

基于串行消息传递机制的 QC-LDPC 码快速译码算法研究

雷菁^① 高永强^① 王建辉^{①②} 贺文辉^③

^①(国防科技大学电子科学与工程学院 长沙 410073)

^②(中国人民解放军 66018 部队 北京 100041)

^③(西安通信学院 西安 710106)

摘要: 针对准循环 LDPC(QC-LDPC)码基于洪水消息传递机制译码算法的不足, 该文提出了一种快速的分组串行译码算法。该算法通过将 LDPC 码的校验节点(或变量节点)按一定规则划分成若干个子集, 在每一轮迭代过程中, 依次对各个子集中的校验节点(或变量节点)并行地进行消息更新, 提高了译码速度。同时根据分组规则, 提出了一种有效的分组方法, 并通过分析发现基于循环置换阵的准循环 LDPC 码非常适合采用这种分组译码算法进行译码。通过对不同消息传递机制下准循环 LDPC 码译码算法性能的仿真比较, 验证了在复杂度不增加的情况下, 该译码算法在继承了串行译码算法性能优异和迭代收敛快等优点的同时, 极大地提高了准循环 LDPC 码的译码速度。分析表明, 分组串行译码算法译码速度至少为串行译码算法的 p 倍(p 为准循环 LDPC 码校验矩阵中循环置换阵的行数或列数)。

关键词: 准循环 LDPC 码; 译码算法; 串行消息传递机制; 分组串行算法

中图分类号: TN911.22

文献标识码: A

文章编号: 1009-5896(2008)12-2938-05

Research on a Fast Decoding Method of the Quasi-cyclic LDPC Codes Based on Serial Message-Passing Schedule

Lei Jing^① Gao Yong-qiang^① Wang Jian-hui^{①②} He Wen-hui^③

^①(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

^②(Unit 66018 of the PLA, Beijing 100041, China)

^③(Xian Communication Institution, Xian 710106, China)

Abstract: A new fast decoding method of Quasi-Cyclic LDPC(QC-LDPC)codes, named semi-serial decoding algorithm, is presented through contraposing the shortcoming of flood decoding method. For the algorithm, the message was updated in a serial sequence of m subsets of check nodes(or variable nodes), while the check nodes(or variable nodes) in the same subset are updated simultaneously. In this way, the decoding speed would be faster than the common serial decoding algorithm several times. An efficient grouping method suitable for the quasi-cyclic LDPC codes based on the circulant architecture is proposed. Simulation results show that, under the same complexity, the proposed algorithm fastens the decoding speed of quasi-cyclic LDPC codes, meanwhile, it inherits all the merits of the serial message-passing decoding algorithm. The conclusion is that the speed of the semi-serial decoding method is at least p times than that of the serial one(p is the number of rows of the cyclic permutation matrix in the parity check matrix of quasi-cyclic LDPC codes).

Key words: QC-LDPC codes; Decoding methods; Serial message-passing schedule; Semi-serial algorithm

1 引言

洪水译码算法是目前普遍采用的 LDPC 译码算法^[1-3], 是一种基于并行处理的消息传递机制, 尽管理论上该算法由于采用全并行处理, 译码速度很快, 但在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。另一方面, 其消息传递的收敛特性也不理想, 当期望的性能较高时, 正确译码所需的迭代次数较大, 且每一次迭代中包含

大量实数运算, 这将使算法整体译码速度大打折扣。鉴于洪水消息传递机制的不足, 许多研究人员对非洪水译码机制进行了研究, 如 Zhang 等提出置乱(shuffled)的 BP 译码算法, 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^[4,5]; Sharon 等人提出了基于校验节点(或变量节点)的串行方案的消息传递机制, 相比洪水消息传递机制, 其具有消息传递收敛快, 译码复杂度低, 实际应用中占用硬件资源少等优点^[6]。但由于串行消息传递机制在每次迭代过程中, 消息的更新是按节点依次进行的, 虽然所需的迭代总次数减少, 但是每轮迭代所

需时间增长。为进一步提高译码速度, Sharon 等人又提出了一种分组串行译码的设想, 但没有给出具体的实现方法。本文对该想法进行了具体的分析与设计, 给出了一种 LDPC 码校验节点的分组方法, 并针对基于循环置换阵的准循环 LDPC 码校验矩阵的结构特点, 提出了一种准循环 LDPC 码的快速译码算法, 该算法在继承串行消息传递译码优点的同时, 能够较大幅度地提高 LDPC 迭代译码的速度。

