

## 基于平均概率和停止准则的多元 LDPC 码加权符号翻转译码算法

郭锐<sup>1</sup>, 刘春于<sup>1</sup>, 王美洁<sup>1</sup>, 潘鹏<sup>1</sup>, 包建荣<sup>1,2</sup>, 郭春生<sup>1</sup>

(1. 杭州电子科技大学通信工程学院, 浙江 杭州 310018; 2. 东南大学移动通信国家重点实验室, 江苏 南京 210096)

**摘要:** 为了提高多元低密度奇偶校验(LDPC, low density parity-check)码符号翻转译码算法的性能并降低译码的复杂度, 提出了基于平均概率和停止准则的多元 LDPC 码加权符号翻转译码(APSCWSF, average probability and stopping criterion weighted symbol flipping)算法。该算法将校验节点邻接符号节点的平均概率信息作为权重, 使翻转函数更加有效, 提高符号的翻转效率, 进而改善译码性能。并且通过设置迭代停止准则进一步加快算法的收敛速度。仿真结果显示, 在加性高斯白噪声信道下, 误符号率为  $10^{-5}$  时, 相比 WSF 算法、NSCWSF 算法( $O_{sc}=10$ ) 和 NSCWSF 算法( $O_{sc}=6$ ), APSCWSF 算法( $O_{sc}=10$ ) 分别获得约 0.68 dB、0.83 dB 和 0.96 dB 的增益。同时, APSCWSF 算法( $O_{sc}=6$ ) 的平均迭代次数也分别降低 78.60%~79.32%、74.89%~75.95% 和 67.20%~70.80%。

**关键词:** 多元低密度奇偶校验码; 加权符号翻转译码; 平均概率; 停止准则; 翻转函数

中图分类号: TN911.22

文献标识码: A

## Weighted symbol-flipping decoding for non-binary LDPC codes based on average probability and stopping criterion

GUO Rui<sup>1</sup>, LIU Chun-yu<sup>1</sup>, WANG Mei-jie<sup>1</sup>, PAN Peng<sup>1</sup>, BAO Jian-rong<sup>1,2</sup>, GUO Chun-sheng<sup>1</sup>

(1. Communication Engineering Institute, Hangzhou Dianzi University, Hangzhou 310018, China;

2. National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China)

**Abstract:** In order to improve the decoding performance and reduce the decoding complexity of weighted symbols flipping decoding algorithm for nonbinary low density parity-check(LDPC) code, a weighted symbols flipping decoding for nonbinary LDPC codes based on average probability and stopping criterion was proposed. The algorithm puts the average probability of all the information nodes adjacent to the check node as weights to make the flipping-function more effectively and improve the efficiency of the flipping-symbols, and then improves the decoding performance. At the same time, the algorithm adds a stopping criterion controlling decoding iterations to further improve the decoding speed. Simulation results show that compared to WSF algorithm, NSCWSF algorithm( $O_{sc}=10$ ) and NSCWSF algorithm( $O_{sc}=6$ ), the new algorithm proposed( $O_{sc}=10$ ) is gotten about 0.68 dB, 0.83 dB and 0.96 dB gain at the symbol error rate of  $10^{-5}$  in the presence of additive white Gaussian noise(AWGN), meanwhile, the average number of decoding iterations is also reduced by 78.60%~79.32%, 74.89%~75.95% and 67.2%~70.8%, respectively.

**Key words:** non-binary low density parity-check codes, weighted symbols flipping decoding, average probability, stopping criterion, flipping-function

收稿日期: 2015-04-07; 修回日期: 2015-07-30

通信作者: 郭锐, guorui@hdu.edu.cn

基金项目: 浙江省自然科学基金资助项目(No.LY16F010013); 浙江省重点科技创新团队基金资助项目(No.2013TD03); 国家自然科学基金资助项目(No.61471152, No.61401130, No.61001133); 浙江省公益性技术应用研究计划基金资助项目(No.2015C31103); 东南大学移动通信国家重点实验室开放研究基金资助项目(No.2014D02)

**Foundation Items:** Zhejiang Provincial Natural Science Foundation of China (No.LY16F010013), Zhejiang Province Science and Technology Innovation Program Foundation (No.2013TD03), The National Natural Science Foundation of China (No.61471152, No.61401130, No.61001133), Zhejiang Province Public Welfare Technology Application Research Project Foundation (No.2015C31103), Foundation of State Key Laboratory of Mobile Communications, Southeast University (No.2014D02)

## 1 引言

相比二元 LDPC 码,多元 LDPC 码具有更强的抗突发错误能力,但译码的复杂度相对较高,因而寻找译码性能好且复杂度相对较低的译码算法显得尤为重要<sup>[1,2]</sup>。多元 LDPC 软判决译码算法主要有 GF( $q$ )上的置信传播(BP, belief propagation)译码算法<sup>[3]</sup>、基于快速傅里叶变换的置信传播(FFT-BP, fast Fourier transform belief propagation)译码算法、最大置信传播算法(MAX-LOG-BP)及扩展最小和(EMS, extended min-sum)译码算法<sup>[4,5]</sup>。这些软判决译码算法虽然具有较好的译码性能,但译码迭代过程中涉及到的实数运算量较大,不利于硬件实现。多元 LDPC 硬判决译码算法主要是符号翻转(SF, symbol flipping)译码算法,该算法实现简单,因此比较适用于一些对译码速度要求较高的场合。在 SF 算法的基础上进行改进,形成 WSF(weighted symbol flipping)算法<sup>[6-8]</sup>。该算法通过引入不同的可靠性度量值进而改善 SF 算法的译码性能。此后,又提出了基于新停止准则的 WSF(NSCWSF, new stopping criterion weighted symbol flipping)算法<sup>[9,10]</sup>和多样投票的符号翻转(MV-SF, multiple-vote symbol flipping)算法<sup>[11]</sup>。

为了提高多元 LDPC 码符号翻转译码算法的译码性能并加快译码的速度,本文提出了基于平均概率和停止准则的多元 LDPC 码加权符号翻转译码(APSCWSF, average probability and stopping criterion weighted symbol flipping)算法。该算法将与校验节点邻接符号节点的平均概率信息作为权重,得到更加有效的可靠度软信息,提高符号的翻转效率,从而改善算法的译码性能并加快算法的收敛速度。与此同时,通过增加控制迭代次数的迭代停止准则进一步加快译码的收敛速度,进而降低译码的复杂度。

## 2 多元 LDPC 码符号翻转译码算法

多元 LDPC 码与二元 LDPC 码类似,可以用稀疏检验矩阵  $H_{M \times N}$  来描述, $M$  是校验矩阵的行数,每行代表一个校验方程, $N$  是校验矩阵的列数,每列则代表一个编码码字。对于  $(N, K)$  码, $M$  必须不小于  $N-K$ 。校验矩阵中的每个元素  $H_{mn}$  都必须取值于有限域  $GF(q=2^b)$ 。校验矩阵中每行、每列中非零元素个数分别相等,称为规则码,反之称为非规则码。对于规则多元 LDPC 码,稀疏检验矩阵  $H_{M \times N}$

