

文章编号:1001-2486(2006)05-0052-06

## 基于变量节点串行消息传递的 LDPC 码译码研究\*

雷菁,文磊,唐朝京

(国防科技大学 电子科学与工程学院,湖南长沙 410073)

**摘要:**针对标准 LDPC 码译码中洪水消息传递机制的不足,提出以串行机制进行消息传递,按照变量节点的顺序进行消息处理和传递,对每个变量节点同时接收校验消息和发送变量消息。该方法使更新的消息能够很快进入当前迭代计算,改善了 LDPC 迭代译码的收敛性能。通过对几种常用译码算法的仿真比较,验证了在复杂度不增加的情况下,该方法性能优于其它几种最大后验概率准则的译码方法,且算法收敛快,是一种能较好兼顾性能与实现复杂度的译码方法。

**关键词:**LDPC 译码;串行消息传递;迭代译码;变量节点;校验节点

**中图分类号:**TN911.22 **文献标识码:**A

## A Serial Decoding Schedule for LDPC Based on a Serial Update of Bit Nodes' Messages

LEI Jing, WEN Lei, TANG Chao-jing

(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract** Contrapositing the shortcoming of "flooding schedule" in the standard LDPC (low-density parity-check) decoding algorithm, this paper presents a serial decoding schedule, which shuffles the message passing schedule of standard LDPC decoding algorithm. The new schedule processes and propagates messages according to a serial update order of bit nodes. Each bit node receives check messages and sends bit messages in the same iteration so that the updated messages could join iterative process immediately. As a result, the convergence property is improved. Simulation results show that, under the same decoding complexity, the proposed version is optimal in the sense of MAP decoding and converges faster than the flooding schedule. Meanwhile, it keeps the appropriate decoding complexity, so it is a good method that can offer better trade-offs between performance and complexity.

**Key words:** decoding for low-density parity-check(LDPC) codes; serial schedule; iterative decoding; bit nodes; check nodes

LDPC 译码器一般所采用的是洪水消息传递机制(flooding schedule),即迭代过程中每个变量节点向与其相连的校验节点发送变量消息;接着每个校验节点向与其相连的变量节点发送校验消息,这是一种基于并行处理的消息传递机制,每轮迭代过程中所有的变量节点和校验节点均同时处理和传送消息,实际应用中会占用大量硬件资源,尤其需要大容量的存储器保存中间变量。洪水译码算法是目前普遍采用的 LDPC 译码算法,但并不表明其性能最优,研究结果表明其消息传递的收敛特性并非最佳,正确译码所需的迭代次数很大,而每一次迭代需要许多实数运算,如果收敛较慢,那么为了获得期望的性能就需要大量的迭代,这导致在高速通信中不希望看到的高运算量及很长的译码延时。因此研究能进一步改进译码性能的消息传递机制是很有意义的。Yongyi Mao 和 Anmir H. Banihasheau 提出了基于概率方案的消息传递机制<sup>[3]</sup>,考虑了码的 Tanner 图的结构,特别是变量节点的围长分布,使得变量节点输出消息的平均更新频率与通过该变量节点的最短环的长度成正比。对非洪水译码机制也进行了研究,如 Juntan Zhang 等提出置乱(shuffled)的 BP 译码算法<sup>[1]</sup>,在每次迭代中用局部更新的计算值参与变量消息的计算,并证明了改进方法收敛性不低于标准 BP 算法;Eran Shron 提出了基于校验节点的串行方案的消息传递机制<sup>[2]</sup>,使消息传递的收敛更快,本文在此基础上,将洪水消息传递机制转换成以变量节点为序的串行消息传递机制,推导了串行消息传递机制下的迭代公式,给出基于变量节点的串行机制译码流

\* 收稿日期:2006-04-16

基金项目:国家自然科学基金资助项目(60572176)

