

改进的LDPC译码算法研究

陈旭灿, 刘冬培

(国防科技大学计算机学院 长沙 410073)

【摘要】基于LDPC码的BP译码简化算法, 结合RMP调度和Offset最小和算法, 提出了一种改进的LDPC译码算法。在相同的前提下, 改进的译码算法在计算复杂度方面, 与Offset最小和算法相比, 改善了算法的收敛特性; 采用优化的存储方式, 降低了存储需求, 适合硬件实现。仿真结果表明, 改进的译码算法降低了平均迭代次数, 减少了量化实现占用的存储单元。

关键词 BP算法; LDPC码; 改进的译码算法; Offset最小和算法; RMP

中图分类号 TN911.22

文献标识码 A

doi:10.3969/j.issn.1001-0548.2010.02.014

Modified Decoding Algorithm of LDPC Codes

CHEN Xu-can and LIU Dong-pei

(Computer School, National University of Defense Technology Changsha 410073)

Abstract The simple algorithms based on belief propagation algorithm of LDPC codes are studied. A modified algorithm which integrates Row Message Passing schedule with Offset Min-Sum algorithm for decoding LDPC codes is proposed. The proposed algorithm facilitates hardware implementation because it improves the convergence property while its computation complexity is the same as Offset Min-Sum algorithm, and it reduces the storage for the optimized memory method. The simulation results demonstrate that the proposed algorithm reduces the average iterations and quantization implementation memory.

Key words belief propagation algorithm; LDPC codes; modified decoding algorithm; Offset min-sum algorithm; row message passing

LDPC(low density parity check)译码通用算法即文献[1]最初提出的BP(belief propagation)算法。之后, 文献[2-3]等相继提出了各种基于BP算法的简化算法, 包括最小和算法^[2]、Normalized最小和算法和Offset最小和算法^[3]。

硬件实现时, 最小和算法只需求最小值和加法操作, 避免了BP算法中非线性函数的查找表操作, 极大地降低了计算复杂度, 但也带来了译码性能的损失。为了降低性能损失, Normalized最小和算法和Offset最小和算法两种修正的最小和算法相继被提出。

目前采用的译码算法多是基于SMP(standard message passing)调度的, 每次迭代过程中更新的消息直到下次迭代才能被传出去; 而基于RMP(row message passing)^[4]调度的译码算法能使一次迭代过程中更新的消息立即被传递, 改进了消息的收敛特性^[4-6]。与SMP调度相比, 采用RMP调度的译码算法的迭代次数大约降低了一半。

本文提出一种改进的基于RMP调度的Offset最小和算法, 在译码性能、平均迭代次数和量化实现等方面进行优化, 并基于改进的译码算法, 对DVB-S2(digital video broadcasting-satellite 2)^[7]标准中的LDPC码进行仿真测试和性能比较。

1 LDPC译码算法

LDPC码的校验矩阵 H 可以用Tanner图表示, Tanner图中包含 N 个变量节点和 M 个校验节点。每个变量节点对应校验矩阵的一列, 每个校验节点对应校验矩阵的一行。第 i 个校验节点和第 j 个变量节点存在一条边相连, 当且仅当校验矩阵中 $H_{ij} = 1$ 。假设 $N(m)$ 表示与校验节点 m 相连的所有变量节点集合, $M(n)$ 表示与变量节点 n 相连的所有校验节点集合; $N(m) \setminus n$ 表示集合 $N(m)$ 中去掉变量节点 n , $M(n) \setminus m$ 表示集合 $M(n)$ 中去掉校验节点 m 。

1.1 对数域BP算法

假设在第 k 次迭代过程中, $\lambda_n^{(k)}$ 表示第 n 位输入

数据的后验对数似然比, $\lambda_{mn}^{(k)}$ 表示变量节点 n 向校验节点 m 传递的消息, $A_{mn}^{(k)}$ 表示校验节点 m 向变量节点 n 传递的消息, 对数域BP算法步骤如下。