的行重记为  $d_c$ ,列重记为  $d_v$ 。与第  $m$  个校验方程邻接的  $d_c$  个符号节点集合记为  $N(m)$ ,与第  $n$  个符号节点相关联的  $d_v$  个校验方程记为  $M(n)$ 。对于任意一个长度为  $N$  的向量  $c$ ,如果满足  $cH^T=0$ ,则称向量  $c$  是一个合法的码字。假设  $x=[x_0x_1 \cdots x_{k-1}]$  为任一输入信息序列,其中  $x$  中的每个元素  $x_i$  都取值于有限域  $GF(q=2^b)$ , $x$  的长度为  $K$ 。经编码后的码字  $c=[c_0c_1 \cdots c_{N-1}] \in GF(q=2^b)$ 。采用 BPSK 调制方式,经过加性高斯白噪声(AWGN, additive white Gaussian noise)信道输出  $r=[r_0r_1 \cdots r_{Nb-1}]$ 。

### 2.1 符号翻转译码算法

SF 算法是一种硬判决译码算法。首先,对信道输出的二进制序列  $r$  序列进行硬判决,得到硬判决序列  $b=[b_0b_1 \cdots b_{Nb-1}]$ 。判决规则如下:若  $r_i > 0$ ,将  $b_i$  判决为 1,若  $r_i < 0$  则将  $b_i$  判决为 0 ( $0 \leq i < Nb-1$ )。因为硬判决得到的  $b$  是一个二进制序列,所以要根据二进制和  $q$  进制的转换原则,将长度为  $Nb$  的二进制序列  $b$  转换成一个长度为  $N$  的  $q$  进制序列  $y=[y_0y_1 \cdots y_{N-1}]$ ,定义映射  $h: GF(2)^b \rightarrow GF(q)$ ,  $y_n=h([b_{nb} \cdots b_{(n+1)b-1}])$  ( $0 \leq n < N$ )。这里得到的序列  $y=[y_0y_1 \cdots y_{N-1}]$  就是 SF 算法的输出序列<sup>[12,13]</sup>。

### 2.2 加权符号翻转译码算法

为了改善 SF 算法的译码性能,在 SF 算法的基础上,WSF 算法增加适量的可靠度软信息作为权重。硬判决后,WSF 算法需要计算符号的初始化似然概率,为后面译码迭代过程中计算与校验方程邻接所有符号节点的平均似然概率做准备。定义  $GF_0(q) = \{a_1 \cdots a_{q-1}\}$  为除零元素之外的有限域,则符号概率对数似然比如式(1)所示。

$$l(n=a_i) = \ln \frac{p(a=a_i)}{p(a=0)} = \frac{2}{s} \sum_{jh(a_i)_{j=1}^{d_v} r_{nb+j}} \quad (1)$$

显然,符号概率对数似然比与  $\sum_{jh(a_i)_{j=1}^{d_v} r_{nb+j}}$  成正比,这里忽略和信道噪声相关的系数  $\frac{s^2}{2}$ <sup>[14]</sup>,用  $\sum_{jh(a_i)_{j=1}^{d_v} r_{nb+j}}$  近似作为符号的符号概率对数似然比,而不会对性能产生任何影响。定义  $l_{n,a_i} = \sum_{jh(a_i)_{j=1}^{d_v} r_{nb+j}$  ( $0 \leq j < b-1$ ),然后计算每一个符号节点  $n$  分别取值为  $a_i$  ( $0 \leq i < q-1$ ) 的  $l_{n,a_i}$ ,从而得到概率矩阵  $L$ ,如式(2)所示。

$$L = [l_{n,a_i}], 0 \leq n < N, a_1 \leq a_i < a_{q-1} \quad (2)$$

WSF 算法是一种基于可靠度软信息的 SF 算法。为了确定要翻转符号的位置,首先计算每一个

符号节点  $n$  分别取每一个  $\text{GF}_0(q)$  中元素的度量值  $E_{n,a_i}$  ( $0 \leq i < q-1$ )，最后通过求和的方式得到每一个符号节点  $n$  的可靠性度量值  $E_n$  [15]，如式(3)所示。

$$E_n = \sum_{a_i \in \text{GF}_0(q)} E_{n,a_i}, 0 \leq n < N \quad (3)$$

### 3 基于平均概率的加权符号翻转译码算法

#### 3.1 APWSF 算法描述

WSF 算法只是将与校验节点邻接的符号节点(除去待翻转符号节点)的最小概率作为权重,而忽略了大量符号节点的可靠度软信息[16,17],因此 WSF 算法的性能并不理想。为了使翻转函数获得更加准确的可靠性软信息,提高符号的翻转效率,从而改善算法的译码性能,本文提出了基于平均概率的加权符号翻转译码(APWSF, average probability weighted symbol-flipping)算法。该算法将与校验方程邻接符号节点的平均概率信息作为权重,从而把校验方程邻接的所有符号节点的可靠度软信息都考虑在内,同时引入加权因子  $a$ ,将信道直接提供给待翻转符号节点自身的可靠度软信息进行加权。

APWSF 算法的具体流程如下。

**步骤 1** 初始化参数 将迭代次数  $k$  初始化为零,设置最大迭代次数  $K_{\max}$  和加权因子  $a$ 。将排除符号序列  $A$  初始化为空,比特翻转标识 flip 初始化为 1。其中排除符号序列  $A$  是用来存储那些不满足翻转条件的符号所对应的位置;比特翻转标识 flip 作为一个计数器使用,它的最大可取值为  $b$ ,即  $\text{lb}q$ 。flip 的值决定待翻转符号中需要翻转比特的个数。

**步骤 2** 计算初始硬判决输出序列,并将该序列作为  $k=0$  时译码迭代的输出,记为  $y^{(0)}$ 。

**步骤 3** 获取与校验方程邻接所有符号节点的平均概率信息。首先利用式(1)计算概率矩阵  $L=[l_{n,a_i}]$  ( $0 \leq n < N, a_1 \leq a_i < a_{q-1}$ ),进而计算平均概率矩阵  $W[w_{m,a_i}]$ ,其中  $0 \leq m < M, a_1 \leq a_i < a_{q-1}$ 。 $W$  是一个  $m$  行、 $q-1$  列的概率矩阵, $w_{m,a_i}$  表示与第  $m$  个校验方程相邻接的所有符号节点取符号  $a_i$  的平均概率信息, $w_{m,a_i}$  的计算式为

$$w_{m,a_i} = \frac{1}{d_{cm}} \sum_{i \in N(m)} |l_{i,a_i}|, 0 \leq m < M, a_1 \leq a_i < a_{q-1} \quad (4)$$

