

## 一种基于分层译码和 Min-max 的多进制 LDPC 码译码算法

杨威\* 张为

(天津大学电子信息工程学院 天津 300072)

**摘要:** 该文在现有译码算法的基础上提出一种高效的非二进制低密度奇偶校验码(NB-LDPC)译码方法, 充分利用了分层译码算法与 Min-max 算法的优点, 不但译码复杂度低、需要的存储空间小, 而且可将译码速度提高一倍。应用该算法, 对一种定义在  $GF(2^5)$  上的(620,509)码进行了仿真。该码的仿真结果表明: 在相同误码率下, 该文译码算法所需最大迭代次数仅为 Zhang 的算法(2011)的 45%。

**关键词:** 非二进制低密度奇偶校验码(NB-LDPC); 分层译码; Min-max; 准循环码

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2013)07-1677-05

DOI: 10.3724/SP.J.1146.2012.01634

## A Decoding Algorithm Based on Layered Decoding and Min-max for Nonbinary LDPC Codes

Yang Wei Zhang Wei

(School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China)

**Abstract:** Based on decoding algorithms available nowadays, a high efficiency decoding algorithm for nonbinary low-density parity-check codes is proposed. The algorithm takes advantage of both the layered decoding algorithm and the Min-max algorithm, which not only achieves the low complexity and the low memory requirement, but also speeds up the decoding by two times. Simulations for a (620,509) code over  $GF(2^5)$  show that with the same bit error rate, the maximum iteration times of the proposed algorithm required is just 45% of Zhang's algorithm presented in 2011.

**Key words:** NonBinary Low-Density Parity-Check (NB-LDPC) codes; Layered decoding; Min-max; Quasi-cyclic codes

### 1 引言

LDPC(Low-Density Parity-Check) 码由 Gallager 于 20 世纪 60 年代首次提出<sup>[1]</sup>。构造在有限域上的二进制 LDPC 码在码长非常长时可以达到非常接近香农限的译码性能, 当码长较短时, 二进制 LDPC 码的性能有所下降。多进制 LDPC 码(NB-LDPC)的提出<sup>[2]</sup>有效地解决了二进制 LDPC 码在码长较短时译码性能下降的问题。研究表明, 定义在  $GF(q)(q>2)$  上的 LDPC 码对于短码或中等长度的码有更好的纠错性能<sup>[2]</sup>。

NB-LDPC 码的译码可以采用 BP(Belief Propagation)算法<sup>[2]</sup>。NB-LDPC 码应用 BP 算法译码的复杂度很高。为降低复杂度, 可以在频率域<sup>[3]</sup>或者对数域<sup>[4]</sup>实现 BP 算法, 然而在频率域译码时需要大量的乘法运算, 并且需要额外的计算来实现 Fourier 正/反变换; 在对数域, 乘法被转换为加法,

但是在对数域计算 Fourier 变换非常困难。为利用以上两种译码算法的优势, 一种在混合域进行译码的算法在文献[5]中被提出, 然而两种域的转换需要查找表, 这将需要大量的存储空间。

对数域 EMS(Extended Min-Sum)<sup>[6]</sup>和 Min-max<sup>[7]</sup>是近似的 BP 算法, 这两种算法在对校验节点的处理过程中只需要加法运算或者比较大小运算, 因此译码复杂度可以大大降低。在 Min-max 算法中, 用比较运算代替了 EMS 算法的加法运算, 所以 Min-max 算法的计算复杂度比 EMS 算法更低, 但性能略差于 EMS 算法<sup>[8]</sup>。为降低存储需求, 在文献[6]中提出在校验节点的处理过程中只保留  $n_m < q$  个可靠度信息, 虽然这样会使译码性能降低, 但可很大程度上减少需要存储的信息量, 更利于硬件实现。

对校验节点的处理是 NB-LDPC 译码过程中复杂度最高的部分, 在文献[2]中提出的 forward-backward 算法简化了这一过程, 使译码得到了简化因而更易于实现。然而在 forward 和 backward 处理过程中产生的信息必须被存储起来以供后续的计算

使用, 这需要占用大量的存储空间。在文献[8]中提出了一个新颖的 forward-backward 校验节点处理方案, 这种算法只选取了一部分最可靠的信息来参加 forward-backward 处理过程。此外, 文献[8]中改变了 C-to-V 信息的生成方法, 这在很大程度上减少了信息的存储需求并且降低了计算复杂度。