(1) 初始化。对 $n \in \{1, 2, \dots, N\}$, $\lambda_n^{(0)} = \lambda_n^{(0)}$, $m \in M(n)$, 计算经过信道后接收到的初始对数似然比为:

$$\lambda_n^{(0)} = \log \frac{p(x_i = 0 | y_i)}{p(x_i = 1 | y_i)}$$

(2) 校验节点更新。对每个校验节点 m 和 $n \in N(m)$, 计算:

$$A_{mn}^{(k)} = \left(\prod_{n' \in N(m) \setminus n} \text{sign}(\lambda_{mn'}^{(k-1)}) \right) \Phi^{-1} \left(\sum_{n' \in N(m) \setminus n} \Phi(|\lambda_{mn'}^{(k-1)}|) \right)$$

式中 $\text{sign}(x)$ 和 $\Phi(x)$ 分别定义为:

$$\text{sign}(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

$$\Phi(x) = \Phi^{-1}(x) = \lg \left(\frac{e^x + 1}{e^x - 1} \right) \quad x > 0$$

(3) 变量节点更新。对每个变量节点 n 和 $m \in M(n)$, 计算:

$$\lambda_{mn}^{(k)} = \lambda_n^{(0)} + \sum_{m' \in M(n) \setminus m} A_{m'n}^{(k)}$$

对每个变量节点, 计算后验对数似然比:

$$\lambda_n^{(k)} = \lambda_n^{(0)} + \sum_{m \in M(n)} A_{mn}^{(k)}$$

(4) 译码判决。一次迭代完成后, 进行译码判决。如果 $\lambda_n^{(k)} \geq 0$, $\hat{x}_n^{(k)} = 0$; 否则 $\hat{x}_n^{(k)} = 1$ 。由此可以得到关于译码码字的一个估计值 $\hat{x}^{(k)}$, 再计算伴随式 $s = H(\hat{x}^{(k)})^T$, 如果 $s = 0$, 那么译码成功, 结束译码, 并将 $\hat{x}^{(k)}$ 作为有效输出值; 否则转步骤(2)继续迭代, 直至达到预定的最大迭代次数。

1.2 基于BP算法的简化算法

BP算法性能优异, 但在校验节点的更新步骤中存在非线性函数 $\Phi(x)$ 的计算, 硬件实现时, 通常用查找表的方法实现 $\Phi(x)$, 对 $\Phi(x)$ 的量化实现直接影响查找表的精度和复杂度, 从而影响译码性能和硬件复杂度。最小和算法^[2]是基于BP算法的简化算法, 它将校验节点的更新步骤简化为:

$$A_{mn}^{(k)} = \prod_{n' \in N(m) \setminus n} \text{sign}(\lambda_{mn'}^{(k-1)}) \left(\min_{n' \in N(m) \setminus n} |\lambda_{mn'}^{(k-1)}| \right)$$

最小和算法避免了非线性函数 $\Phi(x)$ 的计算, 只需求最小值和加法操作, 大大降低了计算复杂度; 但也带来了译码性能的损失, 尤其在低码率下性能损失较大。为了提高最小和算法的性能, 提出了Normalized最小和以及Offset最小和算法^[3], 它们都是在最小和算法的基础上加以修正。

在Normalized最小和算法中, 对校验节点的更新在最小和算法的基础修正为:

$$A_{mn}^{(k)} = \alpha A_{mn}^{(k)} \quad 0 < \alpha < 1$$

在Offset最小和算法中, 对校验节点的更新在最小和算法的基础上修正为:

$$A_{mn}^{(k)} = \max(A_{mn}^{(k)} - \beta, 0) \quad \beta > 0$$

在上面两种修正的最小和算法中, 最优的 α 、 β 的值可以根据密度演化理论或计算机仿真得到。相对于Normalized最小和算法, Offset最小和算法更便于硬件实现。

2 改进的译码算法

文献[4]中比较了两种不同的消息传递机制SMP和RMP, 并指出RMP译码算法与SMP译码算法具有相同的计算复杂度, 但RMP译码算法收敛速度更快, 减少了迭代次数。对数域BP算法和基于BP算法的简化算法都是基于SMP调度的, 在SMP译码算法的每次迭代过程中, 变量节点和校验节点均同时处理和传递消息, 各个节点对消息的更新是同步的; 而在RMP译码算法中, 与校验节点相连的变量节点的更新可以使用该次迭代中已经更新过的消息, 从而改进消息的收敛特性, 所以RMP译码算法一次迭代的结果相当于SMP算法多次迭代的结果^[5]。