其中, $d_{cm}$  表示与第  $m$  个校验方程邻接的符号节点的个数。

**步骤 4** 计算伴随向量  $s^{(k)}, s^{(k)} = y^{(k)} H^T$ 。若伴随

向量  $s^{(k)} = \mathbf{0}$ ,则停止译码迭代,将此时的  $y^{(k)}$  和迭代次数  $k$  输出;若伴随向量  $s^{(k)} \neq \mathbf{0}$ ,且此时  $k$  等于最大迭代次数  $K_{\max}$ ,表明译码失败,停止译码迭代,输出此时的  $y^{(k)}$  和迭代次数  $k$ ;否则将迭代次数  $k$  自动加 1,进行下一次迭代。

**步骤 5** 计算第  $k$  次迭代时每一个符号节点  $n$  的可靠度量值  $E_n^{(k)}$  ( $0 \leq n < N$ )。为了计算  $E_n^{(k)}$  则首先需要计算符号节点  $n$  分别取  $\{a_1 \cdots a_{q-1}\}$  ( $0 \leq i < q-1$ ) 的可靠度量值  $E_{n,a_i}^{(k)}$

$$\begin{cases} E_{n,a_i}^{(k)} = \sum_{m \in M(n)} (2s_m^{(k)} - 1)w_{m,a_i} - a |L_{n,a_i}|, \\ 0 \quad m < M, 0 \leq i < q-1 \\ s_m^{(k)} = \begin{cases} 0, & s^{(k)}(1,m) = 0 \\ 1, & s^{(k)}(1,m) \neq 0 \end{cases} \\ E_n^{(k)} = \sum_{a \in \text{GF}_0(q)} E_{n,a_i}^{(k)}, 0 \leq n < N \end{cases} \quad (5)$$

其中,符号节点  $n$  的可靠度量值  $E_n^{(k)}$  由该符号分别取  $\{a_1 \cdots a_{q-1}\}$  ( $0 \leq i < q-1$ ) 的可靠度量值  $E_{n,a_i}^{(k)}$  求和得到。计算  $E_{n,a_i}^{(k)}$  时需要用到加权因子  $a$ ,接下来对加权因子  $a$  的取值做如下分析。

加权因子  $a$  代表对待翻转符号节点自身可靠度软信息进行加权的比重。由式(5)可知, $E_{n,a_i}^{(k)}$  由 2 项组成。因此,加权因子  $a$  可以根据 2 项所提供软信息的可靠性比率来决定,如式(6)所示。

$$a(n, a_i, m) = \frac{|L_{n,a_i}|}{w_{m,a_i}} = \frac{|L_{n,a_i}|}{\frac{1}{d_{cm}} \sum_{i \in N(m)} |l_{i,a_i}|}, \quad 0 \leq n < N, 0 \leq m < M, a_1 \leq a_i < a_{q-1} \quad (6)$$

由式(6)可知,加权因子  $a(n, a_i, m)$  随  $n$ 、 $m$ 、 $a_i$  的不同而自适应变化。它与信道直接提供给待翻转符号节点  $n$  自身的可靠度信息值  $|L_{n,a_i}|$  成正比,与第  $m$  个校验方程相邻接的符号节点的平均概率信息值  $w_{m,a_i}$  成反比。待翻转符号节点  $n$  参与的第  $m$  个校验方程中,若  $\frac{|L_{n,a_i}|}{w_{m,a_i}} > 1$ ,表明相比其他符号节点提供

给待翻转符号节点  $n$  的可靠度软信息,它自身的可靠度软信息比重较大,此时应增大对待翻转符号节点  $n$  自身可靠度软信息  $|L_{n,a_i}|$  加权的比重;反之,  $\frac{|L_{n,a_i}|}{w_{m,a_i}} < 1$  越小,应减小对待翻转符号  $n$  自身可靠度

软信息  $|L_{n,a_i}|$  加权的比重。对加权因子  $a$  进行自适应处理后, 能够提高翻转函数获得软信息的可靠度, 但同时也大大增加了  $E_{n,a_i}^{(k)}$  的计算量, 从而增加译码算法的复杂度。一般情况下,  $|L_{n,a_i}|$  的值不会偏差  $w_{m,a_i}$  很多。因此  $a(n, a_i, m)$  的取值一般在 1 附近。在算法性能不会受到较大影响的情况下, 为了降低算法的复杂度, 本文的仿真中将  $a$  取近似值 1。

步骤 6 选出待翻转符号的位置  $n^{(k)}$

$$n^{(k)} = \arg \max E_n^{(k)}, n \in [1, N], n \notin A \quad (7)$$

步骤 7 确定待翻转符号需要翻转的幅度大小。符号位置  $n^{(k)}$  中具体要翻转的比特位置是由经过 AWGN 信道传输后输出的二进制序列  $r=[r_0r_1 \cdots r_{Nb-1}]$  所决定。待翻转符号位置  $n^{(k)}$  处的符号可以转换成  $b$  个比特, 这  $b$  个比特与  $r$  中的  $[r_{nb}r_{nb+1} \cdots r_{(n+1)b-1}]$  相对应。将  $|r_i|(nb-i \cdots (n+1)b-1)$  进行从大到小排序,  $|r_i|$  越小, 则对应比特的可靠性越低。因此根据比特翻转标识 flip 选取  $[r_{nb} \cdots r_{(n+1)b-1}]$  中绝对值最小的 flip 个比特位置, 并翻转对应位置的 flip 个比特, 从而得到新符号序列  $y^{(k)}=[y_0y_1 \cdots y_{N-1}]$ 。

步骤 8 检测是否存在死循环。死循环的检测方法如下, 首先定义输出符号序列矩阵  $Y_{(k+1) \times N}$ , 如式(8)所示。

$$Y = \begin{bmatrix} y^{(0)} \\ y^{(1)} \\ \vdots \\ y^{(k)} \end{bmatrix} \quad (8)$$

接着定义一个死循环检测矩阵  $E_{k \times N}$ , 如式(9)所示。

$$E = \begin{bmatrix} y^{(k)} - y^{(0)} \\ y^{(k)} - y^{(1)} \\ \vdots \\ y^{(k)} - y^{(k-1)} \end{bmatrix} \quad (9)$$

只要死循环检测矩阵  $E$  中存在一行元素全部为零, 称检测到死循环。否则没有检测到死循环。

1) 如果检测到死循环且 flip 并未达到其最大值  $b$ , 将 flip 加 1, 然后返回到步骤 7 重新确定翻转符号要翻转的具体比特位置, 并翻转对应位置的 flip 个比特。

