

LDPC Code's Girth Calculation Algorithm

Huanming Zhang, Bin Zhang

Electron Information Engineering Department, Foshan Institute of Science and Technology, Foshan
Email: 511106085@qq.com

Received: Dec. 10th, 2013; revised: Dec. 13th, 2013; accepted: Dec. 16th, 2013

Copyright © 2014 Huanming Zhang, Bin Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for Hans and the owner of the intellectual property Huanming Zhang, Bin Zhang. All Copyright © 2014 are guarded by law and by Hans as a guardian.

Abstract: Low-Density Parity-Check codes have advantageous performances discovered in 1996 with Shannon limit only 0.0045 dB. Decoding algorithm adopts SPA (Sum-Product Algorithm). However, girths in LDPC codes are detrimental to the code's performance, and the existence of loops makes the decoding information iterative and the performance reduced. This paper, through computer simulation, using the Matlab cellular array, to convert the binary check matrix to tree matrix, realizes the algorithm of LDPC codes loop.

Keywords: Low-Density Parity-Check Codes; Tree; Girth

LDPC 码校验矩阵回路的求解算法

张焕明, 张 宾

佛山科学技术学院电子与信息工程系, 佛山
Email: 511106085@qq.com

收稿日期: 2013 年 12 月 10 日; 修回日期: 2013 年 12 月 13 日; 录用日期: 2013 年 12 月 16 日

摘要: 1996 年 LDPC(低密度奇偶校验, Low-Density Parity-Check)码是性能限与香农限仅差 0.0045 dB 的一种差错控制码^[1], 译码采用 SPA(和积算法), 但其性能受 Tanner 图中回路长度和回路数目的影响, 回路的存在使译码信息重复迭代, 性能下降^[2]。本论文通过计算机仿真, 采用 Matlab 元胞数组, 将二元校验矩阵转换为树矩阵, 实现了求解 LDPC 码回路的算法。

关键词: LDPC 码; 树; 回路

1. LDPC 码性能

LDPC 码可以用奇偶校验矩阵 H 来描述, 和所有的线性分组码一样, $C = \{x \in F_n \mid xH^T = 0\}$ 。同时也可以图论中的二分(Tanner)图来表示^[3,4]。若某个校验方程中出现了某个码字比特, 则在二分图的信息节点和校验节点出现连线, 对应着 H 矩阵中的元素 1。

LDPC 码的译码方法为 BP 算法, 本质上是一种迭代算法。影响译码性能的因素有二分图中回路的长度、数目, 尤其其中较短的回路。

2. LDPC 码回路的求解算法

为了求解 LDPC 码的 Tanner 图中的回路, 则需要先将校验矩阵进行转换为图论中的树^[5]。把 LDPC 码二分图重画为图论中的生长树, 生长过程中对重复出现的节点截断, 停止其生长, 并且记录截断节点的数目^[6]。这样的图为连通图中的单树, 否则为多树^[7]。

由树图方法可以看出, 每个回路中包含至少一个截断端点。因为截断端点的出现必然意味着节点的重

复, 又由于它们是树上的不同的分支, 所以, 那么必定有回路的出现。且如果一个节点在这样的树图中总共出现 n 次, 那么此节点出现在 C_n^2 个不同的回路中。

循着这样的思路, 本论文通过 Matlab 仿真, 实现求解 LDPC 码回路的算法如图 1。

仿真实现与结果:

对于校验矩阵为 $H = [1\ 0\ 1\ 0\ 1\ 0\ 1\ 0; 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1; 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0; 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1]$, 可以得到如下的树矩

阵 T , 并采用 matlab 的元胞数组进行描述:

其中树矩阵 T 中的元素 T_{ij} 的前两个值 $T_{ij(1,2)}$ 描述当前节点, 随后的两个值 $T_{ij(3,4)}$ 表明当前节点的父节点, 父节点唯一, 接续的值为 1, 1 的个数表示此节点被截断次数, 空矩阵表示无节点。由 T 可知, H 的 3 行 7 列的(3, 7)节点被截断 1 次, 它存在于 1 个回路中, 同理(4, 2)节点被截断 3 次, 存在于 3 个不同的回路中, 依次类推。从而可以得到回路如下:

[1 1 0 0]	[]	[]	[]	[]	[]	[]	[]	[]
[1 1 1 1]	[1 3 1 1]	[1 5 1 1]	[1 7 1 1 1]	[]	[]	[]	[]	[]
[2 1 1 1]	[3 3 1 3 1 1]	[4 5 1 5 1 1 1]	[3 7 1 7 1 1 1]	[]	[]	[]	[]	[]
[2 4 2 1 1]	[2 6 2 1 1]	[2 8 2 1 1]	[3 2 3 3 1]	[3 6 3 3 1]	[3 7 3 3 1]	[4 2 4 5 1]	[4 4 4 5 1]	[4 8 4 5 1]
[4 4 2 4 1 1 1 1]	[3 6 2 6 1 1 1]	[4 8 2 8 1 1 1 1]	[4 2 3 2 1 1 1 1]	[]	[]	[]	[]	[]

