

# 基于变量节点更新的梯度下降比特翻转译码算法

张旋<sup>1,2</sup> 姜超<sup>1</sup> 李晓强<sup>2</sup> 燕莎<sup>2</sup>

(西安电子科技大学计算机学院 西安 710071)<sup>1</sup> (西安理工大学 西安 710082)<sup>2</sup>

**摘要** 由于变量节点自身的可靠性度量值不会随着迭代译码过程中比特被翻转而更新,因此翻转函数值的计算并不准确,进而影响了梯度下降比特翻转译码算法的译码性能。在深入分析梯度下降比特翻转译码算法的基础上,提出了一种基于变量节点更新的加权梯度下降比特翻转译码算法。该算法为翻转函数引入了校验节点的可靠度外信息权重和变量节点可靠性的更新规则,从而使翻转函数值的计算更加精确。仿真结果表明,与现有的梯度下降译码算法相比,在加性高斯白噪声信道下,该算法有效提升了误码率性能。

**关键词** 低密度奇偶校验码,置信传播,加权比特翻转译码,梯度下降比特翻转译码

**中图分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.08.014

## Gradient Descent Bit-flipping Decoding Algorithm Based on Updating of Variable Nodes

ZHANG Xuan<sup>1,2</sup> JIANG Chao<sup>1</sup> LI Xiao-qiang<sup>2</sup> YAN Sha<sup>2</sup>

(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)<sup>1</sup>

(Xi'an University of Technology, Xi'an 710082, China)<sup>2</sup>

**Abstract** The reliability metric of the variable node does not change with flipping the bits during the process of iterative decoding, so the calculation of flipping-function is not accurate, which affects the decoding performance of gradient descent bit-flipping (GDBF) algorithm. Based on the analysis of gradient descent bit-flipping decoding algorithm, a weighted GDBF algorithm was proposed based on updating of variable nodes. This algorithm introduces extrinsic reliability information weights of the check nodes and update rules of the variable nodes for flipping-function, which makes the calculation of flipping-function more accurate. Simulation results show that the BER performance of the proposed algorithm is better than that of the gradient descent bit-flipping decoding algorithm over the additive white Gaussian noise channel.

**Keywords** Low-density parity-check codes, Belief propagation, Weighted bit-flipping decoding, Gradient descent bit-flipping decoding

## 1 引言

低密度奇偶校验 (Low Density Parity Check, LDPC) 码是一种逼近香农限的渐进好码<sup>[1]</sup>, 由于其校验矩阵稀疏、结构灵活且译码复杂度低, 在深空通信和无线通信等领域已被广泛应用<sup>[2-4]</sup>, 并且被 802.11n, 802.16e 和 10GBASE-T 等各种现代通信标准普遍采纳, 其良好的应用前景已经成为研究的热点。

随着 LDPC 码的广泛应用, 其译码算法也日趋成熟, 主要分为两大类<sup>[1]</sup>: 1) 软判决译码算法, 该类算法在译码过程中传送的消息是与后验概率相关的消息, 主要包括置信传播 (Belief Propagation, BP) 译码算法与最小和 (Min Sum, MS) 译码算法<sup>[5]</sup>; 2) 硬判决译码算法, 该类算法在译码过程中传送的消息是 0, 1 比特值, 主要包括比特翻转 (Bit Flipping, BF) 译码

算法及改进的比特翻转算法。BP 译码算法可以使 LDPC 码的性能逼近香农限, 但是在译码过程中涉及较多实数运算, 译码复杂度高, 硬件实现也相对困难; 而硬判决译码算法只涉及逻辑运算, 算法简单、快速且易于硬件实现, 但是纠错性能相对不足。

加权类比特翻转译码算法是利用信道软信息为校验节点引入可靠性权重的一类混合译码算法, 其结合了软判决译码算法和硬判决译码算法的优势, 在提高 BF 译码算法译码性能的同时保持着较低的译码复杂度。Kou 等于 2001 年在 BF 译码算法的基础上引入了校验方程可靠性权重, 提出了加权比特翻转 (Weight Bit Flipping, WBF) 译码算法, 提高了 BF 译码算法的译码性能<sup>[6]</sup>。文献<sup>[7]</sup>同时考虑校验节点传来的外部消息和变量节点的自身消息, 提出了修正加权比特翻转 (Modified Weighted Bit Flipping, MWBF) 译码算法。文献