2) 如果检测到死循环但 flip 已达到其最大值  $b$ , 将当前所选择的翻转符号位置  $n^{(k)}$  存储到排除符

号序列  $A$  中, 并将比特翻转标识 flip 置 1。然后返回到步骤 6 重新寻找翻转符号的位置  $n^{(k)}$ 。

3) 如果没有检测到死循环 则将排除符号序列  $A$  设置为空, 比特翻转标识 flip 置 1, 返回步骤 4。

### 3.2 对 APWSF 算法的改进

APWSF 算法与 WSF 算法相比, 在译码性能和译码速度方面都得到了很大程度的提高。为了进一步加快译码的收敛速度, 对本文提出的 APWSF 算法做进一步的改进, 称之为 APSCWSF 算法。该算法是在 APWSF 算法的基础上, 通过增加控制迭代次数的迭代停止准则得到。

为了阐述 APSCWSF 算法, 首先定义“可正确译码的码字”和“不可正确译码的码字”。所谓“可正确译码的码字”是指那些随着迭代次数  $k$  的增加, 不满足校验方程的个数迅速减少, 即伴随向量  $s^{(k)}$  中的非零元素个数迅速减少的那些码字; 而“不可正确译码的码字”是指那些随着迭代次数  $k$  的增加, 不满足校验方程的个数没有呈现明显减少趋势, 而出现了反复振荡的那些码字。

迭代停止准则具体描述如下。

首先, 定义检测反复振荡期的下限为  $th1$ , 上限为  $th2$ ; 定义振荡计数器 Count 并赋初值 0, Count 的最大值为  $Osc$ ; 定义检测窗口向量  $Win(k)$ , 窗口的长度为  $l_w$ 。其中,  $Osc$  和  $l_w$  值可以先选取, 再通过数据仿真检验调整<sup>[10]</sup>。计算第  $k+1$  次迭代后不满足校验方程的个数  $u(k+1)$ , 并与第  $k$  次迭代后不满足校验方程的个数  $u(k)$  作差得到差值  $dv(k)$ , 如式(10)所示。

$$dv(k) = u(k+1) - u(k) \quad (10)$$

然后令检测窗口向量  $Win(k)=[dv(k) \ dv(k+1) \cdots \ dv(k+l_w-1)]$ , 当迭代次数  $k$  达到该窗口长度  $l_w$  的整数倍时, 开始对检测窗口向量  $Win(k)$  的每一个元素  $dv(k+i)$  ( $0 \leq i \leq l_w-1$ ) 进行检测。如果对于检测窗口内的每一个元素  $dv(k+i)$  ( $0 \leq i \leq l_w-1$ ) 都满足  $th1 \leq dv(k+i) \leq th2$ , 则说明不满足校验方程的个数处于振荡期, 将振荡计数器 Count 加 1。当振荡计数器 Count 达到它的最大值  $Osc$  时, 就认为该码字是不可正确译码的, 提前停止译码迭代并输出此时的  $y^{(k)}$  和迭代次数  $k$ 。

### 3.3 复杂度理论分析

本节对复杂度进行定性分析, 在 4.2 节中将对复杂度进行具体的定量分析, 并详细比较各个算法所需的加法运算平均总次数。表 1 分别给出了 WSF

译码算法、NSCWSF 译码算法、MV-SF 译码算法以及本文提出的 APWSF 译码算法和 APSCWSF 译码算法所需的运算复杂度,其中  $A_i$  指各个算法所需的迭代次数,  $M$  表示校验矩阵的行数,  $N$  表示校验矩阵的列数,  $d_c$  代表校验矩阵的行重,  $d_v$  代表校验矩阵的列重。其中对于 MV-SF 译码算法,  $L$  表示测试向量的长度,  $h$  作为协调译码复杂度和译码性能参数。

由表 1 可得如下结论。

1) APWSF 算法的复杂度低于 WSF 算法。WSF 算法和 APWSF 算法每次迭代的计算量相同,差别仅在于迭代次数。但 APWSF 算法的迭代次数要远小于 WSF 译码算法。

2) NSCWSF 算法的复杂度低于 WSF 算法。虽然 NSCWSF 算法每次迭代所需的加法运算总次数(多出迭代停止准则的部分— $2lw$ )稍大于 WSF 算法,但是 NSCWSF 译码算法的迭代次数远小于 WSF 算法。MV-SF 算法不仅需要大量加法运算,而且需要乘法运算。

3) APSCWSF 算法的复杂度低于 APWSF 算法。APSCWSF 算法每次迭代所需的加法运算总次数(多出迭代停止准则的部分— $2lw$ )稍大于 APWSF 算法,但 APSCWSF 算法的迭代次数远少于 APWSF 算法。

表 1 5 种不同译码算法实数运算的复杂度

译码算法	加法运算次数	乘法运算次数
WSF	$A_1 [(q-1)d_c d_v + N - 1 + q - 2 + b1bb]$	0
NSCWSF	$A_2 [(q-1)d_c d_v + N - 1 + q - 2 + b1bb + 2lw]$	0
MV-SF	$A_3 [2((d_c - 1)M + L) + 2d_c ML + Nq + M \sum_{i=1}^d i]$	$2(d_c M + h)$
APWSF	$A_4 [(q-1)d_c d_v + N - 1 + q - 2 + b1bb]$	0
APSCWSF	$A_5 [(q-1)d_c d_v + N - 1 + q - 2 + b1bb + 2lw]$	0

## 4 性能仿真与分析

### 4.1 仿真参数设置

仿真所采用的码字具体参数如下。

1)  $GF(q=2^4)$  的 QC LDPC 码<sup>[18]</sup>(999,888)(简称码 A),码 A 的码率为  $\frac{8}{9}$ ,行重为 27,列重为 3。在码 A 的系统仿真中,将译码的最大迭代次数  $K_{max}$  设置为 100 次;对于 NSCWSF 算法和 APSCWSF 算法,检测窗口的长度  $l_w$  设置为 10。2)  $GF(q=2^6)$  的 QC LDPC

码(999,888)(简称码 B),码 B 的码率为  $\frac{8}{9}$ ,行重为 27,

列重为 3。对码 B 的系统仿真中,将译码的最大迭代次数  $K_{max}$  设置为 100 次;对于 NSCWSF 算法和 APSCWSF 算法,检测窗口的长度  $l_w$  设置为 10。3)  $GF(q=2^6)$  的 LDPC 码(142,71)(简称码 C),码 C 的码率为  $\frac{1}{2}$ ,行重为 4,列重为 2。对码 C 的系统仿真中,将译码的最大迭代次数  $K_{max}$  设置为 50 次,对于 NSCWSF 算法和 APSCWSF 算法,检测窗口的长度  $l_w$  设置为 4。4)  $GF(q=2^5)$  的 LDPC 码(834,723)(简称码 D),码 D 的码率约为 0.867,行重为 27,列重为 2。对码 D 的系统仿真中,将译码的最大迭代次数  $K_{max}$  设置为 10 次,对于 NSCWSF 算法和 APSCWSF 算法,检测窗口的长度  $l_w$  设置为 2。5)  $GF(q=2^5)$  的 LDPC 码(1455,1344)(简称码 E),码 E 的码率约为 0.923,行重为 27,列重为 3。在码 E 的系统仿真中,将译码的最大迭代次数  $K_{max}$  设置为 10 次,对于 NSCWSF 算法和 APSCWSF 算法,检测窗口的长度  $l_w$  设置为 2。