本文第 2 节对分组串行消息传递机制进行了简要叙述; 第 3 节详细描述了分组串行译码算法的分组规则, 并提出了一种适用于任意类型 LDPC 码的校验节点分组的方法; 第 4 节提出了一种准循环 LDPC 码快速串行译码算法; 第 5 节给出了几种基于不同消息传递机制的准循环 LDPC 码译码算法的性能仿真结果; 最后得出结论, 分组串行译码算法在基本保持串行译码算法性能前提下, 译码速度至少为串行译码算法的 p 倍(p 为准循环 LDPC 码校验矩阵中循环置换阵的行数或列数)。

2 分组串行消息传递机制

串行和分组串行译码算法都可分为基于校验节点和基于变量节点两种, 本文只对基于校验节点的译码算法分别进行简要叙述, 基于变量节点的译码算法可做类似处理。

基于校验节点的串行译码算法过程是一种迭代 SISO 译码算法在每一轮迭代过程中, 对校验节点按一定顺序进行消息处理和传递, 对每个校验节点同时接收变量消息 $L(Q_{vc})$ 并向与之相连的消息节点发送校验消息 $L(R_{cv})$ 。与洪水译码算法相比, 串行译码算法具有收敛快、性能优异、节省硬件资源等优点^[7,8], 但由于在每轮迭代过程中, 串行译码算法对节点消息是顺序更新的, 这样每次译码所需时间就会增长, 导致译码速度的降低, 而分组串行译码可以较好地解决这一问题。

基于校验节点的分组串行译码算法是在串行译码算法中, 加入校验节点集合分组处理, 以实现局部并行操作。即先将校验节点集合 C 按一定规则划分成多个子集, 消息的更新以子集为顺序依次进行(即按串行方式完成消息传递), 而位于同一子集内的校验节点同时进行消息的更新(即并行操作)。因此, 该算法在继承了串行译码算法性能优异和迭代收敛快等各种优点的同时, 可以提高 LDPC 串行迭代译码的译码速度, 特别是各分组内包含的校验节点多时, 提高的效果更佳。在这种基于校验节点的分组串行译码算法中一个关键的问题是校验节点子集的划分, 其分组方法的不同将直接影响算法的译码性能^[7]。

3 分组串行译码中分组方法研究

3.1 校验节点分组规则

分析采用基于校验节点的串行译码算法流程可知, 对某校验节点 c 的校验消息 $L(R_{cv})$ 进行更新之后, 与之相连的变

量节点的后验概率 $L(Q_v)$ 也要进行更新, 此时 $L(Q_v)$ 的计算只用到了 c 这一个校验节点的信息。那么在采用分组串行译码算法译码时, 就要求在任意一个校验节点分组内避免两个或两个以上校验节点与相同的变量节点 v 相连。若在某个分组内存在这种情况, 则在该分组内部所有校验节点消息更新完毕后, 对变量节点 v 进行后验概率 $L(Q_v)$ 更新时, 由于这时采用的是串行译码算法, 将只取与 v 相连的校验节点中的一个校验信息值进行计算, 这样就造成了信息的遗漏, 对译码性能带来不利的影响。因此校验节点分组规则可表述为

$$\forall i \in \{1, \dots, m\}, \forall c, c' \in B_i, N(c) \cap N(c') = \emptyset \quad (1)$$

3.2 校验节点分组方法

在满足分组规则基础上, 为了提高算法运行的并行性, 以加快译码算法速度, 就要求同一分组内包含更多与不同变量节点相连的校验节点。据此本文提出了一种校验节点分组方法, 其具体实现步骤如下:

第 1 步 将变量节点按其度值由大到小的顺序重新排列, 重排之后的变量节点依次记为 v'_1, v'_2, \dots, v'_N , 它们的度分别记为 d_1, d_2, \dots, d_N ($d_1 \geq d_2 \geq \dots \geq d_N$);

第 2 步 取度最大的变量节点 v'_1 , 对其连接的各校验节点分组, 每个节点占一个分组, 分别记为 B_1, B_2, \dots, B_{d_1} , 分组集合 $B = \{B_1, B_2, \dots, B_{d_1}\}$, 如图 1(a)所示;