NB-LDPC 码也可以应用分层译码方法<sup>[9]</sup>, 分层译码不仅改善了译码性能, 而且使译码速度得到了很大的提高。在文献[10]中提出了一种 NB-LDPC 码的分层译码方法, 但需要存储的信息量大, 并且复杂度较高。文献[11]同样应用了分层译码方案来对 NB-LDPC 码进行译码, 但同样存在复杂度高、存储量大的问题。

本文在文献[8]的基础上将分层译码算法引入, 将文献[8]的低复杂度、低存储需求和分层译码的高收敛速度结合起来。仿真结果表明, 在同样的误码率的情况下, 本文提出的方法只需要文献[8]中少于一半的最大迭代次数, 因此可以达到更高的吞吐量, 是一种更加快速、低复杂度和低存储需求的译码方法。

## 2 Min-max 译码算法<sup>[7]</sup>

构造在  $GF(q)$  上的 NB-LDPC 码是一种线性分组码, 在它的奇偶校验矩阵中  $\mathbf{H}_{M \times N}$  只有少量的非零元素, 如果我们用  $\alpha$  代表  $GF(q)$  的本原元, 那么  $\alpha$  的幂生成了  $GF(q)$  中的所有非零元素。一个 LDPC 码的  $\mathbf{H}$  矩阵可以用 Tanner 图来表示, Tanner 图中的校验节点和变量节点分别对应着  $\mathbf{H}$  的行和列, 如果  $\mathbf{H}$  的第  $i$  行  $j$  列的元素是非零元素, 那么在第  $i$  个校验节点和第  $j$  个变量节点间存在一条连线, 我们把这条连线称为 Tanner 图的一条边, 可以通过在 Tanner 图的每一个边上传递可靠度信息来对 LDPC 码进行迭代译码。对于接收到的一个码元, 在发送端传出的码元可能是  $GF(q)$  中  $q$  个元素的任意一个, 因此在 Tanner 图的每一条边上都要传递  $q$  个可靠度信息, 用其来表示各个元素的可能性大小。在对数域 Min-max 算法中, 码元的可靠度信息可用对数似然比(LLR)  $L(\alpha) = \log(P(\beta)/P(\alpha))$  来表示, 其中  $\beta$  是可能性最大的元素,  $\alpha$  可以是  $GF(q)$  中任意一个元素, 所以在 Tanner 图的每条边上传递的是  $q$  个 LLR, 我们把这  $q$  个 LLR 用一个包含  $q$  个元素的矢量来表示。

用  $\mathbf{v}_{m,n}$  代表由校验节点  $m$  传递到变量节点  $n$  的 LLR 矢量, 用  $\mathbf{u}_{m,n}$  代表由变量节点  $n$  传递到校验节点  $m$  的 LLR 矢量, 用  $S_c(n)$  代表连接到变量节点  $n$  的校验节点的集合, 用  $S_v(m)$  代表连接到校验节点  $m$

的变量节点的集合。  $L(m | \alpha_n = \alpha)$  是满足限制条件  $\sum_{j \in S_v(m) \setminus n} h_{m,j} \alpha_j = h_{m,n} \alpha$  的域元  $(\alpha_j)_{(j \in S_v(m) \setminus n)}$  的序列集合, 这里的  $h_{i,j}$  是  $\mathbf{H}$  中第  $i$  行  $j$  列的元素。假定由信道获得的变量节点  $n$  的 LLR 矢量用  $\gamma_n$  表示, 在乘以  $h_{i,j}$  或者除以  $h_{i,j}$  由独立的运算部分完成之后<sup>[3]</sup>, Min-max 译码算法可以用算法 A 来描述。当译码成功或者达到最大迭代次数时, 算法 A 终止。

### 算法 A: Min-max 译码算法

初始化:

$$\mathbf{u}_{m,n}(\alpha) = \gamma_n(\alpha) \quad (1)$$

迭代:

处理校验节点

$$\mathbf{v}_{m,n}(\alpha) = \min_{(\alpha_j) \in L(m | \alpha_n = \alpha)} \left\{ \max_{j \in S_v(m) \setminus n} \mathbf{u}_{m,j}(\alpha_j) \right\} \quad (2)$$