### 4.2 译码性能仿真与结果分析比较

本节主要对文献[7]中提出的 WSF 算法、文献[10]中提出的 NSCWSF 算法( $Osc=3$ 、 $Osc=6$ 、 $Osc=10$ )、文献[11]中提出的 MV-SF 算法,以及本文提出的 APWSF 算法和 APSCWSF 算法( $Osc=3$ 、 $Osc=6$ 、 $Osc=10$ )进行系统仿真,并对各个译码算法的误符号率(SER, symbol-error rate)、误帧率(FER, frame-error rate)和平均迭代次数这 3 个参数指标进行比较。其中,码 A、码 B、码 C 主要对 WSF 算法、NSCWSF 算法、APWSF 算法、APSCWSF 算法进行系统仿真,误符号率曲线、误帧率曲线、平均迭代次数曲线分别如图 1~图 9 所示。利用码 D、

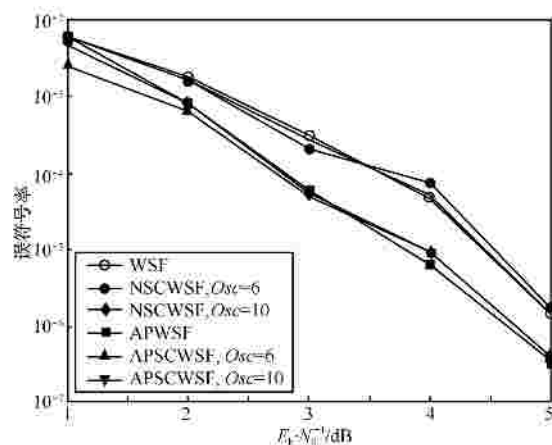


图 1  $GF(q=2^4)$  下的 QC LDPC 码(999, 888)不同译码算法下的误符号率性能比较(码 A)

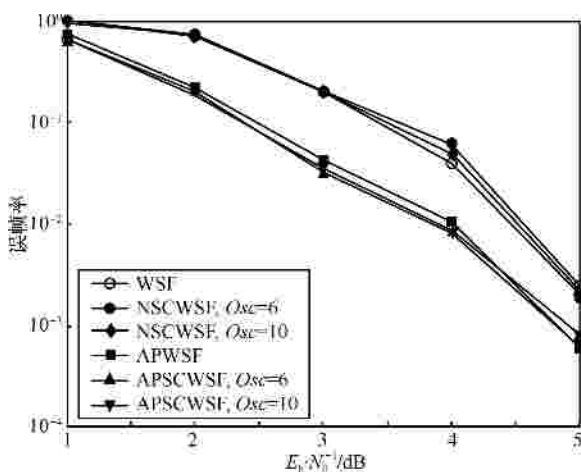


图 2 GF( $q=2^4$ )下的 QC LDPC 码(999,888)不同译码算法下的误帧率性能比较 (码 A)

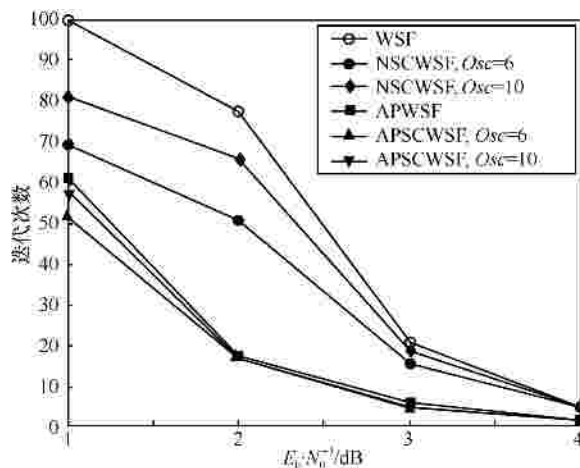


图 3 GF( $q=2^4$ )下的 QC LDPC 码(999,888)不同译码算法下的平均迭代次数比较 (码 A)

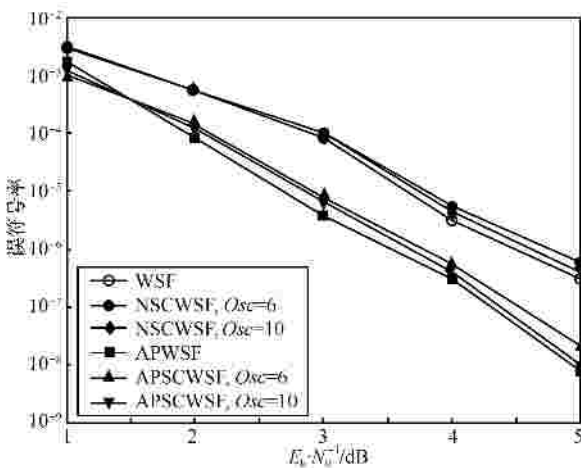


图 4 GF( $q=2^6$ )下的 QC LDPC 码(999,888)不同译码算法下的误符号率性能比较 (码 B)

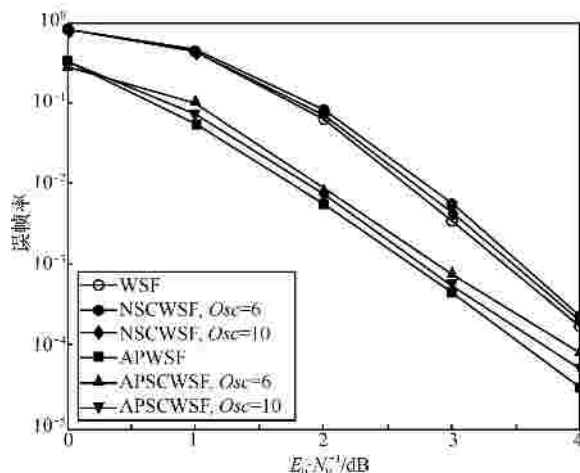


图 5 GF( $q=2^6$ )下的 QC LDPC 码(999,888)不同译码算法下的误帧率性能比较 (码 B)