作者简介:雷菁(1968—),女,副教授,硕士。

程,并结合仿真对洪水机制算法以及两种串行机制算法进行了比较分析。

## 1 基于变量节点的串行消息传递机理

### 1.1 洪水机制消息传递简述

设 LDPC 码的一致校验矩阵为一个  $M$  行  $N$  列的稀疏矩阵  $\mathbf{H}_{MN}$ ,所有码字  $\mathbf{v}$  均满足  $\mathbf{v}\mathbf{H}^T = 0$ 。由线性码特点可知,校验矩阵的每一行表示一个校验约束,其中所有非零元素对应的码元变量构成一个校验集,由一个校验方程表示。校验矩阵的每一列表示一个码元符号参与的校验约束。令  $N(c) = \{n: \mathbf{H}_{MN} = 1\}$  为矩阵  $\mathbf{H}$  第  $m$  行中非零元所在位置,即被第  $m$  个校验元  $c$  监督的变量节点集合; $N(v) = \{m: \mathbf{H}_{MN} = 1\}$  为矩阵  $\mathbf{H}$  第  $n$  列中非零元所在位置,即与第  $n$  个变量节点  $v$  有约束关系的校验元集合。

LDPC 码的 SISO 译码过程是一种迭代概率译码算法,可用由校验矩阵决定的 Tanner 图来描述,如图 1 所示,该图由两个不相交的集合组成——变量节点( $v_i$ )代表了码字的各比特,校验节点( $c_i$ )则代表了校验约束,两个节点集合之间的连线(边)表示一致校验矩阵中的非零元。

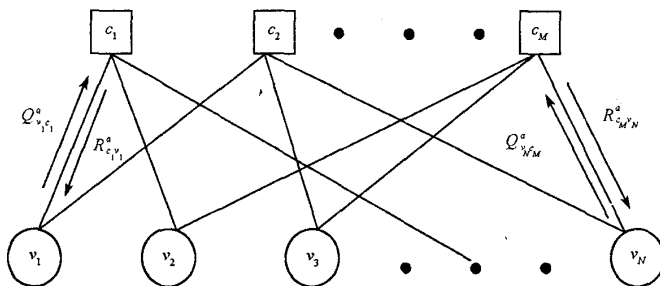


图 1 Tanner 图上的译码消息流的迭代传递

Fig. 1 The iterative operation of message passing based on Tanner graph

洪水译码算法的每一轮迭代都需要对码字中各个比特关于接收码字和信道参数的后验概率进行估算,因此准确地说,它是一种逐比特计算最大后验概率(MAP)算法<sup>[8]</sup>,而整个迭代过程则可以看成在图 1 所示的 Tanner 图上进行的消息传递过程。每个变量节点计算并向与其相连的校验节点发送变量消息  $Q_{v,c}^a$  (对二元码而言,  $a \in \{0, 1\}$ ),该函数为在已知与变量节点  $v$  相连的除  $c$  外其它校验节点所发送的校验消息  $\{R_{c',v}^a, c' \in N(v) \setminus c\}$  前提下,变量节点为  $a$  的条件概率;接着每个校验节点计算并向与其相连的变量节点发送校验消息  $R_{c,v}^a$ ,同理,它是在已知变量节点取值为  $a$  以及与校验节点  $c$  相连的其它变量消息  $\{Q_{v',c}^a, v' \in N(c) \setminus v\}$  的前提下,校验关系成立的条件概率。这种迭代算法中很重要的一点是某节点  $u$  沿某边  $e$  发送的消息与上次  $u$  从  $e$  接收到的消息无关,而决定于和  $u$  相连的其它边上接收的信息,这就保证了在任一条边上,只有外来消息传递。

一轮迭代之后,需对每个变量节点计算它关于接收值和码结构的对数似然比概率  $Q_v$ ,并做出判决:若  $Q_v < 0$ ,则  $\hat{v}_i = 1$ ;否则  $\hat{v}_i = 0$ 。由此可以得到对发送码字的一个估计  $\hat{\mathbf{v}} = (v_1, v_2, \dots, v_N)$ 。再计算伴随式  $\mathbf{S} = \mathbf{v}\mathbf{H}^T$ ,若  $\mathbf{S} = 0$ ,则认为译码成功,结束迭代过程;否则继续迭代直至达到预定的最大迭代次数。