第 3 步 对与变量节点 v'_2 连接的各校验节点分组, 其中与前一节点 v'_1 也有连接关系的校验节点(假设这样的节点有 z 个, $z \geq 0$), 不对其所在分组进行更新, 其余 $(d_2 - z)$ 个校验节点分别“均匀地”放入 B 中剩余分组, 图 1(b)和 1(c)分别表示的是 $z = 1$ 和 0 的情况。

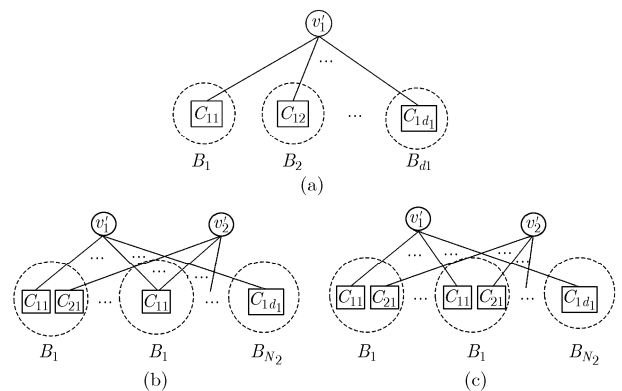


图 1 校验节点分组方法示意图

第 $i+1$ 步, 对第 i 个变量节点 v'_i 相连的各校验节点进行分组, 首先扫描与前面已经处理过的变量节点也相连的校验节点所在的分组: 若发现在同一分组中有多个校验节点(假设有 $t \geq 2$ 个)与 v'_i 相连, 则保留一个节点在原分组中, 而将其余的 $(t-1)$ 个校验节点取出, 重新建立 $t-1$ 个分组, 每个校验节点分别置于不同的分组, 此时集合 B 中分组数增加了 $t-1$ 个; 否则(即 $t = 1$ 时)保持这些校验节点的已有分组位置, 此

时 B 中分组数不变。然后将剩余的校验节点“均匀地”放入分组集合 B 中其它没有包含与 v_i' 相连的校验节点所在分组中。

按照同样的方法依次对各变量节点进行处理，直至所有变量节点全部处理完毕。

需要说明的是，上文中“均匀地”分配校验节点是为了尽量保证各个分组内校验节点数目相同，以便于译码器的硬件实现。因此在分组时应优先向节点数少的分组分配节点。

以上分组方法的核心在于：将与变量节点相连的各校验节点分别置于不同的分组内，同时使分组数目尽可能地小(即分组中元素尽量多)。根据 Tanner 图与校验矩阵 H 的对应关系不难看出，此方法中对校验节点的分组就相当于对 H 矩阵的行进行分组，即通过对 H 矩阵进行行变换，使矩阵每一列的非零元素处在不同的行分组内部，为了方便表述，将该对应关系称为满足分组串行译码算法的校验矩阵行分组规则。

4 QC-LDPC 码快速译码算法

由文献[9-12]可知，基于循环置换阵的准循环 LDPC 码具有形如式(2)结构校验矩阵，其中， $I(S_{i,j})$ 为 $p \times p$ 维的循环置换阵， $S_{i,j}$ 为 $I(S_{i,j})$ 相对于单位矩阵循环右移的次数，其中 $S_{i,j}$ 为 -1 时 $I(S_{i,j})$ 表示全零矩阵，为 0 时 $I(S_{i,j})$ 表示单位阵。

$$H = \begin{bmatrix} I(S_{1,1}) & I(S_{1,2}) & \cdots & I(S_{1,k}) \\ I(S_{2,1}) & I(S_{2,2}) & \cdots & I(S_{2,k}) \\ \vdots & \vdots & & \vdots \\ I(S_{m,1}) & I(S_{m,2}) & \cdots & I(S_{m,k}) \end{bmatrix} \quad (2)$$

通过分析可知，具有式(2)形式的准循环 LDPC 码校验矩阵 H' 每列的非零元素位于不同的循环置换阵内部，因此若将 H' 矩阵每个循环置换阵 $I(S_{i,1})(1 \leq i \leq m)$ 所在行作为一个分组(例第一个 p 行 $I(S_{1,1}), I(S_{1,2}), \dots, I(S_{1,k})$ 作为一个分组，……，第 m 个 p 行 $I(S_{m,1}), I(S_{m,2}), \dots, I(S_{m,k})$ 也作为一个分组)，则该矩阵包含 m 个校验节点分组 $\{B_1, B_2, \dots, B_m\}$ ，每个分组内不存在多个校验节点与同一变量节点相连的情况——符合分组串行译码算法的校验矩阵行分组规则，可以采用分组译码算法译码对其译码。每轮迭代过程中依次对 m 个分组内的 p 个校验节点并行地进行消息接收和处理，那么基于校验节点的分组串行译码算法速度为串行译码算法的 p 倍。