- 回路 1: 截断节点(3, 7)—(1, 7)—(1, 1)—(1, 3)—(3, 3)
- 回路 2: 截断节点(3, 6)—(3, 3)—(1, 3)—(1, 1)—(2, 1)—(2, 6)
- 回路 3: 截断节点(3, 7)—(3, 3)—(1, 3)—(1, 1)—(1, 7)
- 回路 4: 截断节点(4, 2)—(4, 5)—(1, 5)—(1, 1)—(1, 3)—(3, 3)—(3, 2)
- 回路 5: 截断节点(4, 4)—(4, 5)—(1, 5)—(1, 1)—(2, 1)—(2, 4)
- 回路 6: 截断节点(4, 8)—(4, 5)—(1, 5)—(1, 1)—(2, 1)—(2, 8)
- 回路 7: 截断节点(4, 4)—(2, 4)—(2, 1)—(1, 1)—(1, 5)—(4, 5)
- 回路 8: 截断节点(3, 6)—(2, 6)—(2, 1)—(1, 1)—(1, 3)—(3, 3)
- 回路 9: 截断节点(3, 6)—(2, 6)—(2, 1)—(1, 1)—(1, 7)—(3, 7)
- 回路 10: 截断节点(4, 8)—(2, 8)—(2, 1)—(1, 1)—(1, 5)—(4, 5)
- 回路 11: 截断节点(4, 8)—(2, 8)—(2, 4)—(4, 4)
- 回路 12: 截断节点(4, 2)—(3, 2)—(3, 3)—(1, 3)—(1, 1)—(1, 5)—(4, 5)
- 回路 13: 截断节点(4, 2)—(3, 2)—(3, 3)—(1, 3)—(1, 1)—(2, 1)—(2, 4)—(4, 4)
- 回路 14: 截断节点(4, 2)—(3, 2)—(3, 3)—(1, 3)—(1, 1)—(2, 1)—(2, 8)—(4, 8)

重新排序, 按从根节点出发, 就近节点生长原则, 得到

- 回路 1: (1, 1)—(1, 3)—(3, 3)—(3, 7)—(1, 7)
- 回路 2: (1, 1)—(1, 3)—(3, 3)—(3, 6)—(2, 6)—(2, 1)
- 回路 3: (1, 1)—(1, 3)—(3, 3)—(3, 7)—(1, 7)
- 回路 4: (1, 1)—(1, 3)—(3, 3)—(3, 2)—(4, 2)—(4, 5)—(1, 5)
- 回路 5: (1, 1)—(1, 5)—(4, 5)—(4, 4)—(2, 4)—(2, 1)
- 回路 6: (1, 1)—(1, 5)—(4, 5)—(4, 8)—(2, 8)—(2, 1)
- 回路 7: (1, 1)—(1, 5)—(4, 5)—(4, 4)—(2, 4)—(2, 1)
- 回路 8: (1, 1)—(1, 3)—(3, 3)—(3, 6)—(2, 6)—(2, 1)
- 回路 9: (1, 1)—(1, 7)—(3, 7)—(3, 6)—(2, 6)—(2, 1)
- 回路 10: (1, 1)—(1, 5)—(4, 5)—(4, 8)—(2, 8)—(2, 1)
- 回路 11: (2, 4)—(2, 8)—(4, 8)—(4, 4)
- 回路 12: (1, 1)—(1, 3)—(3, 3)—(3, 2)—(4, 2)—(4, 5)—(1, 5)
- 回路 13: (1, 1)—(1, 3)—(3, 3)—(3, 2)—(4, 2)—(4, 4)—(2, 4)—(2, 1)
- 回路 14: (1, 1)—(1, 3)—(3, 3)—(3, 2)—(4, 2)—(4, 8)—(2, 8)—(2, 1)