处理变量节点

$$\left. \begin{aligned} \mathbf{u}'_{m,n}(\alpha) &= \gamma_n(\alpha) + \sum_{i \in S_c(n) \setminus m} \mathbf{v}_{i,n}(\alpha) \\ \mathbf{u}_{m,n}(\alpha) &= \mathbf{u}'_{m,n}(\alpha) - \min_{w \in GF(q)} \left( \mathbf{u}'_{m,n}(w) \right) \end{aligned} \right\} \quad (3)$$

后验信息计算

$$\tilde{\gamma}_n(\alpha) = \gamma_n(\alpha) + \sum_{i \in S_c(n)} \mathbf{v}_{i,n}(\alpha) \quad (4)$$

在处理校验节点时, 直接计算  $L(m | \alpha_n = \alpha)$  的复杂度非常高, forward-backward 算法的提出避免了直接计算  $L(m | \alpha_n = \alpha)$ , 使得译码过程更加简便易行, 用  $n_i$  表示连接到校验节点的第  $i$  个变量节点, 用  $d_c$  表示校验节点的度数, forward-backward 算法可以描述为以下几个步骤<sup>[12]</sup>:

前向计算

$$\left. \begin{aligned} f_1(\alpha) &= \mathbf{u}_{m,n_1}(\alpha) \\ f_i(\alpha) &= \min_{\alpha' + \alpha'' = \alpha} \left( \max \left( f_{i-1}(\alpha'), \mathbf{u}_{m,n_i}(\alpha'') \right) \right) \end{aligned} \right\} \quad (5)$$

后向计算

$$\left. \begin{aligned} b_{d_c}(\alpha) &= \mathbf{u}_{m,n_{d_c}}(\alpha) \\ b_i(\alpha) &= \min_{\alpha' + \alpha'' = \alpha} \left( \max \left( b_{i+1}(\alpha'), \mathbf{u}_{m,n_i}(\alpha'') \right) \right) \end{aligned} \right\} \quad (6)$$

校验节点信息计算

$$\left. \begin{aligned} \mathbf{v}_{m,n_1}(\alpha) &= b_2(\alpha) \\ \mathbf{v}_{m,n_{d_c}}(\alpha) &= f_{d_c-1}(\alpha) \\ \mathbf{v}_{m,n_i}(\alpha) &= \min_{\alpha' + \alpha'' = \alpha} \left( \max \left( f_{i-1}(\alpha'), b_{i+1}(\alpha'') \right) \right) \end{aligned} \right\} \quad (7)$$

在文献[8]中提出的算法进一步降低了 forward-backward 的计算量, 依照文献[8]中的计算方法, 选取少量的可靠度信息得到的译码性能与应

用全部的可信度信息得到的译码性能几乎一样。在文献[8]中还校验节点信息的生成方法进行了改进,文献[8]中的改进算法不仅大大降低了存储需求,而且进一步减少了计算量。

### 3 分层译码及其与 Min-max 算法的结合

分层译码方案<sup>[13]</sup>已经被广泛地应用于二进制 LDPC 码的译码<sup>[14,15]</sup>, NB-LDPC 的译码同样可以应用分层译码方案, 分层译码不仅可以减少存储需求、提高译码性能, 而且可以极大地提高收敛速度, 实现快速译码。在这种译码方案中,  $\mathbf{H}$  矩阵以行为基准被分成若干块, 每一块被称为一层, 例如对于矩阵  $\mathbf{H}_{60 \times 255}$  共有 60 行, 那么我们可以把第 1 到 15 行作为第 1 块, 也就是第 1 层, 第 16 到 30 行分为第 2 层, 依此类推, 此矩阵可被分为 4 层。从上一层得到的 C-to-V 信息马上被应用到下一层的 C-to-V 信息的计算中, 在每一次 C-to-V 信息更新后计算码元的后验概率, 这就导致更加频繁地更新后验概率, 由此得到了更高的收敛速度。

本文提出将分层译码和文献[8]中 Min-max 译码算法结合, 结合后的译码过程可以用算法 B 来描述。对于行重为定值  $d_c$  的 NB-LDPC 码, 在 Tanner 图上, 每一个校验节点将会有个  $d_c$  变量节点与之相连接, 每一个变量节点传递给校验节点的信息都是一个  $q$  维矢量, 在本文中我们将每一个变量节点称为一阶, 那么每一阶都由  $q$  个信息点组成, 因此总共有  $d_c \times q$  个信息, 我们将这  $d_c \times q$  个信息的前  $R$  个最小值找出。由算法 A 中式(3)的运算可以得出每一阶都会有一个 LLR 值为 0 的信息点, 那么在找出这  $R$  个值后, 在某一阶上可能只有一个 LLR 值为 0 的信息点, 在此我们将这样的阶称为 zero stage, 其它的不是只有 LLR 值为 0 的信息点的阶称为 nonzero stage。关于 zero stage 以及 nonzero stage 的划分详见参考文献[8]。