到稿日期: 2017-05-27 返修日期: 2017-08-22 本文受国家自然科学基金项目 (61271004, 61471286) 资助。

张旋 (1980-), 男, 硕士, 讲师, 主要研究方向为计算机网络与差错控制技术、大数据系统中数据的可靠性技术, E-mail: zxlive@163.com (通信作者); 姜超 (1991-), 男, 硕士, 主要研究方向为差错控制编码技术, E-mail: 66877905@qq.com; 李晓强 (1980-), 男, 硕士, 讲师, 主要研究方向为嵌入式系统设计, E-mail: mrglhome@xaut.edu.cn; 燕莎 (1975-), 女, 硕士, 副教授, 主要研究方向为数字电路系统设计, E-mail: 1446067285@qq.com。

[8]提出变量节点对应的校验方程权重信息来自于除变量节点本身之外的其他变量节点,使权重的计算更加准确。Guo 等于 2005 年提出了基于可靠度比率的加权比特翻转(Reliability Ratio Based Weighted Bit Flipping, RRWBF)译码算法,相比于 WBF 译码算法可以提高 1 dB 编码增益,且复杂度几乎不变<sup>[9]</sup>。文献[10-11]中校验节点的权重值分别取自于所有变量节点的初始信道软信息幅度的平均值和幅度值之和,进一步提高了纠错性能。文献[12-14]从最优化理论的角度将译码过程转换为非线性函数最优化问题,并基于梯度下降算法提出了一种性能更优的梯度下降比特翻转(Gradient Descent Bit Flipping, GDBF)译码算法。

GDBF 译码算法是基于最优化理论的一类混合译码算法,将译码过程转换为目标函数的最优化过程,其纠错性能优于加权比特翻转译码算法。在深入分析 WBF 和 GDBF 译码算法的基础上,通过增加校验节点的可靠度外信息权重(Extrinsic Reliability Information, ERI)和变量节点可靠性的更新规则,本文提出一种改进的梯度下降比特翻转译码算法,该算法能获得更好的译码性能。

## 2 梯度下降比特翻转译码算法

### 2.1 基本定义

对于二元 LDPC 码( $N, K$ ),其码长为  $N$ ,校验位为  $M = N - K$ 。 $H$  为  $M \times N$  的校验矩阵,将  $H$  中第  $j$  行中所有“1”的位置表示为  $A(j) = \{i: h_{ij} = 1\}$ ,即参与第  $j$  个校验方程计算的所有变量节点;第  $i$  列中所有“1”的位置表示为  $B(i) = \{j: h_{ij} = 1\}$ ,即校验第  $i$  个变量节点的所有校验节点;对于规则 LDPC 码, $H$  的列重  $w_i$  等于  $|B(i)|$ ,而行重  $w_r$  等于  $|A(j)|$ <sup>[10]</sup>。码字序列  $c = \{c_1, c_2, c_3, \dots, c_n\}$ ,经过 BPSK 调制之后,可以得到  $u = \{u_1, u_2, u_3, \dots, u_n\}$ ,  $u_i = 1 - 2c_i$ ,  $u$  经过加性高斯白噪声(Additive White Gaussian Noise, AWGN)信道之后得到接收码字序列  $y = \{y_1, y_2, y_3, \dots, y_n\}$ ,  $y_i = (1 - 2c_i) + v_i$ ,其中  $v_i$  是均值为 0、功率谱密度为  $\frac{N_0}{2}$ (即方差  $\sigma^2 = N_0/2$ )的高斯白噪声<sup>[11]</sup>。根据硬判决规则  $z_i = \frac{(1 - \text{sgn}(y_i))}{2}$ ,可以得到硬判决输出序列  $z = \{z_1, z_2, z_3, \dots, z_n\}$ 。伴随式表示为  $s = \{s_1, s_2, s_3, \dots, s_m\} = Hz^T$ 。

### 2.2 梯度下降比特翻转译码算法

梯度下降比特翻转译码算法将硬判决规则改为  $z_i = \text{sgn}(y_i)$ ,  $z_i = \{-1, +1\}$ ,将校验矩阵  $H$  中第  $j$  行对应的校验方程表示为  $s_j = \prod z_i$ ,其中,  $s_j = 1$  表示校验方程满足约束条件,  $s_j = -1$  表示校验方程不满足约束条件。