洪水译码算法的每轮迭代都是并行处理消息的过程:变量节点同时处理和传送变量消息,校验节点同时处理和传送校验消息。在每一轮迭代中,校验消息  $R_{c,v}^a$  和变量消息  $Q_{v,c}^a$  的计算都是利用上一轮迭代的结果,每个校验节点和变量节点在本轮迭代中已更新的消息只能在下一轮迭代计算中被使用,为加快算法速度,本文将在同一次迭代中充分利用这些已更新的消息。

### 1.2 串行消息传递算法推导

对于均值为 0、方差为  $\sigma^2$  的 AWGN 信道,采用 BPSK 调制方式,将码字  $\mathbf{v} = (v_1, v_2, \dots, v_N) \in V$  按  $x_i = 2v_i - 1, v_i \in \{0, 1\} (1 \leq i \leq N)$  关系,映射为发送序列  $\mathbf{x} = (x_1, x_2, \dots, x_N)$ ,经过信道传输后,接收变量

为  $y_i = x_i + g_i (1 \leq i \leq N)$ , 其中  $g_i$  为均值为 0、方差为  $\sigma^2$  的独立分布 Gauss 噪声。LDPC 码的图模型结构仍如图 1 所示, 设各校验关系相互无关, 且设信道模型为离散无记忆信道 (DMC), 则和积译码算法中校验消息及变量消息的具体计算公式分别为<sup>[5]</sup>:

$$\begin{cases} R_w^0 = \frac{1}{2} + \frac{1}{2} \prod_{v' \in N(c) \setminus v} (1 - 2Q_{v'}^1) \\ R_w^1 = \frac{1}{2} - \frac{1}{2} \prod_{v' \in N(c) \setminus v} (1 - 2Q_{v'}^1) \end{cases} \quad (1)$$

$$\begin{cases} Q_w^0 = P(z_i = 0 | \{c'\} |_{c' \in N(v) \setminus c}, y_i) = K_w P_v^0 \prod_{c' \in N(v) \setminus c} R_{c'}^0 \\ Q_w^1 = K_w P_v^1 \prod_{c' \in N(v) \setminus c} R_{c'}^1 \end{cases} \quad (2)$$

其中,  $K_w$  是归一化因子, 保证  $Q_w^0 + Q_w^1 = 1$ 。

定义对数似然比:  $L(P_v) = \log\left(\frac{P_v^0}{P_v^1}\right)$ ,  $L(Q_w) = \log\left(\frac{Q_w^0}{Q_w^1}\right)$ ,  $L(R_w) = \log\left(\frac{R_w^0}{R_w^1}\right)$ ,  $L(Q_v) = \log\left(\frac{Q_v^0}{Q_v^1}\right)$ 。

由式(1)可得,  $1 - 2R_w^1 = \prod_{v' \in N(c) \setminus v} (1 - 2Q_{v'}^1)$ , 再根据 Tanh 准则<sup>[6]</sup> 及公式  $\tanh\left[\frac{1}{2} \log\left(\frac{p_0}{p_1}\right)\right] = p_0 - p_1 = 1 - 2p_1$  (其中  $p_0 + p_1 = 1$ ) 得:

$$L(R_w) = 2 \tanh^{-1} \prod_{v' \in N(c) \setminus v} \tanh\left(\frac{1}{2} L(Q_{v'})\right)$$

令

$$\phi(x) = -\log \tanh\left(\frac{x}{2}\right) = \log \frac{e^x + 1}{e^x - 1}$$

则有

$$L(R_w) = \phi^{-1} \sum_{v' \in N(c) \setminus v} \phi(L(Q_{v'})) \quad (3)$$

将式(2)的  $Q_w^0$  和  $Q_w^1$  代入  $L(Q_w)$  的定义式, 可得:

$$L(Q_w) = \log \frac{Q_w^0}{Q_w^1} = \log \frac{K_w P_v^0 \prod_{c' \in N(v) \setminus c} R_{c'}^0}{K_w P_v^1 \prod_{c' \in N(v) \setminus c} R_{c'}^1} = L(P_v) + \sum_{c' \in N(v) \setminus c} L(R_{c'}) \quad (4)$$