为更好地描述此算法, 我们假定将  $\mathbf{H}$  矩阵分为  $L$  层, 每一层有  $P$  行, 第  $k$  次迭代第  $l$  层校验节点到变量节点的信息用  $\mathbf{v}_{m,n}^{(k,l)}$  来表示, 用  $\mathbf{u}_{m,n}^{(k,l)}$  代表第  $k$  次迭代第  $l$  层变量节点到校验节点的信息, 用  $\tilde{\gamma}_n^{(k,l)}$  代表第  $k$  次迭代第  $l$  层后所得到的后验信息, 用  $\mathbf{R}_{n_i}$  表示连接到校验节点  $m$  的第  $i$  个变量节点在经过筛选后剩余的元素集合, 对第 0 层, 第 1 层, ..., 第  $L-1$  层的变量节点和校验节点都顺序处理一遍后才构成了一次完整的迭代过程。

#### 算法 B: 本文提出的译码算法步骤

步骤 1 初始化: 计算  $\gamma_n(\alpha)$ ,  $\mathbf{u}_{m,n}(\alpha) = \gamma_n(\alpha)$ 。对于所有  $m$  以及所有的  $n \in S_v(m)$ , 令  $\mathbf{v}_{m,n}(\alpha) = 0$ 。

步骤 2 预处理: 对于每个  $m$ , 找出  $\mathbf{u}_{m,n}(\alpha)$  的前  $R$  个最小值, 找出  $T$  个 nonzero stage<sup>[8]</sup>, 标记每一阶的对数似然比值为 0 处的  $\alpha$  值。

步骤 3 开始迭代:

(1) 对于第  $k$  次迭代第  $l$  层, 计算  $\mathbf{v}_{m,n}^{(k,l)}(\alpha)$ , 其中 forward-backward 计算方法如式(8)和式(9)

$$\left. \begin{aligned} f_1(\alpha) &= \mathbf{u}_{m,n_1}^{(k,l)}(\alpha) \\ f_i(\alpha) &= \min_{\{\alpha' + \alpha'' = \alpha, \alpha'' \in R_{n_i}\}} \left( \max \left( f_{i-1}(\alpha'), \mathbf{u}_{m,n_i}^{(k,l)}(\alpha'') \right) \right) \end{aligned} \right\} (8)$$

$$\left. \begin{aligned} b_{d_c}(\alpha) &= \mathbf{u}_{m,n_{d_c}}^{(k,l)}(\alpha) \\ b_i(\alpha) &= \min_{\{\alpha' + \alpha'' = \alpha, \alpha'' \in R_{n_i}\}} \left( \max \left( b_{i+1}(\alpha'), \mathbf{u}_{m,n_i}^{(k,l)}(\alpha'') \right) \right) \end{aligned} \right\} (9)$$

由此可得  $\mathbf{v}_{m,n}(\alpha)$  为

$$\left. \begin{aligned} \mathbf{v}_{m,n_1}^{(k,l)}(\alpha) &= b_2(\alpha) \\ \mathbf{v}_{m,n_{d_c}}^{(k,l)}(\alpha) &= f_{d_c-1}(\alpha) \end{aligned} \right\} (10)$$

对于  $0 < i < d_c$ , 如果第  $i$  阶是 zero stage, 那么

$$\mathbf{v}_{m,n_i}^{(k,l)}(\alpha) = f_{d_c}(\alpha + \alpha_{i0}) \quad (11)$$

其中  $\alpha_{i0}$  代表的是 zero stage 的  $\alpha$  值, 如果第  $i$  阶是 nonzero stage, 那么

$$\mathbf{v}_{m,n_i}^{(k,l)}(\alpha) = \min_{\alpha' + \alpha'' = \alpha} \left( \max \left( f_{i-1}(\alpha'), b_{i+1}(\alpha'') \right) \right) \quad (12)$$