GDBF 译码算法利用了最大似然译码准则,如式(1)所示,即在给定接收码字  $y$  的条件下,找出可能性最大的码字序列  $z$ ,将其转化为梯度下降算法的目标函数,如式(2)所示<sup>[12]</sup>。

$$z = \arg \max_{z \in \bar{z}} \sum_{i=1}^n z_i y_i \quad (1)$$

$$f(z) \triangleq \sum_{i=1}^n z_i y_i + \sum_{j=1}^m \prod_{i \in A(j)} z_i \quad (2)$$

采用梯度下降算法逐步逼近,求取目标函数的最大值。对式(2)中的  $z_k$  ( $k = \{1, 2, 3, \dots, n\}$ )求偏导数,得到式(3)<sup>[12]</sup>:

$$\frac{\partial}{\partial z_k} f(z) = y_k + \sum_{j \in B(k)} \prod_{i \in A(j) \setminus k} z_i \quad (3)$$

两端同时乘以  $z_k$ ,得到式(4):

$$z_k \frac{\partial}{\partial z_k} f(z) = z_k y_k + \sum_{j \in B(k)} \prod_{i \in A(j)} z_i \quad (4)$$

偏导数为正数时才能保证目标函数递增,而偏导数为负数时目标函数向反方向递增。GDBF 译码算法的译码过程可以看作通过梯度下降算法逼近  $-f(z)$  最小值的过程。

整理式(4),得到梯度下降比特翻转译码算法的翻转函数,如式(5)所示。GDBF 译码算法将每个翻转函数值作为可靠性判断依据<sup>[12]</sup>。

$$\Delta_n^{(GD)}(z) \triangleq z_n y_n + \sum_{j \in B(n)} \prod_{i \in A(j)} z_i \quad (5)$$

GDBF 译码算法的具体步骤如算法 1 所示。首先,计算校验方程,如果所有校验方程均满足约束条件,输出译码结果,结束译码,否则继续根据式(5)计算翻转函数  $\Delta_n^{(GD)}(z)$ ;接着,找出翻转函数值最小的比特,进行比特翻转;算法的终止条件是所有校验方程均得到满足,或者达到了预设的最大迭代次数  $I_{\max}$ 。

### 算法 1 梯度下降比特翻转译码算法

```

1. procedure Decode(z)
2. I=0
3. repeat
4.   for m=1:M do //计算 ERI
5.      $s_m = \prod_{n \in A(m)} z_n$ 
6.   end for
7.   if S=1 or I=Imax then
8.     Finished
9.   else
10.    for n=1:N do //计算翻转函数
11.       $\Delta_n^{(GD)}(z) \triangleq z_n y_n + \sum_{m \in B(n)} \prod_{i \in A(m)} z_i$ 
12.    end for
13.    for n=arg minn ∈ [1, N] Δn(GD)(z) do
14.       $z_n = -z_n$  //翻转比特
15.    end for
16.  end if
17.  I=I+1
18. until Finished
19. end procedure

```

## 3 改进的梯度下降比特翻转译码算法

### 3.1 梯度下降比特翻转译码算法的改进思路

将 GDBF 译码算法的翻转函数式(5)看作由校验节点提供的可靠度外信息  $\sum_{j \in B(n)} \prod_{i \in A(j)} z_i$  与由变量节点自身提供的可靠度内信息  $z_n y_n$  两部分组成。为了区分校验节点的不同可靠度,本文为 GDBF 译码算法增加了可靠度外信息权重 ERI,从而得到一种加权的梯度下降比特翻转函数,如式(6)所示。该翻转函数同时考虑了变量节点自身的信道软信息和校验和信息,将变量节点自身的信道软信息作为校验节点校验和的可靠性度量,根据各个校验方程的可靠性度量计算每个码元的翻转函数  $\Delta_n^{(GD)}(z)$ ,该值的大小可评价判决后每个码元的可靠程度,对最不可靠的码元进行翻转。因此,该翻转函数的计算更加精确,翻转比特的准确性更高。

$$\Delta_n^{(GD)}(z) \triangleq z_n y_n + \sum_{m \in B(n)} \omega_{mm} \prod_{i \in A(m)} z_i \quad (6)$$