这样, 译码算法中最后一步的译码判决公式变为:

$$L(Q_v) = \log \frac{Q_v^0}{Q_v^1} = L(P_v) + \sum_{c \in N(v)} L(R_w) \quad (5)$$

联立(4)、(5)两式得:

$$L(Q_w) = L(Q_v) - \sum_{c \in N(v)} L(R_w) + \sum_{c' \in N(v) \setminus c} L(R_{c'}) = L(Q_v) - L(R_w) \quad (6)$$

通过对洪水消息传递机制中几个关键计算量的解析, 可以发现在迭代计算中, 只要确定了变量节点的消息  $L(Q_w)$  的初始值(可由信道信息获取), 就可利用式(3)对与之相联的校验节点消息  $L(R_w)$  进行更新, 再利用式(6), 更新后的  $L(R_w)$  马上可对该变量节点的消息  $L(Q_w)$  进行更新计算, 依此类推, 每个更新后的节点消息又可马上用于本次迭代中其他节点消息的计算中。以图 1 为例, 设各变量节点  $v_i (i = 1, \dots, N)$  的消息初始值为<sup>[1]</sup>:  $L(Q_{v_i}) = L(P_{v_i}) = \frac{2y_i}{\sigma^2}$ , 与第 1 个变量节点有关的变量消息有  $L(Q_{v_1 c_1}), L(Q_{v_1 c_2})$  (开始时为初始值), 根据式(3)计算与第 1 个变量节点有关的校验消息  $L(R_{c_1 v_1}), L(R_{c_2 v_1})$ , 紧接着由式(4)更新变量消息  $L(Q_{v_1 c_1}), L(Q_{v_1 c_2})$ , 而更新后的变量消息  $L(Q_{v_1 c_1})$  在第 2 个变量节点相关校验消息的计算中马上被使用(如  $L(R_{c_1 v_2}) = \phi^{-1} \{ \phi[L(Q_{v_1 c_1})] + \phi[L(Q_{v_1 c_1})] \}$ ), 而不必等到下一轮迭代, 从而加快了算法收敛速度。

### 1.3 基于变量节点的串行机制译码算法

根据上述推导,同时考虑译码判决需计算每个变量节点的对数似然比函数,本文提出每轮迭代中,按照变量节点的一定顺序,对每个变量节点同时接收校验消息  $L(R_w)$  和发送变量消息  $L(Q_w)$ ,本轮已计算出的校验消息  $L(R_w)$  和变量消息  $L(Q_w)$  直接用于后续节点消息中的更新计算中。

为便于计算,还可将消息空间变换为极坐标表示,将  $L(Q_w)$  分解为极性符号与绝对值两部分,  $L(Q_w) = \alpha_w \beta_w$ , 其中,  $\alpha_w = \text{sign}[L(Q_w)]$ ,  $\beta_w = \text{abs}[L(Q_w)]$ , 这样式(3)转换为:

$$L(R_w) = \prod_{v' \in N(c) \setminus v} \alpha_{v'} \phi^{-1} \left[ \sum_{v' \in N(c) \setminus v} \phi(\beta_{v'}) \right]$$

具体译码过程如下:

#### (1) 初始化

根据接收变量  $y_i$ , 计算  $v_i = 0$  或  $1$  (即  $x_i = -1$  或  $1$ ) 的条件概率  $P_v^0$  或  $P_v^1$ , 再由校验矩阵  $H$  (若  $h_{ij} = 1$ , 即变量节点和校验节点相连) 定义变量消息:  $L(Q_w) = L(P_v) = \frac{2y_i}{\sigma^2}$ 。