(2) 对于第  $k$  次迭代第  $l$  层计算后验信息:

$$\begin{aligned} \tilde{\gamma}_n^{(k,l)}(\alpha) &= \tilde{\gamma}_n^{(k,l-1)}(\alpha) + \mathbf{v}_{m,n}^{(k)}(\alpha) \\ &\quad - \mathbf{v}_{m,n}^{(k-1)}(\alpha), \quad (l-1) \times P \leq m \leq l \times P \end{aligned} \quad (13)$$

其中  $P$  为第  $l$  层的行数。

(3) 如若译码成功或者达到最大迭代次数则终止迭代, 否则计算变量节点的信息, 并对  $\mathbf{u}_{m,n}^{(k,l+1)}(\alpha)$  继续进行预处理, 处理方法同步骤 2。

$$\left. \begin{aligned} \mathbf{u}_{m,n}^{(k,l+1)}(\alpha) &= \tilde{\gamma}_n^{(k,l)}(\alpha) - \mathbf{v}_{m,n}^{(k)}(\alpha) \\ \mathbf{u}_{m,n}^{(k,l+1)}(\alpha) &= \mathbf{u}_{m,n}^{(k,l+1)}(\alpha) - \min_{w \in \text{GF}(q)} \left( \mathbf{u}_{m,n}^{(k,l+1)}(w) \right) \end{aligned} \right\} (14)$$

## 4 仿真结果及分析

近年来具有准循环特性的一种特殊 LDPC 码, 即准循环低密度奇偶校验码(QC-LDPC 码)得到了越来越多的重视。QC-LDPC 码的生成矩阵和校验矩阵均具有分块循环移位特性, 大大降低了编译码复杂度, 同时仍保持了 LDPC 码的优良纠错性能。本文主要关注的是准循环非二进制低密度奇偶校验码(QC-NB-LDPC 码), 这一类码不仅具有非常好的纠错性能, 而且使得更加高效的并行译码成为可能, 一些 QC-NB-LDPC 的构造方法可以在文献[16]中

找到。对于一个 QC-NB-LDPC 码, 与之相关联的  $\mathbf{H}$  矩阵可以被分割成  $r \times t$  个子矩阵, 每一个子矩阵或者是一个 0 矩阵, 或者是一个移位的单位矩阵, 只是这个单位矩阵的非零元素被  $\text{GF}(q)$  中的某一个元素代替。

依照文献[16]中的方法, 我们构造了两种 QC-NB-LDPC 码。在伽罗华域  $\text{GF}(2^5)$  上选取  $C=1$ ,  $N=31$ <sup>[10]</sup> 生成矩阵  $\mathbf{W}^{(1)}$ , 对  $\mathbf{W}^{(1)}$  进行扩散得到矩阵  $\mathbf{H}^{(1)}$ , 然后令  $\square=4$ ,  $\rho=20$  得到  $\mathbf{H}^{(1)}$  的一个子矩阵, 这个矩阵经过高斯消除后确定秩为 111, 所以我们构造的码 1 是行重 20、列重 4 的(620,509)码。对于同一个矩阵  $\mathbf{H}^{(1)}$ , 令  $\square=3$ ,  $\rho=12$  得到  $\mathbf{H}^{(1)}$  的另一个子矩阵, 这个矩阵经过高斯消除后确定秩为 86, 所以我们构造的码 2 是行重 12、列重 3 的(372,286)码。

在高斯信道下, 采用 BPSK 调制, 对这两种码分别应用文献[8]中的 Min-max 算法和本文提出的分层 Min-max 算法进行了软件仿真, 仿真结果如图 1 以及图 2 所示。

在图 1 和图 2 中 Min-max Original 代表的是文献[8]中算法的仿真结果, Min-max Layered 代表的是本文提出的算法的仿真结果。在对码 1 仿真时, 我们选取  $R=38$ ,  $T=4$ ,  $L=4$ , 对码 2 仿真时, 选取  $R=30$ ,  $T=3$ ,  $L=3$ 。从仿真结果可以看出, 用文献[8]中提出的算法得到图 1 所示的译码性能需要进行 22 次迭代, 但应用本文提出的算法得到同样的译码性能只需要进行 10 次迭代, 最大迭代次数减少了一半以上。对码 2 的仿真可以得到与码 1 相同的结论。从图 1 可以看出两种译码方法在 SNR 为 3.8 dB 时, BER 相差甚微, 应用 VS2010 软件平台, 我们对这种情况下的译码所需时间分别进行了测量, 得到的数据如表 1 所示。从表 1 可以看出, 在同样的仿真环境下, 为得到同样的译码性能, 应用本文提出的算法耗费的时间仅为文献[8]的 50%。对码 2 在 SNR