其中,可靠度内信息  $z_n y_n$  由硬判决结果  $z_n$  和初始信道软信

息  $y_n$  的相关性决定,当硬判决结果  $z_n$  与初始信道软信息  $y_n$  的符号一致时,硬判决结果很可能是正确的;并且初始信道软信息幅度  $|y_n|$  越大,硬性判决的可靠性越高;相反,  $|y_n|$  越小,硬判决的可靠性越低(硬判决出错的可能性越高),比特出错的可能性越大。可靠度外信息  $\sum_{m \in B(n)} \omega_{mn} \prod_{n \in A(m)} z_n$  由不满足校验方程的数量和可靠性权重  $\omega_{mn}$  共同决定,其中,可靠性权重  $\omega_{mn}$  为正值。显然,  $\sum_{m \in B(n)} \omega_{mn} \prod_{n \in A(m)} z_n$  越小,比特出错的可能性越大。因此,综合分析变量节点的可靠度内信息和校验节点提供的可靠度外信息,可得翻转函数值  $\Delta_n^{(GD)}(z)$  越小,比特出错的可能性越大,该比特在迭代译码过程中需要被翻转。

由于迭代过程中不可靠比特被翻转之后,其参与的校验方程在下次迭代中会被更新,然而被翻转比特的自身可靠度信息在整个迭代译码过程中始终保持不变。迭代过程中被翻转的比特被认为是最不可靠的比特,因此该比特被翻转之后应该具有较高的可靠性。由此,可以动态更新比特的可靠度内信息,对每轮迭代过程中被翻转的比特所对应的初始信道软信息幅度扩大  $\beta$  倍,增加该比特的可靠性。

### 3.2 改进的梯度下降比特翻转译码算法

基于增加可靠度外信息权重和动态更新可靠度内信息,提出一种改进的梯度下降比特翻转译码算法。其中,变量节点对应的校验方程权重信息是除变量节点自身之外,对所有其他相关变量的节点信道软信息求和后取平均值得到,  $\omega_{mn}$  可作为可靠度外信息 ERI 的权重,如式(7)所示。该权重的选择涉及该校验方程所有变量节点的信道软信息,因此可获得更好的性能。

$$\omega_{mn} = \frac{1}{w_r - 1} \sum_{n' \in A(m) \setminus n} |y_{n'}| \quad (7)$$

改进的梯度下降比特翻转译码算法的具体步骤如算法 2 所示。首先,根据硬判决结果计算校验方程,若所有校验和均满足约束条件,则输出译码结果,结束译码,否则继续根据式(6)计算翻转函数  $\Delta_n^{(GD)}(z)$ ;接着找出翻转函数值最小的比特进行翻转,同时对翻转比特对应的变量节点信道软信息幅度扩大  $\beta$  倍;算法的终止条件是校验和为 0,或者达到了预设的最大迭代次数  $I_{\max}$ 。

#### 算法 2 改进的梯度下降比特翻转译码算法

```

1. procedure Decode(z)
2. I=0
3. repeat
4.   for m=1:M do//计算校验节点的权重
5.      $\omega_{mn} = \frac{1}{w_r - 1} \sum_{n' \in A(m) \setminus n} |y_{n'}|$ 
6.   end for
7.   for m=1:M do//计算 ERI
8.      $s_m = \prod_{n \in A(m)} z_n$ 
9.   end for
10.  if S=1 or I=Imax then
11.    Finished
12.  else
13.    for n=1:N do//计算翻转函数
14.       $\Delta_n^{(GD)}(z) \triangleq z_n y_n + \sum_{m \in B(n)} \omega_{mn} \prod_{n \in A(m)} z_n$ 
15.    end for
16.    for n=arg minn ∈ [1, N] Δk(GD)(z) do

```

```

17.       $z_n = -z_n$ 
18.       $|y_n| = \beta |y_n|$  //更新可靠性
19.    end for
20.  end if
21. I=I+1
22. until Finished
23. end procedure

```

## 4 仿真实验分析

为了对比改进的梯度下降比特翻转译码算法与梯度下降比特翻转译码算法,以及几种典型的加权比特翻转译码算法的译码性能,仿真实验选取码率为 0.5、列重为 3 的规则 PEG-LDPC 码(1008, 504)  $C_1$  和规则 PEG-LDPC 码(504, 252)  $C_2$  进行仿真<sup>[15]</sup>。

