 55(4): 633–643. doi: [10.1109/TCOMM.2007.894088](#).
- [5] MA Xiao, ZHANG Kai, CHEN Haiqiang, *et al.* Low complexity X-EMS algorithms for nonbinary LDPC codes[J]. *IEEE Transactions on Communications*, 2012, 60(1): 9–13. doi: [10.1109/TCOMM.2011.092011.110082](#).
- [6] DEKA K, RAJESH A, and BORA P K. A novel truncation rule for the EMS decoding of non-binary LDPC codes[C]. 2018 International Conference on Signal Processing and Communications, Bangalore, India, 2018: 11–15. doi: [10.1109/SPCOM.2018.8724394](#).
- [7] LI Yibing, ZHANG Sitong, YE Fang, *et al.* Improved extended min-sum algorithm for non-binary LDPC codes based on node reliability[C]. 2019 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications, Granada, Spain, 2019: 292–297. doi: [10.1109/APWC.2019.8870565](#).
- [8] 杨威, 张为. 一种基于分层译码和Min-max的多进制LDPC码译码算法[J]. *电子与信息学报*, 2013, 35(7): 1677–1681.  
YANG Wei and ZHANG Wei. A Decoding algorithm based on layered decoding and Min-max for nonbinary LDPC codes[J]. *Journal of Electronics & Information Technology*, 2013, 35(7): 1677–1681.
- [9] SONG Liyuan, HUANG Qin, WANG Zulin, *et al.* Two enhanced reliability-based decoding algorithms for nonbinary LDPC codes[J]. *IEEE Transactions on Communications*, 2016, 64(2): 479–489. doi: [10.1109/TCOMM.2015.2512582](#).
- [10] LI Xiangcheng, QIN Tuanfa, CHEN Haiqiang, *et al.* Hard-information bit-reliability based decoding algorithm for majority-logic decodable nonbinary LDPC codes[J]. *IEEE Communications Letters*, 2016, 20(5): 866–869. doi: [10.1109/LCOMM.2016.2537812](#).
- [11] YEO S and PARK I C. Improved hard-reliability based majority-logic decoding for non-binary LDPC codes[J]. *IEEE Communications Letters*, 2017, 21(2): 230–233. doi: [10.1109/LCOMM.2016.2623783](#).
- [12] RYBIN P and FROLOV A. On the decoding radius realized by low-complexity decoded non-binary irregular LDPC codes[C]. 2018 International Symposium on Information Theory and Its Applications, Singapore, 2018: 384–388. doi: [10.23919/ISITA.2018.8664375](#).
- [13] JAGIELLO K and RYAN W E. Iterative plurality-logic and generalized algorithm B decoding of q-ary LDPC codes[C]. IEEE Information Theory and Applications Workshop, La Jolla, USA, 2011: 1–7.
- [14] HUANG Chaocheng, WU C J, CHEN Chaoyu, *et al.* Parallel symbol-flipping decoding for non-binary LDPC codes[J]. *IEEE Communications Letters*, 2013, 17(6): 1228–1231. doi: [10.1109/LCOMM.2013.051313.130303](#).
- [15] NHAN N Q, NGATCHED T M N, DOBRE O A, *et al.* Multiple-votes parallel symbol-flipping decoding algorithm for non-binary LDPC codes[J]. *IEEE Communications Letters*, 2015, 19(6): 905–908. doi: [10.1109/LCOMM.2015.2418260](#).
- [16] GARCIA-HERRERO F, DECLERCQ D, and VALLS J. Non-binary LDPC decoder based on symbol flipping with multiple votes[J]. *IEEE Communications Letters*, 2014, 18(5): 749–752. doi: [10.1109/LCOMM.2014.030914.132867](#).
- [17] WANG Shuai, HUANG Qin, and WANG Zulin. Symbol flipping decoding algorithms based on prediction for non-binary LDPC Codes[J]. *IEEE Transactions on Communications*, 2017, 65(5): 1913–1924. doi: [10.1109/TCOMM.2017.2677438](#).
- [18] MU Haiwei, MENG Jiahui, ZHANG Liang, *et al.* Weighted symbol flipping decoding for non-binary LDPC codes based on iteration stopping criterion[C]. The 37th Chinese Control Conference, Wuhan, China, 2018: 8447–8452. doi: [10.23919/ChiCC.2018.8483599](#).
- [19] DAI Bin, LIU Rongke, GAO Chenyu, *et al.* Symbol flipping algorithm with self-adjustment strategy for LDPC codes over GF(q)[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(7): 7189–7193. doi: [10.1109/TVT.2019.2915802](#).
- [20] OH J, HAN S, and HA J. An improved symbol-flipping algorithm for nonbinary LDPC codes and its application to

- NAND flash memory[J]. *IEEE Transactions on Magnetics*, 2019, 55(9): 3500113. doi: [10.1109/TMAG.2019.2918985](https://doi.org/10.1109/TMAG.2019.2918985).
- [21] HUANG Qin, ZHANG Mu, WANG Zulin, *et al.* Bit-Reliability based low-complexity decoding algorithms for non-binary LDPC codes[J]. *IEEE Transactions on Communications*, 2014, 62(12): 4230–4240. doi: [10.1109/TCOMM.2014.2370032](https://doi.org/10.1109/TCOMM.2014.2370032).
- [22] ZENG Lingqi, LAN Lan, TAI Yingyu, *et al.* Transactions Papers - Constructions of nonbinary quasi-cyclic LDPC codes: A finite field approach[J]. *IEEE Transactions on Communications*, 2008, 56(4): 545–554. doi: [10.1109/TCOMM.2008.060024](https://doi.org/10.1109/TCOMM.2008.060024).
- [23] ZENG Lingqi, LAN Lan, TAI Yingyu, *et al.* Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: A finite geometry approach[J]. *IEEE Transactions on Communications*, 2008, 56(3): 378–387. doi: [10.1109/TCOMM.2008.060025](https://doi.org/10.1109/TCOMM.2008.060025).
- 陈海强：男，1976年生，教授，研究方向为现代编码理论、协作通信。  
王瑶玲：女，1997年生，硕士生，研究方向为通信与信息系统。  
韦文娟：女，1996年生，硕士生，研究方向为通信与信息系统。  
蒋炳旭：男，1994年生，硕士生，研究方向为通信与信息系统。  
孙友明：男，1975年生，讲师，研究方向为通信与信息系统。  
黎相成：男，1979年生，讲师，研究方向为通信与信息系统。  
覃团发：男，1966年生，教授，研究方向为多媒体通信理论与技术。
- 责任编辑：余 蓉