为了提高译码速度，结合本文 3.2 节提出的分组方法，对 QC-LDPC 码校验节点重新分组，以期进一步增大各校验节点分组的容量，得到一种更快速的改进算法，具体做法为

第 1 步 将 H' 中的循环置换阵 $I(S_{i,j})$ 分别用单符号‘0’或‘1’替代，如果 $I(S_{i,j})$ 为全零矩阵，则用‘0’替代，否则用‘1’替代，替代后的矩阵为校验矩阵 H' 的基矩阵；

第 2 步 按照 2.2 节提出的分组方法，对 H' 的基矩阵按

行进行置换与分组，即调整基矩阵行排列次序，使其每列非零元素位于不同行分组中，则该矩阵包含 m 个行分组 $\{h_1, h_2, \dots, h_m\}$ ；

第 3 步 将置换分组后的基矩阵进行还原扩展，即将‘0’或‘1’符号重新用原来的循环置换阵反替换，得到一个仍然具有式(2)形式的校验矩阵 H'' 。此时由于‘0’或‘1’符号分别被扩展为一个 $p \times p$ 维的循环置换阵，因此基矩阵中行分组 $h_i(i = 1, \dots, m)$ 的每行(称为基矩阵行分组 h_i 中的一个元素)会扩展成 p 行，且均满足分组串行译码算法的校验矩阵行分组规则，即经过前 3 步处理后，每个校验节点分组容量大增。

第 4 步 按照 H'' 矩阵对应的 Tanner 图，采用分组串行传递机制进行译码。具体译码过程如图 2 所示。每一轮迭代过程中，首先对子集 B_1 中的所有校验节点并行地进行变量消息 $L(Q_{vc})$ 的接收和校验消息 $L(R_{cv})$ 的发送(即图 2 虚框所示)，然后对子集 B_2 中的所有校验节点并行地接收变量消息 $L(Q_{vc})$ 和发送校验消息 $L(R_{cv})$ ，……，依次对各个子集内部校验节点并行地接收并处理消息，直至一轮迭代结束(即图 2 实线框所示)。一轮迭代之后，需计算每个变量节点后验概率的对数似然比 $L(Q_v)$ ，并做出判决：若 $L(Q_v) < 0$ ，则 $v_i = 1$ ，否则 $v_i = 0$ 。由此可以得到对发送码字的估计 $v = (v_1, v_2, \dots, v_N)$ 。再计算伴随式 $S = vH^T$ ，若 $S = 0$ ，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。

由于算法过程中前 3 步的置换过程中仅仅是交换 H' 的行次序，故 H'' 与 H' 具有同样的码空间，不改变原码的约束关系，因此第 4 步的译码结果与直接对 H' 的 QC-LDPC 码译码结果是一样的。

可以看出，若利用 QC-LDPC 码校验矩阵特点直接对 H' 进行分组(即按循环置换阵所在行分组)，那么分组内的节点数目均为 p ；若采用改进的算法，则分组内校验节点数目将成 p 的整数倍增长，分组效率较直接分组法将成倍提高(设对基矩阵分组后的行分组 h_i 中最少含 t 个元素，则改进算法的分组效率是直接法的 t 倍)，相应地，其译码速度也将极大

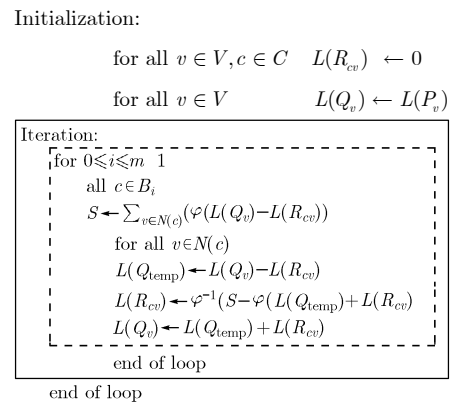


图 2 基于校验节点的分组串行译码算法基本流程

提高。

采用改进的分组串行译码算法对具有式(2)形式的校验矩阵的 QC-LDPC 码译码时,每轮迭代过程中依次对 m 个分组内的 p 的整数倍个校验节点并行地进行消息接收和处理,因此其译码速度至少为串行译码算法的 p 倍,是一种快速的串行译码算法。因此, QC-LDPC 码是一类非常适合分组串行译码的码型,只要恰当提高循环置换阵的维数 p ,就可有效提高其译码速度。而采用本文改进分组的译码算法将进一步提高其速度。