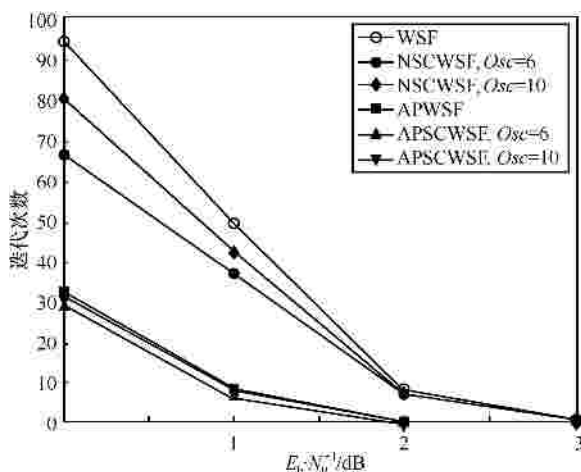


图 6 GF( $q=2^6$ )下的 QC LDPC 码(999,888)不同译码算法下的平均迭代次数比较 (码 B)

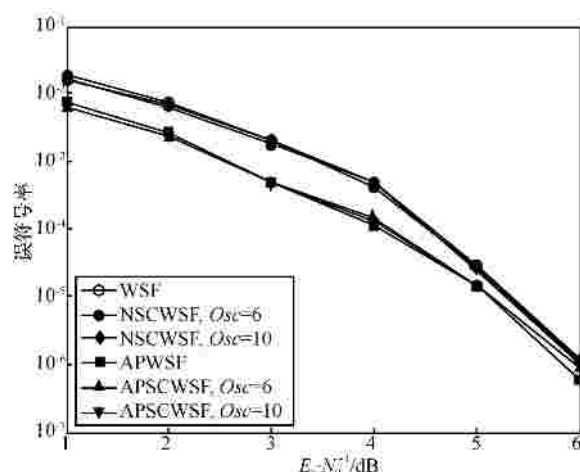


图 7 GF( $q=2^6$ )下的 LDPC 码(142,71)不同译码算法下的误符号率性能比较(码 C)

码 E 对 MV-SF 算法、WSF 算法、NSCWSF 算法、APWSF 算法、APSCWSF 算法的误帧率进行了仿真，分别如图 10、图 11 所示。

由仿真图可知，本文提出的新算法(APWSF 算法和 APSCWSF 算法( $Osc=6,10$ ))性能明显优于已有的 WSF 算法、NSCWSF( $Osc=6,10$ )算法和 MV-SF

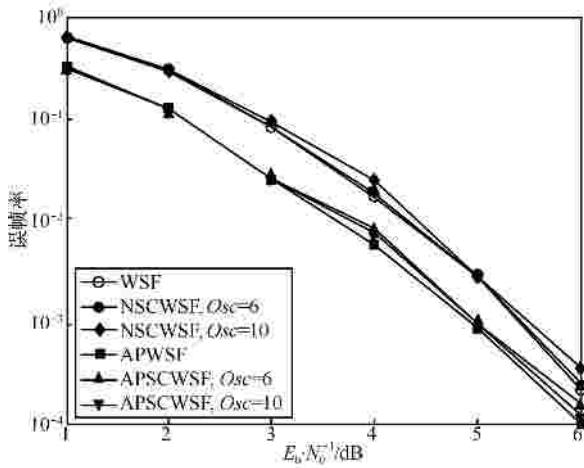


图 8 GF( $q=2^6$ )下的 LDPC 码(142,71) 不同译码算法下的误帧率性能比较(码 C)

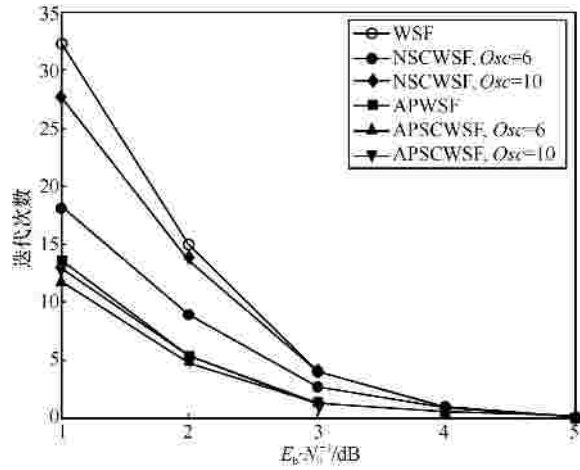


图 9 GF( $q=2^6$ )下的 LDPC 码(142,71) 不同译码算法下的平均迭代次数比较(码 C)

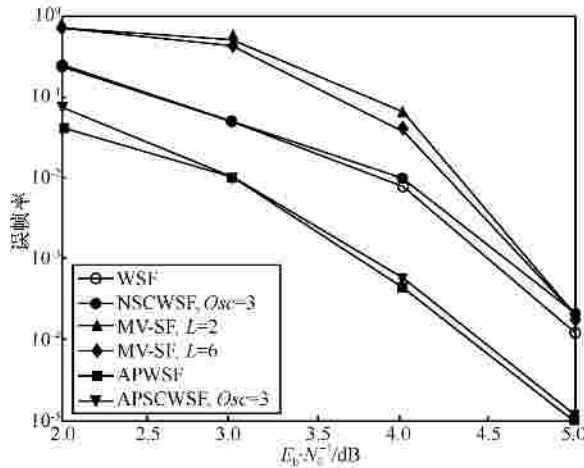


图 10 GF( $q=2^5$ )下的 LDPC 码(834, 723) 不同译码算法下的误帧率性能比较(码 D)

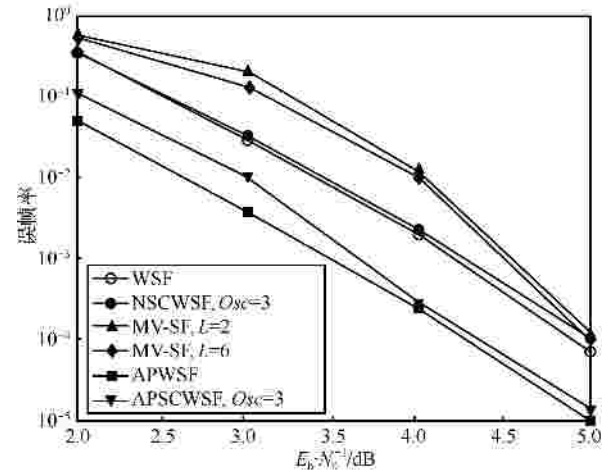


图 11 GF( $q=2^5$ )下的 LDPC 码(1455, 1344) 不同译码算法下的误帧率性能比较(码 E)

算法。随着振荡计数器最大值  $Osc$  的增大，带有停止准则的 NSCWSF 算法和 APSCWSF 算法的性能将分别越接近于 WSF 算法和 APWSF 算法。在信噪比较低时，NSCWSF 算法和 APSCWSF 算法的性能分别略优于 WSF 算法和 APWSF 算法。因为在信噪比较低时，存在较多的“不可正确译码码字”，设置停止准则提前终止迭代反而会减少译码中错误的码字个数。由图 3、图 6、图 9 可知，在相同信噪比情况下，本文提出的新算法(APWSF 算法和 APSCWSF 算法( $Osc=6,10$ ))所需的平均迭代次数远少于已有的 WSF 算法和 NSCWSF( $Osc=6,10$ )算法，其中 APSCWSF 算法( $Osc=6$ )在所有的算法中平均迭代次数最少。