#### (2) 迭代过程

以  $v_1, v_2, v_3, \dots, v_N$  的次序串行进行消息更新, 每个变量节点的计算由内外两个循环构成。

- 外循环——计算变量消息的极性及其幅值:

$$L(Q_w) = \alpha_w \beta_w$$

其中,  $\alpha_w = \text{sign}[L(Q_w)]$ ,  $\beta_w = \text{abs}[L(Q_w)]$ 。

- 内循环——计算与变量节点相连的校验消息及更新变量消息:

$$L(R_w) = \prod_{v' \in N(c) \setminus v} \alpha_{v'} \cdot \phi^{-1} \left[ \sum_{v' \in N(c) \setminus v} \phi(\beta_{v'}) \right]$$

$$L(Q_w) = L(P_v) + \sum_{c \in N(v) \setminus c} L(R_c)$$

#### (3) 译码判决

$$L(Q_v) = L(P_v) + \sum_{c \in N(v)} L(R_w)$$

根据  $L(Q_v)$  做出判决: 若  $L(Q_v) > 0$ , 则  $\hat{v}_i = 0$ ; 否则  $\hat{v}_i = 1$ 。由此可以得到对发送码字的一个估计  $\hat{v} = (v_1, v_2, \dots, v_N)$ 。再计算伴随式  $S = vH^T$ , 如果  $S = 0$ , 则认为译码成功, 结束迭代过程; 否则继续迭代。若迭代次数达到预先设定的最大次数  $\max$  仍未成功, 则宣告失败, 终止迭代。

这种方法按照变量节点的串行顺序进行消息处理和传递, 能保证消息马上得到传递, 而不像洪水消息传递机制, 更新的消息只能等到下次迭代过程才能传出去, 改进了消息的收敛特性。另外, 由式(5)说明译码判决函数  $L(Q_v)$  可与信道初始化信息  $L(P_v)$  存储在同一单元中, 这样可进一步节省存储资源。

## 2 仿真结果

为了验证算法性能, 分别应用基于变量节点的串行传递算法(以串行算法 II 表示)、洪水译码算法、最小和译码算法和文献[2]的串行译码算法(以串行算法 I 表示)进行了对比仿真实验, 仿真中采用的信道都是二进制输入的加性高斯白噪声信道, 调制方式是基带 BPSK 调制, 采用的码型为 (1024, 3, 6) 规则 LDPC 码, 译码最大迭代次数  $\max$  分别取 100 次和 20 次, 仿真环境是 SPW 仿真系统。

图 2、图 3 分别显示了 1/2 码率的 (1024, 3, 6) 规则 LDPC 码在不同最大迭代次数情况下的误码性能。由图 2 可以看出, 在  $\max = 100$  的情况下, 洪水译码算法和两种串行译码算法在性能上差别不大, 但它们的性能都明显好于最小和译码算法, 尤其在中等信噪比的情况下, 当  $E_b/N_0$  介于 2 ~ 2.5 dB 时, 前三种译码算法误码率甚至比最小和译码算法误码率低两个数量级。图 3 显示当  $\max = 20$  的情况下, 最小和译码算法性能仍是最差, 值得注意的是, 这时两种串行译码算法性能却比洪水译码算法明显改善。特别当  $E_b/N_0 \geq 2$  dB 时, 串行译码算法性能上的优势逐渐显现出来, 串行译码算法的性能比洪水译码算法要好一个数量级左右, 当误码率达到  $10^{-5}$  时, 相对于洪水译码算法, 两种串行译码算法均能带来 0.4 dB 左右