表 1 (620,509)码 SNR 为 3.8 dB 时译码时间比较

码字个数	译码时间(s)	
	Min-max Layered	Min-max Original
10	16	28
100	137	280
1000	1424	2846

表 2 (372,286)码 SNR 为 3.1 dB 时译码时间比较

码字个数	译码时间(s)	
	Min-max Layered	Min-max Original
10	12	26
100	1098	219
1000	1083	2164

为 3.1 dB 时的仿真测试结果如表 2 所示, 测试结果同样表明译码所需时间减少了一半。

为进一步验证本文提出算法的译码性能, 我们应用文献[11]中提出的分层译码算法对构造的码 2 进行了仿真分析, 并将得到的仿真结果与本文提出的算法进行了对比, 得到的译码性能曲线如图 3 所示。在图 3 中 Layered 代表的是文献[11]中算法的仿真结果。从图 3 可以看出, 在相同的最大迭代次数时本文提出算法的性能与文献[11]相差很小, 这是因为本文提出的算法只应用了  $R$  个变量节点到校验节点的信息进行译码, 而文献[11]应用了全部的  $d_c \times q$  个变量节点到校验节点的信息进行译码, 并且本文提出算法在计算校验节点到变量节点信息时只选取了  $T$  个 nonzero stage, 这样就造成了部分信息的丢失。在调整  $R$  和  $T$  的值后, 本文提出算法与文献[11]的性能对比如图 4 所示, 从图 4 可以看出, 调整  $R$  和  $T$  的取值后可以得到与文献[11]完全一样的译码性能。

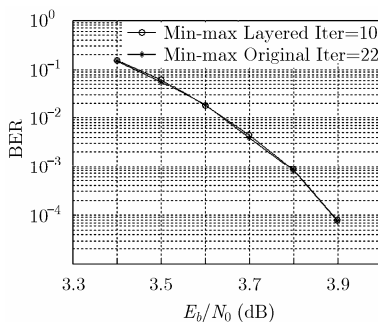


图 1 (620,509)NB-LDPC 码的误码率

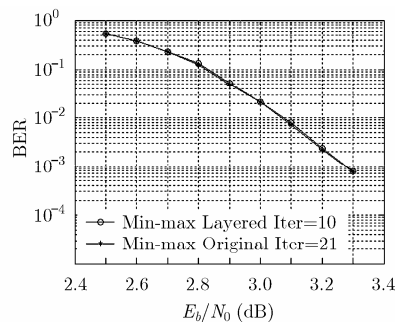


图 2 (372,286)NB-LDPC 码的误码率

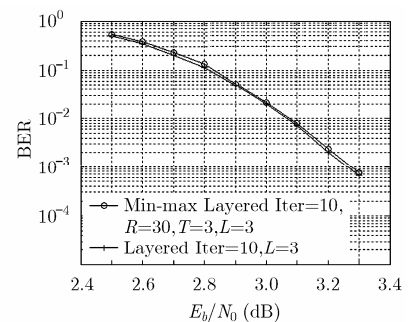


图 3 (372,286)NB-LDPC 码应用本文提出的算法和文献[11]中算法的误码率

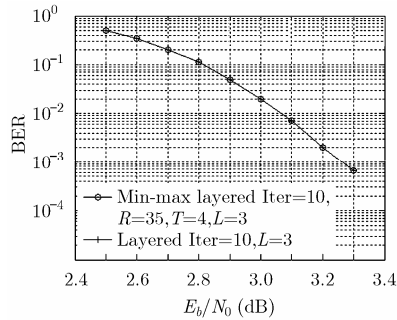


图4 修改参数后(372,286)NB-LDPC码应用本文提出的算法和文献[11]中算法的误码率

对文献[11]和本文提出算法的计算复杂度以及存储量的分析可以得出，对于每一个校验节点，文献[11]在计算时  $\mathbf{v}_{m,n}$  需要进行  $3 \times (d_c - 2) \times q^2$  次 Min-max 运算，需要存储的中间结果占用了  $2w \times q \times (d_c - 2)$  byte 的存储空间，在这里  $w$  代表存储一个 LLR 值需要的比特数。相比之下，本文提出算法在计算  $\mathbf{v}_{m,n}$  时只需要  $q \times (R - 2 \times R_1 - R_{d_c}) + T \times q^2$  次 Min-max 运算，需要存储的中间结果减少为  $(2T + 1) \times w \times q$ ，计算复杂度和存储需求大大降低，在这里的  $R_1$  代表的是从  $\mathbf{u}_{m,n}$  选取的前  $R$  个最小值之中有  $R_1$  个值是属于第 1 阶的， $R_{d_c}$  代表的是从  $\mathbf{u}_{m,n}$  选取的前  $R$  个最小值之中有个值是属于第  $d_c$  阶的。