同时,由图 2 可以看出,对于 QC-LDPC 码来说,采用分组串行译码来进行译码器设计时,译码过程中包含的各种运算和洪水译码算法一样,硬件实现所需的元器件基本一致,但由于译码过程中不用存储 $N(v)$ 和 $L(Q_{vc})$ 等中间变量,因而与洪水译码算法相比,可以节省大量的存储空间,这就使得硬件的实现变得相对简单。

5 仿真结果及分析

为了验证 QC-LDPC 码下分组串行译码算法性能,分别将其与洪水译码算法、归一化最小和译码算法(NMS)、基于校验节点的串行译码算法进行了比较仿真实验。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,调制方式为 BPSK 调制,采用类似文献[10]的方法构造码率为 1/2,码长为 1008,围长为 8 的准循环 LDPC 码,内部小循环置换阵大小为 12×12 。译码的最大迭代次数为 50, NMS 算法中的归一化参数取 0.8。为简单起见,假设每次传输的都是全零码字[12]。

由图 3 仿真结果可以看出,在最大迭代次数为 50 次,误比特率为 10^{-5} 时,4 种译码算法性能差距不是很大,差距最大的两种译码算法(基于校验节点的串行译码算法和洪水译码算法)也仅相差 0.3dB 左右;在高信噪比(2.5~3.0dB)下,基于校验节点的串行译码算法性能是这 4 种译码算法中性能最好的,分组串行译码算法性能与其十分接近,这是由于串行和分组串行译码算法消息的更新以节点的顺序进行,使得更新的消息马上融入到本次迭代,加快了消息的传递速度,从而使得这两种译码算法经过少数的几次迭代就能取得洪水译码算法多次迭代才能达到的性能,因此在最大迭代次数较小的情况下,它们的性能将比洪水译码算法好。在高信噪比(2.5~3.0dB)区间, NMS 算法性能优于洪水译码算法,这是因为采用 NMS 算法译码在对校验节点消息更新时,只用到了与该节点相连的一个变量节点的消息,在一定程度上消除了环的影响。由以上分析可知,分组串行译码算法继承了串行消息传递算法性能优异的优点。

平均迭代次数反映了迭代算法的译码复杂性和时延特性。本文对基于校验节点的串行译码算法、分组串行译码算法以及洪水译码算法收敛所需的平均迭代收敛次数作出了比较仿真,仿真时设最大迭代次数为 50 次,仿真结果如图 4。随着信噪比的增加,各种译码算法收敛所需迭代次数有下降

的趋势,并且从概率统计来看,串行译码算法和分组串行译码算法收敛所需的平均迭代次数比洪水译码算法都少 4 次左右,并且逐步接近洪水译码算法收敛所需平均迭代次数的 1/2,这和文献[8]中的结论一致。这是因为这两种算法顺序进行消息传递,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,这样平均下来这两种算法进行一次迭代相当于洪水译码算法多次迭代的结果,收敛所需的平均迭代次数相应减少。同时由图可知,串行译码算法和分组串行译码算法收敛所需平均迭代次数基本相同,这是因为本文使用分组串行译码算法时,每个校验节点分组内部的任意两个校验节点没有共同的变量节点与它们相连,可并行地对分组内的所有校验节点进行消息更新,更新的消息内容与按分组后校验结点的顺序对其进行消息更新的串行译码算法相比并没有改变。由此分析可知,分组串行译码算法继承了串行译码算法的收敛快的优点。同时,在采用分组串行译码算法对该准循 LDPC 码译码时,每轮迭代过程中依次对 42 个分组内的 12 的整数倍个校验节点并行地进行消息接收和处理,因此基于校验节点的分组串行译码算法的译码速度至少为基于校验节点的串行译码算法速度的 12 倍。对于硬件实现而言,这种提高是可观的。