的增益,并且随着信噪比的增大,性能差异有增大的趋势。

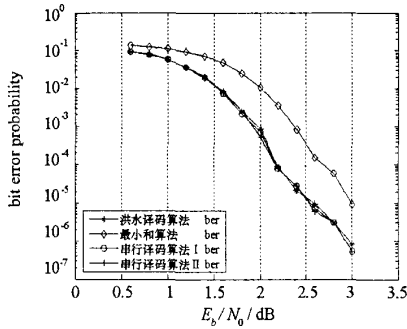


图2 max = 100 时的误比特率

Fig.2 Performance of error rate at max = 100

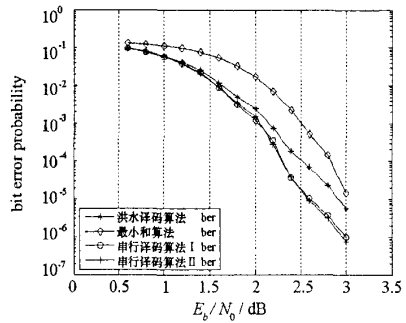


图3 max = 20 时的误比特率

Fig.3 Performance of error rate at max = 20

图4~7分别给出了各种译码算法在不同高斯白噪声方差  $\sigma$  情况下误码性能随迭代次数变化的曲线,可以看到,随着  $\sigma$  的减小,误比特率有了明显的下降。其中两种串行译码算法在初始的几次迭代过程中,误比特率下降的趋势比洪水译码算法和最小和算法都要强烈得多。比如当  $\sigma = 0.7$  时,经过2次迭代后,洪水译码算法、串行译码算法 I 和串行译码算法 II 的误比特率分别约为  $3 \times 10^{-2}$ 、 $6 \times 10^{-3}$ 、 $8 \times 10^{-3}$ ,经过6次迭代后,分别约为  $3 \times 10^{-4}$ 、 $5 \times 10^{-5}$ 、 $3 \times 10^{-6}$ 。可见,在串行消息传递机制下,消息的更新与传递以节点为序顺序进行,使得新的消息能够马上融入迭代过程,加快了平均消息传递速度,经过少数几次迭代便能取得在洪水译码算法下多次迭代才能达到的误码性能。从图中还可以看出,串行译码算法 II 下误比特率下降较快的区间比串行译码算法 I 更大,其译码性能也更好。经过初始几次非

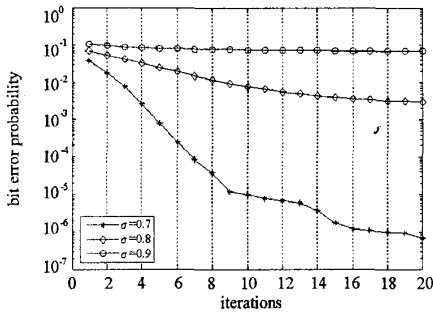


图4 洪水译码算法性能随迭代次数变化曲线

Fig.4 The relation between iterative number and performance of flooding algorithms

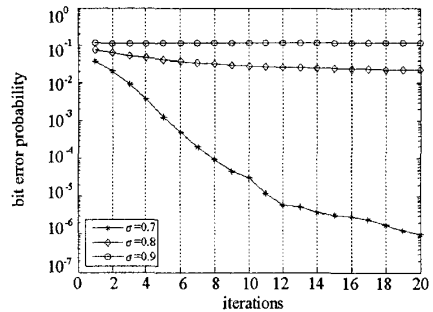


图5 最小和译码算法性能随迭代次数变化曲线

Fig.5 The relation between iterative number and performance of min-sum algorithms

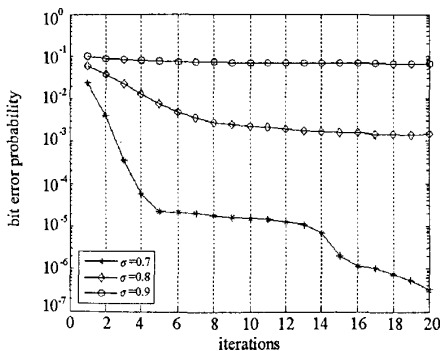


图6 串行译码算法 I 性能随迭代次数变化曲线

Fig.6 The relation between iterative number and performance of serial algorithms- I

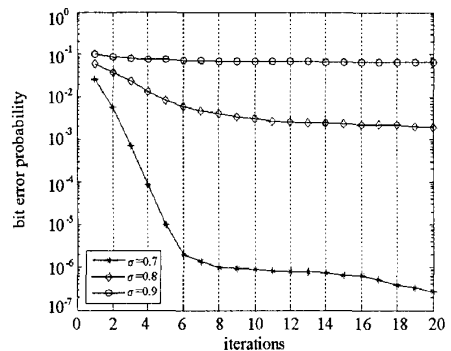


图7 串行译码算法 II 性能随迭代次数变化曲线

Fig.7 The relation between iterative number and performance of serial algorithms- II