## 5 结论

本文将分层译码和 Min-max 译码算法结合起来，提出了一种更加高效的译码算法，不仅复杂度低、占用的存储空间小，而且译码速度得到了极大的提高。仿真结果表明，应用本文提出的算法对 QC-NB-LDPC 码进行译码，所需最大迭代次数减少了一半以上，译码时间可以缩短 50%，更加适合高速数据传输应用需求。

### 参考文献

- [1] Gallager R G. Low-density parity-check codes[J]. *IRE Transactions on Information Theory*, 1962, 8(1): 21-28.
- [2] Davey M C and Mackay D. Low-density parity-check codes over  $GF(q)$ [J]. *IEEE Communications Letters*, 1998, 2(6): 165-167.
- [3] Barnault L and Declercq D. Fast decoding algorithm for LDPC over  $GF(2^q)$ [C]. *IEEE Information Theory Workshop*, Paris, France, 2003: 70-73.
- [4] Wymeersch H, Steendam H, and Moeneclaey M. Log-domain decoding of LDPC codes over  $GF(q)$ [C]. *IEEE International Conference on Communications*, Paris, France, 2004: 772-776.
- [5] Spagnol C, Popovici E, and Marnane W. Hardware

implementation of  $GF(2^m)$  LDPC decoders[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2009, 56(12): 2609-2620.

- [6] Declercq D and Fossorier M. Decoding algorithms for nonbinary LDPC codes over  $GF(q)$ [J]. *IEEE Transactions on Communications*, 2007, 55(4): 633-643.
  - [7] Savin V. Min-Max decoding for non binary LDPC codes[C]. *IEEE International Symposium on Information Theory*, Canada, 2008: 960-964.
  - [8] Zhang X M and Cai F. Reduced-memory forward-backward check node processing architecture for non-binary LDPC decoding[C]. *2011 IEEE 54th International Midwest Symposium on Circuits and Systems*, Seoul, 2011: 1-4.
  - [9] Zhang X M and Cai F. Efficient partial-parallel decoder architecture for quasi-cyclic nonbinary LDPC codes[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2011, 58(2): 402-414.
  - [10] Shuai Z, Jin S, Li L, et al. Layered decoding for non-binary LDPC codes[C]. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, 2010: 481-484.
  - [11] Yeong L U, Chen Y L, Chung J Y, et al. An efficient layered decoding architecture for nonbinary QC-LDPC codes[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2012, 59(2): 385-398.
  - [12] Zhang X M and Cai F. Reduced-complexity decoder architecture for non-binary LDPC codes[J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2011, 19(7): 1229-1238.
  - [13] Yeo E, Pakzad P, Nikolic B, et al. High throughput low-density-parity-check decoder architectures[C]. *Global Telecommunications Conference*, San Antonio, 2001: 3019-3024.
  - [14] Sun Y and Cavallaro J R. VLSI architecture for layered decoding of QC-LDPC codes with high circulant weight[J]. *IEEE Transactions on Very Large Scale Integration Systems*, 2012, DOI: 10.1109/TVLSI.2012.2220388.
  - [15] Minki K, Dongho K, and Ye H L. Serial scheduling algorithm of LDPC decoding for multimedia transmission[C]. *2012 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, Seoul, 2012: 1-4.
  - [16] Zhou B, Zhang L, and Kang J. Array dispersions of matrices and constructions of quasi-cyclic LDPC codes over non-binary fields[C]. *IEEE International Symposium on Information Theory*, Canada, 2008: 1158-1162.
- 杨 威： 男，1988 年生，硕士生，研究方向为 LDPC 码的译码算法。
- 张 为： 男，1975 年生，教授，博士，研究方向包括嵌入式系统、通信与信号处理以及 VLSI 等。