影响LDPC译码复杂度的主要因素有迭代次数和每次迭代过程中的计算复杂度。基于RMP的译码算法可以有效地降低平均迭代次数, 修正的最小和译码算法可以在计算复杂度和译码性能之间进行折衷。本文综合二者的特点, 提出一种改进的基于RMP调度的Offset最小和算法, 改进的译码算法步骤如下。

(1) 初始化

$$A_{mn}^{(0)} = 0, \quad \lambda_n = \lambda_n^{(0)}$$

(2) 迭代过程

for $m = 1, 2, \dots, M$

for $n \in N(m)$

$$A_{mn}^{(k)} = \prod_{n' \in N(m) \setminus n} \text{sign}(\lambda_{mn'} - A_{mn'}^{(k-1)}) \max \left(\min_{n' \in N(m) \setminus n} |\lambda_{mn'} - A_{mn'}^{(k-1)}| - \beta, 0 \right)$$

end

for $n \in N(m)$

$$\lambda_n = \lambda_n - A_{mn}^{(k-1)} + A_{mn}^{(k)}$$

end

end

(3) 译码判决

当 $\lambda_n \geq 0$ 时, $\hat{x}_n^{(k)} = 0$, 否则 $\hat{x}_n^{(k)} = 1$ 。

改进的译码算法的每次迭代过程包含两个循环, 外循环依次计算每个校验节点, 内循环依次更新与校验节点相连的变量节点。因此, 基于RMP消息传递方式的译码算法, 也被称为串行译码算法^[5]或分层译码算法^[6]。

3 算法仿真与分析

为了验证改进的译码算法的性能, 采用BPSK调制、AWGN信道; 选用DVB-S2标准^[7]中码长64 800、码率1/2的LDPC码进行性能仿真。假设所有信噪比下及每次迭代过程中 $\beta = 0.5$ 。

3.1 算法性能比较

BP算法、最小和算法、Offset最小和算法、改进的译码算法在最大迭代次数为30和15时的误码性能曲线分别如图1和图2所示。BP算法、最小和算法、Offset最小和算法3种算法都是基于SMP调度的; 改进的译码算法是基于RMP调度的Offset最小和算法。