常高效的迭代(串行译码算法 I 为 5 次,串行译码算法 II 为 6 次)过程后,两种串行译码算法的误比特率下降趋于平缓。

### 3 复杂性比较

为分析算法效率,对几种算法使用的基本操作次数进行了比较,假设考察对象为  $1/2$  码率的  $(N, J, 2J)$  LDPC 码,表 1 归纳了洪水译码算法、最小和算法、串行译码算法 I 和串行译码算法 II 等几种译码算法的基本运算量,其中  $\phi$  函数用查表法实现。可以看到洪水译码算法与串行译码算法 II 运算量相同,串行译码算法 I 次之,最小和算法运算量最低。

平均迭代次数反映了迭代算法的译码复杂性和时延特性。本文设  $E_b/N_0$  在  $0.5 \sim 3$  dB,分别以最大迭代次数为 100 和 20,考察了几种算法下的平均迭代次数,发现随着信噪比的增加,译码时的迭代次数有下降的趋势。最小和译码算法需要的迭代次数最多,而两种串行译码算法的平均迭代次数都要小于洪水译码算法。当最大迭代次数取 100 时,在整个信噪比区间,最小和译码算法总的平均迭代次数是 50.0603 次,洪水译码算法是 34.7284 次,串行译码算法 I 是 30.6352 次,串行译码算法 II 是 30.7215 次。而当最大迭代次数取 20 时,在整个信噪比区间,最小和译码算法总的平均迭代次数是 14.3037 次,洪水译码算法是 12.3707 次,串行译码算法 I 是 9.5174 次,串行译码算法 II 是 9.7821 次。从概率统计来看,串行译码算法 I 和串行译码算法 II 的平均迭代次数均比洪水译码算法少 4 次左右。

表 1 几种译码算法下一次迭代的运算量比较

Tab.1 Operational quantity in one time iteration of different decoding algorithms

运算步骤	洪水译码算法	最小和算法	串行译码算法 I	串行译码算法 II
$R_w$	$(2J - 2)JN$ 次加法 $2J^2N$ 次查表		$JN$ 次减法 $2JN$ 次查表	$(2J - 2)JN$ 次加法 $2J^2N$ 次查表
$Q_w$	$(J - 1)JN$ 次加法	$(J - 1)JN$ 次加法		$(J - 1)JN$ 次加法
$Q_s$	$JN$ 次加法	$JN$ 次加法	$JN$ 次加法	$JN$ 次加法
$S$			$2JN - N/2$ 次加减法 $JN$ 次查表	

### 4 总结

(1)两种串行译码算法均能带来性能上的提高。尽管最大迭代次数取值较大时洪水译码算法和串行译码算法的误码性能相差不大,但在最大迭代次数较小的情况下,串行译码算法可使译码性能得到明显提高,特别是在中高信噪比区间,性能可提高约  $0.3$  dB 左右。在实际应用中,考虑到硬件资源开销及系统时延,最大迭代次数不可能取得很大,一般都要小于 20 次,此时串行译码算法在性能上的优势便得到充分体现,并且串行译码算法 II 的性能还要略好于串行译码算法 I。

(2)两种串行译码算法的平均迭代次数比洪水译码算法有所减少,改善了消息收敛特性。从消息收敛速度分析,洪水译码算法的每轮迭代都是并行处理消息的过程,所以各个节点对消息的更新是同步的。两种串行译码算法则是顺序进行消息传递,更新的消息马上可以传递给后面的节点,不必等到下次迭代过程,即各个节点对消息的更新是“异步”的,并且越靠后处理的节点所能正确译码的可能性越大。这样平均下来,两种串行译码算法进行一次迭代相当于洪水译码算法多次迭代的结果。因此在迭代译码次数较低时,串行译码算法在性能上的优势更能得到体现。