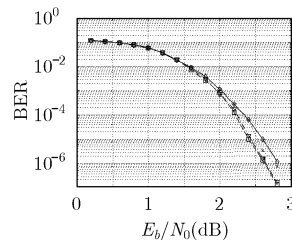


图 3 基于不同消息传递机制的准循环 LDPC 码译码算法性能仿真图

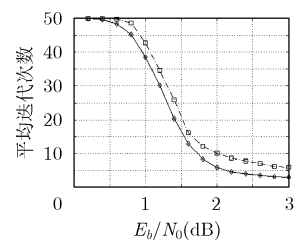


图 4 几种算法收敛性比较仿真图

6 结束语

本文通过对分组串行译码方法的研究,提出了一种适用于任意类型 LDPC 码的校验节点分组的方法,使分组后的 LDPC 码能采用分组串行译码算法进行译码。同时利用分组串行译码算法的思想,针对准循环 LDPC 码校验矩阵的结构特点,提出了一种准循环 LDPC 码的快速译码算法,该译码算法在运算复杂度上和洪水译码算法、串行译码算法相当,但其译码性能在高信噪比情况下和串行消息算法相近,比洪水译码算法要好;其译码速度至少为串行译码算法的 p 倍(p 为准循环 LDPC 码校验矩阵中循环置换阵的行数或列数);其收敛所需的迭代次数也和串行译码算法相近。因此,本文提出的分组串行算法相对洪水译码算法来说,在不增加复杂

度的情况下,提高了译码的性能;而相对串行译码算法来说,极大提高了译码的速度,是一种实用价值较高的准循环LDPC码译码算法。

参 考 文 献

- [1] Radosavljevic P, de Baynast A, and Cavallaro J R. Optimized message passing schedules for LDPC decoding. Conference record of the 39th asilomar conference on signals, systems and computers, California, 2005: 591-595.
- [2] Mao Y and Banilashemi A H. A new schedule for decoding low-density parity-check codes. *Global Telecommunications Conference, USA*, 2001, 47(2): 1007-1010.
- [3] Kschischang F R, Frey B J, and Loeliger H A. Factor graphs and the sum-product algorithm. *IEEE Trans. on Info. Theory*, 2001, 47(2): 498-519.
- [4] Zhang T and Fossorrier M. Shuffled belief propagation decoding. The Proceedings 36th Asilomar Conference on Signal Systems and Computers, Pacific Grove, Grove, USA, 2002: 8-15.
- [5] Zhao Chuan-gang, Yuan Jin-sheng, Lin Xue-hong, and Lin Jia-ru. Improvement of shuffled iterative decoding. Proceedings of 2006 IEEE Information Theory Workshop, Uruguay, 2006: 114-116.
- [6] Sharon E, Litsyn S, and Goldberger J. An efficient message-passing schedule for LDPC decoding. Proc. 23rd IEEE Conv., Israel, 2004: 223-226.
- [7] Sharon E, Litsyn S, and Goldberger J. Convergence analysis of serial message-passing schedules for LDPC decoding. The 4th International Symposium on Turbo Codes, Munich, 2006: 110-116.
- [8] 雷菁, 文磊, 唐朝京. 基于变量结点串行消息传递的LDPC码译码研究. *国防科技大学学报*, 2006, 28(5): 52-57.
- [9] Fossorrier M P C. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. *IEEE Trans. on Info. Theory*, 2004, 50(8): 1788-1793.
- [10] Xu Jun, Chen Lei, Lan Lan, and Lin Shu. Construction of low-density parity-check codes by superposition. *IEEE Trans. on Inf. Theory*, 2005, 53(2): 243-251.
- [11] Li Zongwang and Kumar B V K V. A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph. Conference record of the 38th asilomar conference on signals, systems and computers, California, 2004: 1990-1994.
- [12] Richardson T and Urbanke R L. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. on Inf. Theory*, 2001, 47(1): 599-618.
- 雷菁: 女, 1968年生, 教授, 硕士生导师, 主要研究方向为通信传输与编码技术。
- 高永强: 男, 1983年生, 硕士生, 研究方向为信道编码。
- 王建辉: 男, 1979年生, 硕士生, 研究方向为信道编码。
- 贺文辉: 女, 1968年生, 讲师, 研究方向为计算机应用与计算机通信。