表 2 给出了码 A、码 B、码 C 的误符号率为  $10^{-5}$

时，本文提出的新算法相对于其他已有算法获得的增益。由表 2 可知，对于码 A、码 B、码 C，在误符号率为  $10^{-5}$  时，APWSF 算法的性能是最好的，APSCWSF 算法( $Osc=10$ )次之，NSCWSF 算法( $Osc=6$ )最差。与 WSF 算法相比，本文提出的 APSCWSF 算法( $Osc=10$ )分别获得约 0.72 dB、0.68 dB 和 0.42 dB 的增益；与 NSCWSF 算法( $Osc=10$ )相比，分别获得约 0.75 dB、0.83 dB 和 0.44 dB 的增益；与 NSCWSF 算法( $Osc=6$ )相比，分别获得约 0.83 dB、0.96 dB 和 0.50 dB 的增益。

表 3 给出了码 D、码 E 的误帧率为  $10^{-3}$  时，本文提出的新算法相对于其他已有算法获得的增益。由表 3 可知，对于码 D，相比 MV-SF( $L=2, L=6$ )算法，本文提出的 APSCWSF 算法( $Osc=3$ )分别获得约

1.00 dB 和 0.95 dB 的增益；相比 NSCWSF 算法 ( $O_{sc}=3$ )，获得约 0.83 dB 的增益；相比 WSF 算法 ( $O_{sc}=3$ )，获得约 0.75dB 的增益；对于码 E，相比 MV-SF( $L=2, L=6$ )算法，本文提出的 APSCWSF 算法 ( $O_{sc}=3$ )分别获得约 1.05 dB 和 1.00 dB 的增益；相比 NSCWSF 算法( $O_{sc}=3$ )，获得约 0.80 dB 的增益；相比 WSF 算法( $O_{sc}=3$ )，获得约 0.70 dB 的增益。

表 4、表 5 分别给出了在码 A 情况下，各种译码算法的平均迭代次数和加法运算总次数。由表 4 统计可得，码 A 在 2 dB 和 3 dB 时，相对于 WSF 算法、NSCWSF 算法( $O_{sc}=10$ )和 NSCWSF 算法

( $O_{sc}=6$ )，APSCWSF 算法( $O_{sc}=6$ )平均迭代次数分别降低 60.66% ~79.32%、49.26%~75.95%和 33.81%~70.80%。由表 5 统计可得，码 A 在 2 dB 和 3 dB 时，相比于其他算法，APSCWSF 算法的译码复杂度最低，APSCWSF 算法( $O_{sc}=6$ )的平均加法运算总次数降低 67.20%~79.32%。因此，与已有的 WSF 算法和 NSCWSF 算法相比，APSCWSF 算法不但提高了译码性能，而且极大地减少了译码的迭代次数。

改善性能的主要因素为，构造翻转函数时，将校验方程邻接的符号节点的平均概率信息作为权重，获得了更加可靠的翻转信息，从而译码迭代过

表 2  $SER=10^{-5}$  时，相比已有的算法，本文提出的新算法获得的增益统计

译码算法	码 A		码 B		码 C	
	$\frac{E_b}{N_0}$ /dB	增益/dB	$\frac{E_b}{N_0}$ /dB	增益/dB	$\frac{E_b}{N_0}$ /dB	增益/dB
APWSF	3.52	—	1.70	—	5.25	—
APSCWSF( $O_{sc}=10$ )	3.70	0.18	1.87	0.17	5.28	0.03
APSCWSF( $O_{sc}=6$ )	3.78	0.26	1.96	0.26	5.23	0.07
WSF	4.42	0.90	2.55	0.85	5.70	0.45
NSCWSF( $O_{sc}=10$ )	4.45	0.93	2.70	1.00	5.72	0.47
NSCWSF( $O_{sc}=6$ )	4.53	1.01	2.83	1.13	5.78	0.53

表 3  $FER=10^{-3}$  时，相比已有的算法，本文提出的新算法获得的增益统计

译码算法	码 D		码 E	
	$\frac{E_b}{N_0}$ /dB	增益/dB	$\frac{E_b}{N_0}$ /dB	增益/dB
APWSF	3.75	—	3.50	—
APSCWSF( $O_{sc}=3$ )	3.80	0.05	3.63	0.13
MV-SF( $L=2$ )	4.75	1.00	4.55	1.05
MV-SF( $L=6$ )	4.70	0.95	4.50	1.00
WSF	4.50	0.75	4.20	0.70
NSCWSF( $O_{sc}=3$ )	4.63	0.83	4.30	0.80

表 4 码 A 在 2 dB 和 3 dB 时，APSCWSF 算法( $O_{sc}=6, O_{sc}=10$ )和其他算法的迭代次数比较

译码算法	$\frac{E_b}{N_0}=2$ dB			$\frac{E_b}{N_0}=3$ dB		
	平均迭代次数	相对降低次数	下降比例/%	平均迭代次数	相对降低次数	下降比例/%
APSCWSF( $O_{sc}=6$ )	16.52	—	—	4.41	—	—
APSCWSF( $O_{sc}=10$ )	16.86	0.34	2.02	4.94	0.53	10.72
APWSF	17.35	0.83	4.78	5.22	0.81	15.51
NSCWSF( $O_{sc}=6$ )	50.33	33.81	67.20	15.11	10.70	70.80
NSCWSF( $O_{sc}=10$ )	65.78	49.26	74.89	18.34	13.93	75.95
WSF	77.18	60.66	78.60	21.33	16.92	79.32

表 5 码 A 在 2 dB 和 3 dB 时，各种译码算法所需的译码复杂度

译码算法	$\frac{E_b}{N_0} = 2$ dB		$\frac{E_b}{N_0} = 3$ dB	
	加法运算的平均总次数	下降比例/%	加法运算的平均总次数	下降比例/%
APSCWSF( $O_{sc}=6$ )	37 253	—	9 945	—
APSCWSF( $O_{sc}=10$ )	38 019	2.01	11 140	10.73
APWSF	38 777	3.93	11 667	14.76
NSCWSF( $O_{sc}=6$ )	113 495	67.17	34 073	70.81
NSCWSF( $O_{sc}=10$ )	148 334	74.69	41 357	75.95
WSF	159 087	76.58	47 673	79.14