总之,相对于洪水译码算法,最小和译码算法大幅度降低了译码复杂度,但同时使性能受到较大的损失。串行译码算法 II 在几种译码算法中译码性能最好,其译码复杂度与洪水译码算法相当。因此,本文提出的串行译码算法在不增加译码复杂度的基础上,提高了译码性能,是一种实用价值较高的 LDPC 码译码算法。并且基于变量节点的串行消息传递机制对不等保护能力码的译码将可能带来好处,这也是有待进一步研究的一个课题。

- [3] Watts S, Ward K D. Spatial Correlation in K-distributed Sea Clutter[C]. IEE Proceedings [C], 1987, 134(6):526-532.
- [4] Jao J K. Amplitude Distribution of Composite Terrain Radar Clutter and the K-distribution [J]. IEE Transactions on Antennas and Propagation, 1984, 32(10):1049-1062.
- [5] Jr James M L. Correlated K-distributed Clutter Generation for Radar Detection and Track [J]. IEEE Transactions on Aerospace and Electronic Systems, 1995, 31(2).
- [6] Conte E, Longo M, Lops M. Modeling and Simulation of Non-rayleigh Radar Clutter [C]. IEE Proceedings, 1991, 138(2).
- [7] Szajnowski W J. Simulation Model of Correlated K-distributed Sea Clutter and Thermal Noise [J]. IEEE Trans. Aerosp. Electron Syst., 1987, 23: 568-580.
- [8] Watts S. Radar Detection Prediction in K-distributed Sea Clutter and Thermal Noise [J]. IEEE Trans. Aerosp. Electron Syst., 1987, 23:568-580.
- [9] Rangaswamy M, Weiner D D. Computer Generation of Correlated Non-Gaussian Radar Clutter [J]. IEEE Trans. Aerosp. Electron. Syst., 1995, 31: 106-116.
- [10] Stuart A, ORD J K. Kendall's Advanced Theory of Statistics: Vol 1, Distribution Theory [M]. Edward Arnold, London, 1994: 396-413.
- [11] Gradshteyn I S, Ryzhk L M. Table of Integrals, Series, and Products [M]. Academic Press, San Diego, 1994: 1191.
- [12] Papoulis A. Probability, Random Variables and Stochastic Processes [M]. McGraw-Hill, New York, 1984.
- [13] Li Gang, Yu Kai-bor. Modeling and Simulation of Coherent Weibull Clutter [C]. IEE Proceedings, 1989:136.
- [14] Irina Antipov. Simulation of Sea Clutter Returns [R]. DSTO-TR-0679, 1998.
- [15] 宁海娜, 胡卫东. 低入射余角下雷达海杂波的建模与仿真[J]. 国防科技大学学报, 2000, 22(3):29-33.
- [16] Hawkes C W, Haykin S S. Modeling of Clutter for Coherent Pulsed Radar [J]. IEEE Trans., 1975, 21: 703-707.

---

(上接第 57 页)

## 参考文献:

- [1] Zhang T, Fossorrier M. Shuffled Belief Propagation Decoding[C]. The Proceedings 36<sup>th</sup> Asilomar Conference on Signal Systems and Computers, Pacific Grove, Grove, USA, 2002:8-15.
- [2] Sharon E, Goldberger J. An Efficient Message-passing Schedule for LDPC Decoding[J]. IEEE Commun. Lett., 2004, 4:223-226.
- [3] Mao Y, Banihashemi A H. A New Schedule for Decoding Low-density Parity-check Codes[J]. IEEE, 2001:1007-1010.
- [4] 贺玉成, 杨莉, 等. 置信传播译码算法的性能测度[J]. 电子学报, 2002, 30(4):576-580.
- [5] Ryan W E. An Introduction to LDPC Codes[R]. NSF CCR-9814472, Aug. 2003.
- [6] Kschischang F R, et al. Factor Graphs and Sum Product Algorithm[J]. IEEE Trans. Inform. Theory, 2001, 47(2):498-519.
- [7] 殷剑宏, 吴开亚. 图论及其算法[M]. 合肥: 中国科技大学出版社, 2003.
- [8] 蒋艳凤, 杨学军. 基于搜索编码的简单贝叶斯分类方法[J]. 国防科技大学学报, 2004, 26(5).