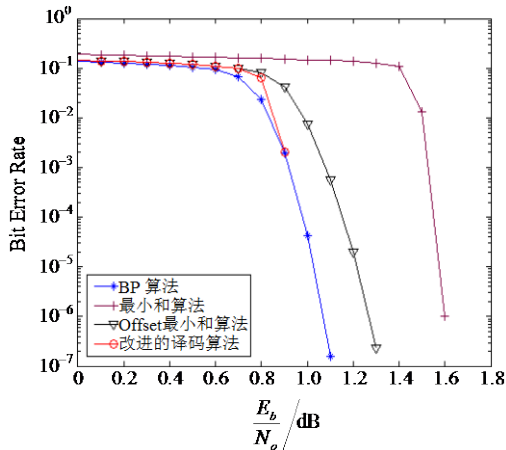


图1 不同算法在最大迭代次数为30时的误码性能

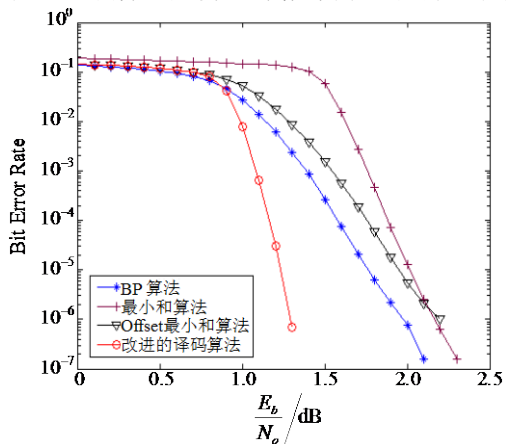


图2 不同算法在最大迭代次数为15时的误码性能

由图1和图2可以看出, 低信噪比条件下, 最小和算法存在误码平台较高的问题。在最大迭代次数为30时, Offset最小和算法的性能比最小和算法有较

大提高, 在误码率为 10^{-6} 时与BP算法相差约0.2 dB。改进的译码算法由于改变了消息传递的方式, 改进了算法的收敛特性, 能在低信噪比和低迭代次数的情况下达到更低的误码率。在最大迭代次数为15、误码率为 10^{-6} 时, 改进的译码算法相对BP算法的编码增益为0.6 dB。

3.2 平均迭代次数

设定算法的最大迭代次数为50, BP算法、最小和算法、Offset最小和算法、改进的译码算法4种不同算法的平均迭代次数曲线如图3所示。

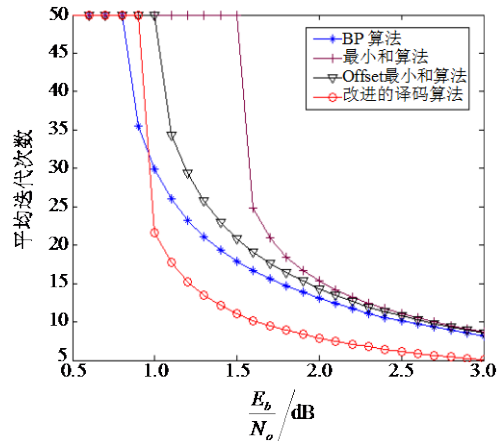


图3 不同算法的平均迭代次数

由图3可以看出, 当信噪比 $E_b/N_o \geq 1.0$ dB时, 改进的译码算法的平均迭代次数明显低于另外3种算法, 这是因为改进的译码算法改进了消息传递方式, 改善了算法的收敛特性。因此, 改进的译码算法能以较低的迭代次数满足性能要求, 降低了译码延迟。

3.3 量化实现与存储分析

实际应用中, 必须考虑LDPC译码硬件实现所涉及的量化与存储问题。在改进的译码算法中, 注意到中间结果 $\min_{n' \in N(m) \setminus n} |\lambda_n - A_{mn}^{(k-1)}|$ 只存在最小值和次小值两个值^[8], 故只需存储最小值的绝对值、次小值的绝对值和最小值所对应的下标, 并存储 $(\lambda_n - A_{mn}^{(k-1)})$ 的符号位, 简化了中间结果的存储; 此外, λ_n 可以与信道初始值 $\lambda_n^{(0)}$ 共用同一存储单元, 降低了存储需求。

本文针对改进的译码算法采用上述存储方式, 提出了一种有效的量化方案。对 λ_n 采用8位(6位整数、2位小数)均匀量化(如果采用6位量化, 频繁的数据溢出导致不能成功译码); $A_{mn}^{(k)}$ 采用6位(4位整数、2位小数)均匀量化, 最小值与次小值的绝对值则用5位(3位整数、2位小数)存储。该量化方案的仿真结果如图4所示。由图4可以看出, 在最大迭代次数为15、误码率为 10^{-6} 时, 改进的译码算法量化后的性能仅损

失约0.1 dB。

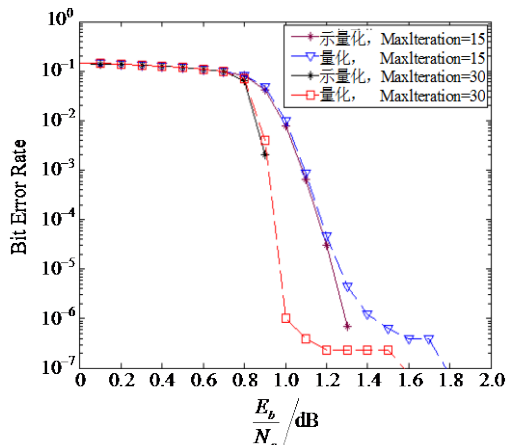


图4 改进的译码算法量化后的误码性能

假设LDPC码每个校验节点的度为 d_c ,校验矩阵为 $M \times N$ 阶的矩阵。对于改进的译码算法,采用上述量化方案,共需存储单元为:

$$T_1 = M \times (5 + 5 + \lceil \log_2 d_c \rceil + d_c) + N \times 8$$

对于BP算法,假设对接收到的对数似然比和中间结果均采用6位量化^[9,10],共需存储单元为:

$$T_2 = M \times (6 \times d_c) + N \times 6$$

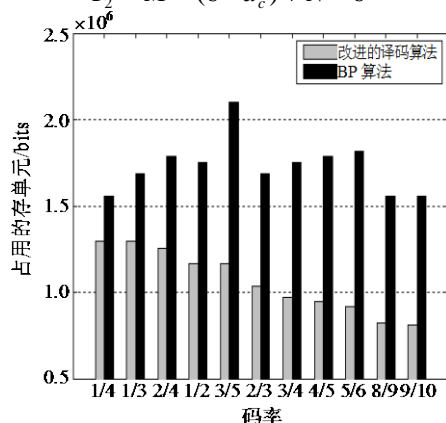


图5 两种算法在不同码率下所占用的存储单元

图5比较了DVB-S2标准中LDPC码在普通帧(码长为64 800)的不同码率下,改进的译码算法与BP算法量化后所占用的存储单元。由图5可以看出,在所有码率下,改进的译码算法所占用的存储单元比BP算法都要小,这是因为改进的译码算法采用了优化的存储方式,且不需要存储接收到的对数似然比。

4 结论

基于RMP调度方式的译码算法改进了消息传递

方式,加快了算法的收敛速度,Offset最小和算法在译码性能和计算复杂度之间达到了较好的折衷。本文结合二者的特点,提出一种改进的LDPC译码算法,能在相同的迭代次数下比BP算法达到更低的误码率,量化实现所占用的存储单元比BP算法小;在相同的前提下,改进的译码算法与Offset最小和算法相比,减少了平均迭代次数,采用优化的存储方式,降低了存储需求,适合硬件实现。

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] FOSSORIER M, MIHALJEVIC M, IMAI H. Reduced complexity iterative decoding of low density parity check codes based on belief propagation[J]. IEEE Trans on Comm, 1999, 47(5): 673-680.
- [3] CHEN J, FOSSORIER M. Density evolution for two improved BP-based decoding algorithms of LDPC codes[J]. IEEE on Commun Letters, 2002, 6(5): 208-210.
- [4] RADOSAVLJEVIC P, BAYNAST A, CAVALLARO J R. Optimized message passing schedule for LDPC decoding[C]//Signals, Systems and Computers, 2005, Conference Record of the Thirty-Ninth Asilomar Conference on. [S.l.]: [s.n.], 2005: 591-595.
- [5] LEI Jing, WEN Lei, FU Qiang. Analysis of LDPC decoding algorithm based on serial schedule[J]. Journal of Sichuan University (Natural Science Edition), 2006, 43(4):790-795.
- [6] HOCEVAR D E. A reduced complexity decoder architecture via layered decoding of LDPC codes[C]//IEEE Workshop on Signal Processing Systems, SIPS: Design and Implementation. Texas: [s.n.], 2004: 107-112.
- [7] European Telecommunications Standards Institute. ETSI. EN 302 307 V1.1.1[S]. 2004.
- [8] HUANG Xiao-fei. Single-scan min-sum algorithms for fast decoding of LDPC codes[C]//2006 IEEE Information Theory Workshop. Chengdu: [s.n.], 2006: 140-143.
- [9] SUN Shao-hui, SUN Rong, WANG Xin-mei. Some quantization issues for decoding of low-density parity check codes with BP Algorithm[J]. Acta Electronica Sinica, 2003, 31(2): 217-220.
- [10] YU Li, ZHANG Xiao-lin, ZHAO Ling. Quantization issues for the decoding of irregular LDPC codes[J]. Journal of Telemetry, Tracking, and Command, 2007, 28(3): 53-58.

编辑 漆蓉