程中得到的待翻转符号更加精确。WSF 算法和 NSCWSF 算法中，只是将校验方程邻接的符号节点的最小概率信息作为权重，MV-SF 算法只是考虑了部分符号节点的概率信息，都忽略了大量信息节点的可靠度信息。迭代次数降低的主要因素有两方面，一方面仍然由于将校验方程邻接的符号节点的平均概率信息作为权重，翻转函数获得了更加可靠的信息，从而加速了译码迭代的收敛；另一方面，增加控制译码迭代次数的停止准则，从而大大减少了“不可正确译码码字”的翻转次数。

## 5 结束语

本文提出了基于平均概率和停止准则的多元 LDPC 码加权符号翻转译码(APSCWSF)算法。该算法将校验节点邻接的符号节点的平均概率信息作为权重，得到更加有效的翻转函数，提高了符号的翻转效率，从而改善算法的译码性能，并通过增加迭代停止准则来进一步加快译码的收敛速度。仿真结果表明，与已有的 WSF 算法、NSCWSF 算法和 MV-SF 算法相比，本文提出的 APSCWSF 算法可分别获得约 0.75 dB、0.83 dB 和 1.00 dB 的增益；相比 WSF 算法、NSCWSF 算法，APSCWSF 算法的加法运算的平均总次数分别降低 76.58%~79.14% 和 67.17%~70.81%。由此可见，本文提出的 APSCWSF 算法获得了性能和复杂度之间更好的匹配，可以更好地满足高速通信系统对译码算法的需求。

## 参考文献：

- [1] PENG R, CHEN R. Design of nonbinary LDPC codes over  $GF(q)$  for multiple-antenna transmission[C]//IEEE Military Communications Conference. Washington, DC, USA, c2006:1-7.
- [2] YIN Y, LIU X, CHULONG L. Design of nonbinary LDPC codes based on cycle entropy[C]//International Conference on Electronics, Communications and Control. Zhejiang, China, c2011: 2014-2016.
- [3] TENG M, LIU X, DENG Z. Improved BP decoding algorithm nonbinary LDPC codes based on bit-flipping method[C]//International Conference on Computational Intelligence and Security. Suzhou, China, c2008:376-380.
- [4] GUO X, FEI Y. Adaptive extended min-sum algorithm for nonbinary LDPC decoding[C]//IEEE Global Telecommunications Conference Houston, TX, USA, c2011:1-6.
- [5] ZHAO S, LU Z, MA X, et al. A variant of the EMS decoding algorithm for nonbinary LDPC codes[J]. IEEE Communications Letters, 2013, 17(8): 1640-1643.
- [6] LIU B, GAO J, DOU G. Majority decision based weighted symbol-flipping decoding for nonbinary LDPC codes[C]//The 2010 2nd International Conference on Future Computer and Communication. Wuhan, China, c2010: 223-226.
- [7] LIU B, DOU G, TAO W, et al. Weighted symbol-flipping decoding for nonbinary LDPC codes[C]//The 2010 2nd International Conference on Networks Security, Wireless Communications and Trusted Computing. Wuhan, China, c2010: 223-226.
- [8] LIU B, DOU G, TAO W, et al. Efficient stopping criterion for hard weighted symbol-flipping decoding of nonbinary LDPC codes[J]. IEEE Communications Letters, 2011, 15(3): 337-339.
- [9] CHEN X, MEN A, ZHOU W. A stopping criterion for nonbinary LDPC codes over  $GF(q)$ [C]//The 2008 11th IEEE Singapore International Conference on Communication Systems. Guangzhou, China, c2008: 1312-1315.
- [10] 刘冰, 陶伟, 窦高奇, 等. 基于新停止准则的多进制 LDPC 码加权符号翻转译码算法[J]. 电子与信息学报, 2011, 33(2): 309-314. LIU B, TAO W, DOU G Q, et al. Weighted symbol-flipping decoding for nonbinary LDPC codes based on a new stopping criterion[J]. Journal of Electronics & Information Technology, 2011, 33(2): 309-314.
- [11] GARCI-HERRERO F, LI E, DECLERCQ D, et al. Multiple-vote symbol-flipping decoder for nonbinary LDPC codes[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2011, 22(11): 2256-2267.

- [12] ZHAO S, MA X, ZHANG X, et al. A class of nonbinary LDPC codes with fast encoding and decoding algorithms[J]. IEEE Transactions on Communications, 2013, 61(1): 1-6.
- [13] CHAO C H, CHI J W, CHAO Y C, et al. Parallel symbol-flipping decoding for non-binary LDPC codes[J]. IEEE Communications Letters, 2013, 17(6): 1228-1231.
- [14] GUO F, HANZO L. Reliability ratio based weighted bit-flipping decoding for low-density parity-check codes[J]. Electronics Letters, 2004, 40(21): 1356-1358.
- [15] GARCIA-HERRERO F, DECLERCQ D, VALLS J. Non-binary LDPC decoder based on symbol-flipping with multiple votes[J]. IEEE Communications Letters, 2014, 18(5): 749-752.
- [16] 张高远, 周亮, 文红. 基于幅度和的 LDPC 码加权比特翻转译码算法[J]. 系统工程与电子技术, 2014, (4): 752-757.  
ZHANG G Y, ZHOU L, WEN H. Sum of the magnitude based weighted bit-flipping decoding algorithms for LDPC codes[J]. Systems Engineering and Electronics, 2014, (4): 752-757.
- [17] 张高远, 周亮, 苏伟伟, 等. 基于平均幅度的 LDPC 码加权比特翻转译码算法[J]. 电子与信息学报, 2013, (11): 2572-2578.  
ZHANG G Y, ZHOU L, SU W W, et al. Average magnitude based weighted bit-flipping decoding algorithm for LDPC codes[J]. Journal of Electronics & Information Technology, 2013, (11): 2572-2578.
- [18] MANKAR M V, PATIL A, ASUTKAR G M. Single mode quasi-cyclic LDPC decoder using modified belief propagation[C]//International Conference on Communications and Signal Processing, Melmaruvathur, India, c2014: 862-866.

## 作者简介:



郭锐 (1980-), 男, 湖北十堰人, 博士, 杭州电子科技大学副教授、硕士生导师, 主要研究方向为无线通信、信道编码等。

刘春于 (1992-), 女, 河南洛阳人, 杭州电子科技大学硕士生, 主要研究方向为无线通信、信道编码等。

王美洁 (1991-), 女, 山东烟台人, 杭州电子科技大学硕士生, 主要研究方向为无线通信、信道编码等。

潘鹏 (1983-), 男, 浙江丽水人, 博士, 杭州电子科技大学副教授、硕士生导师, 主要研究方向为无线通信、多天线技术等。

包建荣 (1978-), 男, 浙江杭州人, 博士, 杭州电子科技大学副教授、硕士生导师, 主要研究方向为无线通信、信道编码等。

郭春生 (1971-), 男, 甘肃武威人, 博士, 杭州电子科技大学副教授、硕士生导师, 主要研究方向为视频分析与模式识别。