LDPC 码校验矩阵回路的求解算法

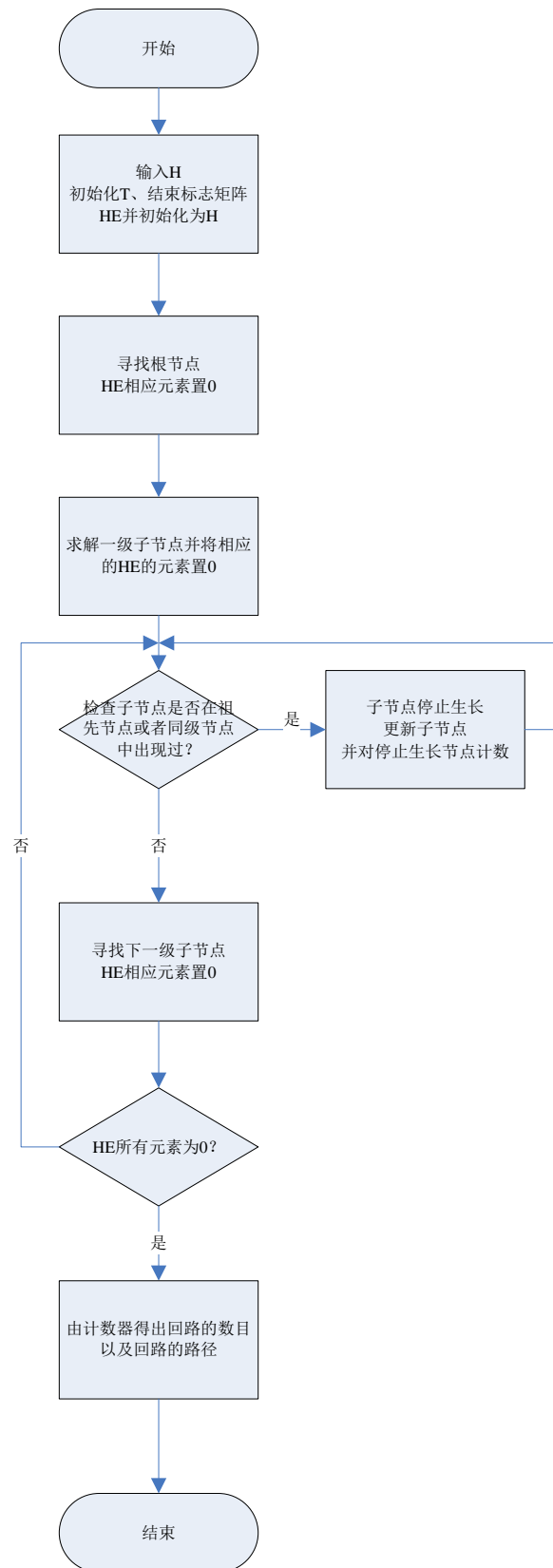


Figure 1. Simulation chart of principle
图 1. 仿真原理框图

通过对回路的重新排序容易找出重复出现的回路，排序规则如下：选择回路中节点 (m, n) ， $m + n$ 值最小的为排序第一个节点，如果有多个节点 $m + n$ 的值最小且相等，则选择 m 值小的优先，其次选择余下节点 n 值小的节点为排序第二个节点，然后按回路。由上面结果：回路 1 等价于回路 3，回路 2 等价回路 8，回路 4 等价回路 10，回路 5 等价回路 7，回路 6 等价回路 9。由于截断的次序，它们的出现只是方向不同而已，重新排序后一样。也就是说其中的回路只有回路 1(3)、2(8)、4(12)、5(7)、6(10)、9、11、13、14，其回路的长度分别为 4、6、6、6、6、6、4、8、8。但由于 H 的复杂性，需要注意的下面几点：

如果 H 为连通的，则为单个的截断树矩阵， HE 元素全为零，即为结束标志。否则为多树，此时对每次不再生长后的 HE ，作为新的矩阵，按上面方法重求截断树矩阵，形成迭代运算，直到最终 HE 的元素全为 0。

图为计算机仿真实现的算法流程图 1。

3. 结论

求解校验矩阵的回路是个复杂的过程，特别对于

大型的 H 矩，而且对于回路缠绕情况下有待给出更明确的定义。本文利用 Matlab 的元胞数组，把 Tanner 转换为图论中的截断树，并且利用 Matlab 将截断树表示为元胞矩阵，以矩阵的形式表述树，从而给出了求解 LDPC 码 Tanner 图中的回路的算法，并通过计算机进行仿真。仿真结果对于分析回路和性能之间的关系奠定了基础。

参考文献 (References)

- [1] Gallager, R.G. (1963) Low-density parity check codes. MIT Press, Cambridge.
- [2] Mackay, D.J.C. (1999) Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, **45**, 399-431.
- [3] Tanner, R.M. (1981) A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, **27**, 533-547.
- [4] Tanner, R.M., Sridhara, D. and Fuja, T. (2001) A class of group-structured LDPC codes. *Proceedings of ICSTA*, Amleside, 15-20 July 2001.
- [5] Wiberg, N. (1996) Codes and decoding on general graphs, Ph.D. Dissertation, Linkoping Studies in Science and Technology, University of Linkoping.
- [6] McGowan, J.A. and Williamson, R.C. (2003) Loop removal from LDPC codes. *IEEE Information Theory Workshop*, 230-233.
- [7] Reinhard, D. (1997) Graph theory. Springer-Verlag, New York.