修正加权比特翻转 MWBF 译码算法、改进的修正加权比特翻转 IMWBF 译码算法以及平均幅度加权比特翻转 AMWBF 译码算法中的最优权重因子  $\alpha$  设置为 0.4<sup>[10]</sup>。对于改进的梯度下降比特翻转译码算法,每轮迭代中被翻转的比特所对应的信道软信息幅度扩大的倍数  $\beta$  由仿真实验得到,设置为 1.3。最大迭代次数设为 100,不同信噪比下译 50000 帧,收集错误比特信息。

图 1 和图 2 分别给出了译码  $C_1$  和译码  $C_2$  时,改进的梯度下降比特翻转译码算法、梯度下降比特翻转 GDBF 译码算法、平均幅度加权比特翻转 AMWBF 译码算法、改进的修正加权比特翻转 IMWBF 译码算法与加权比特翻转 WBF 译码算法这 5 种译码算法的译码性能。

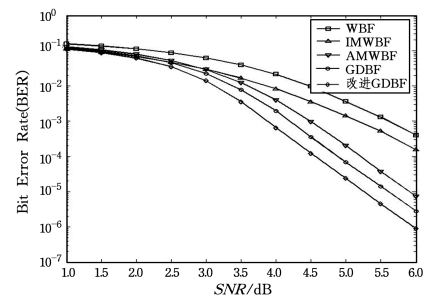


图 1 几种译码算法针对码  $C_1$  的译码性能

Fig. 1 Performances of several decoding algorithms for code  $C_1$

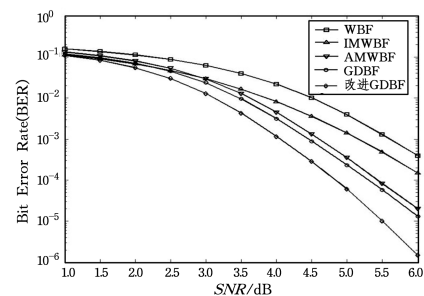


图 2 几种译码算法针对码  $C_2$  的译码性能

Fig. 2 Performances of several decoding algorithms for code  $C_2$

仿真结果表明,在对两种码字译码时,几种译码算法的译码性能由低到高依次是加权比特翻转 WBF 译码算法、改进的修正加权比特翻转 IMWBF 译码算法、平均幅度加权比特翻转 AMWBF 译码算法、梯度下降比特翻转 GDBF 译码算法

和改进的梯度下降比特翻转译码算法。相对于梯度下降比特翻转译码算法,改进的梯度下降译码算法能获得更好的译码性能,这是因为本文算法加入了变量节点可靠性的更新规则和校验节点的可靠度外信息 ERI 权重,使得翻转函数的计算更加精确,因此增加翻转比特的准确性提高了译码性能。

由前文可知, $H$  为  $M \times N$  的校验矩阵,其行重和列重分别为  $w_r$  和  $w_c$ ,其中  $Mw_r = Nw_c$ ,表示校验矩阵  $H$  中“1”的数量。通过深入分析几种译码算法,统计得到每轮迭代的运算次数,表 1 列出了几种译码算法每轮迭代过程中比较运算、加法运算与乘法运算 3 种运算的次数。通过表 1 可得,改进的梯度下降比特翻转译码算法的运算次数与梯度下降比特翻转 GDBF 译码算法基本相同,其余 3 种译码算法的运算次数大致相同。

表 1 几种译码算法每轮迭代的运算次数

Table 1 Operation number of several decoding algorithms for per iteration

| 译码算法    | 比较运算  | 加法运算                 | 乘法运算                      |
|---------|-------|----------------------|---------------------------|
| WBF     | $N-1$ | $M(w_r-1)+N(2w_c-1)$ | $3Nw_c$                   |
| IMWBF   | $N-1$ | $M(w_r-1)+2Nw_c$     | $3Nw_c+N$                 |
| AMWBF   | $N-1$ | $M(w_r-1)+2Nw_c$     | $3Nw_c+N$                 |
| GDBF    | $N-1$ | $Nw_c$               | $M(w_r-1)+Nw_c(w_r-1)+N$  |
| 改进 GDBF | $N-1$ | $Nw_c$               | $M(w_r-1)+Nw_c(w_r-1)+2N$ |

表 2 列出了译 50000 帧码字  $C_1$  时(最大迭代次数为 100),在几种译码算法下译每帧码字的平均迭代次数。

表 2 几种译码算法针对码  $C_1$  的平均迭代次数Table 2 Average iteration number of several decoding algorithms for Code  $C_1$ 

| SNR/dB | WBF   | IMWBF | AMWBF | GDBF  | 改进 GDBF |
|--------|-------|-------|-------|-------|---------|
| 1.0    | 100   | 100   | 100   | 100   | 100     |
| 2.0    | 100   | 100   | 99.99 | 99.99 | 99.99   |
| 3.0    | 99.99 | 99.98 | 98.94 | 98.86 | 97.55   |
| 4.0    | 98.63 | 98.21 | 81.93 | 80.90 | 73.21   |
| 5.0    | 79.63 | 78.32 | 48.58 | 48.73 | 42.96   |
| 6.0    | 40.44 | 40.77 | 25.68 | 25.93 | 24.10   |

与 WBF,IMWBF,AMWBF 和 GDBF 译码算法相比,本文提出的改进梯度下降比特翻转译码算法在提高译码性能的前提下,平均迭代次数有所降低。这是因为改进的梯度下降比特翻转译码算法的翻转函数规则提高了翻转比特的可靠度,错误比特被正确翻转的概率变大,因此译码器每轮迭代译码更加有效,进而减少了译码的迭代次数。

**结束语** 在深入分析 GDBF 译码算法的基础上,通过增加校验节点的可靠度外信息权重和变量节点可靠性的更新规则,提出了一种改进的梯度下降比特翻转译码算法,该算法对翻转函数的计算更加精确。与现有的混合译码算法相比,改进的梯度下降译码算法能获得更好的译码性能。

## 参 考 文 献

[1] JOHNSON S J. Introducing low-density parity-check codes[EB/OL]. <http://sigpromu.org/sarah/SJohnsonLDPCintro.pdf>.

[2] LIVA G, PAOLINI E, COLA T D, et al. Codes on high-order fields for the CCSDS next generation uplink in Proc[C]// Proceedings of IEEE Conference on Signal Processing for Space Communications Workshop (SPSC). New York: IEEE Press, 2012:44-48.

[3] DING Z, IVAN B. FPGA-based rate-compatible LDPC codes for the next generation of optical transmission systems[J]. IEEE Photonics Journal, 2016, 8(5):1-8.

[4] ZHANG M X, LIC X, KIM S. Efficient utilization of rateless LDPC codes for satellite broadcasting services[C]// Proceedings of IEEE Conference on Wireless Communications & Signal Processing (WCSP). New York: IEEE Press, 2016:1-5.

[5] FABIÁN A, JAVIER V, VICEN Ç A. Reduced-complexity minimum-sum algorithm for decoding LDPC codes with low error-floor[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2014, 61(7):2150-2158.

[6] KOU Y, LIN S, 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.

[7] ZHANG J, FOSSORIER M. A modified weighted bit-flipping decoding of low-density Parity-check codes[J]. IEEE Communications Letters, 2004, 8(3):165-167.

[8] JIANG M, ZHAO C, SHI Z, et al. An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes[J]. IEEE Communications Letters, 2005, 9(9):814-816.

[9] GUO F, HANZO L. Reliability ratio based weighted bit-flipping decoding for LDPC codes[C]// Proceedings of IEEE Conference on Vehicular Technology Conference. New York: IEEE Press, 2005:709-713.

[10] 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, 35(11):2572-2578. (in Chinese)

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,2013,35(11):2572-2578.

[11] 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, 36(4):752-757. (in Chinese)

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,2014,36(4):752-757.

[12] WADAYAMA T, NAKAMURA K, YAGITA M, et al. Gradient descent bit flipping algorithms for decoding LDPC codes[J]. IEEE Transactions on Communications, 2010, 58(6):1610-1614.

[13] KHOA L, FAKHREDDINE G, DAVID D, et al. Efficient hardware implementation of probabilistic gradient descent bit-flipping[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2017, 64(4):906-917.

[14] LI H, DING H, ZHENG L H. An escaping scheme for gradient descent bit-flipping decoding of LDPC codes[C]// Proceedings of IEEE Conference on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). New York: IEEE Press, 2016:2026-2030.

[15] DAVID J C. Encyclopedia of sparse graph codes[EB/OL]. <http://www.inference.phy.cam.ac.uk/mackay/codes>.