適用於5G NR接收機的LDPC Codes平行處理解碼器之研究

(Study on Parallel LDPC Decoder for 5G NR Receiver)

學生: 蘇沛錦　　指導教授: 李昌明

摘要

低密度同位檢查碼 (LDPC Codes) 為錯誤更正碼的一種，用於更正資料在傳輸時受到通道雜訊干擾所造成的錯誤。由於LDPC Codes具有非常接近Shannon極限的卓越性能和保護大量資料傳輸之優勢，因此被3GPP採用作為5G NR標準中數據傳輸的編碼方式。本專題是使用Column by Column的平行處理方式，設計適用於5G NR標準的LDPC Codes解碼器。然而此解碼方式必須確保檢測矩陣內同一個Row的偏移值 (Offset) 都不同，否則會產生記憶體存取衝突，導致解碼時間延遲。所以需要標定NR LDPC檢測矩陣中同一個Row發生衝突的位置，並將這些會產生衝突的Column分成多個Column Set進行處理。

最小和解碼演算法

LDPC Codes常見的解碼方式為訊息傳遞演算法 (MPA)，針對接收碼字y_i進行事後機率之對數相似比例 (LLR)計算，以判斷c_i的正確值為1或0。假設通道模型為AWGN channel，則第i個碼字c_i之LLR計算如下:

The AWGN channel with mean $\mu = 0$ and variance σ^2

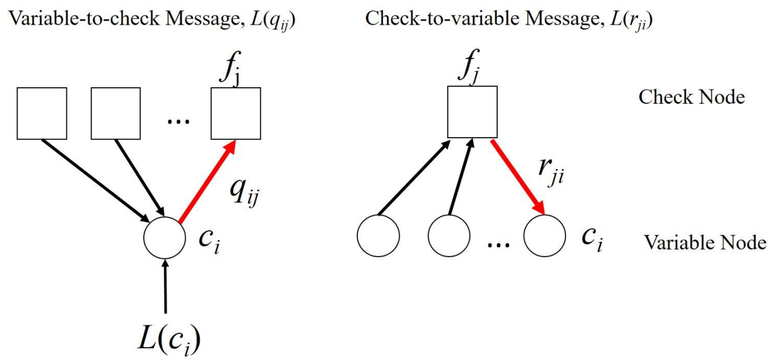
c: Transmitted Codeword　　$c = [c_0\ c_1\ \cdots\ c_{n-1}]_{1\times n}$

y: Received Codeword　　$y = [y_0\ y_1\ \cdots\ y_{n-1}]_{1\times n}$

$$L(c_i) \equiv \log\big(l(c_i)\big) = \log\left(\frac{P_r(c_i = 0|y_i)}{P_r(c_i = 1|y_i)}\right) = 2\frac{y_i}{\sigma^2} \qquad (1)$$

最小和演算法 (MSA)是將對數域之MPA的機率運算過程再進行化簡、近似處理，整個解碼流程包含以下四個步驟:

1. 根據(1)式，初始化各變數節點之訊息$L(c_i)$，且$L(q_{ij}) = L(c_i)$。
2. 檢測節點向下更新訊息$L(r_{ji})$至所連接的變數節點。
3. 反之，變數節點向上傳遞訊息$L(q_{ij})$至所連接的檢測節點。
4. 觀察是否達到終止條件或最大迭代次數，否則回到步驟2進行迭代運算。



Normalized MSA:

$$L(r_{ji}) = \left(\prod_{i'\in R_{j\setminus i}} \alpha_{i'j}\right) \times k \times \min_{i'\in R_{j\setminus i}} \beta_{i'j} \qquad (2)$$

$$\alpha \equiv sign\left(L(q_{ij})\right),\ \beta \equiv \left|L(q_{ij})\right|$$

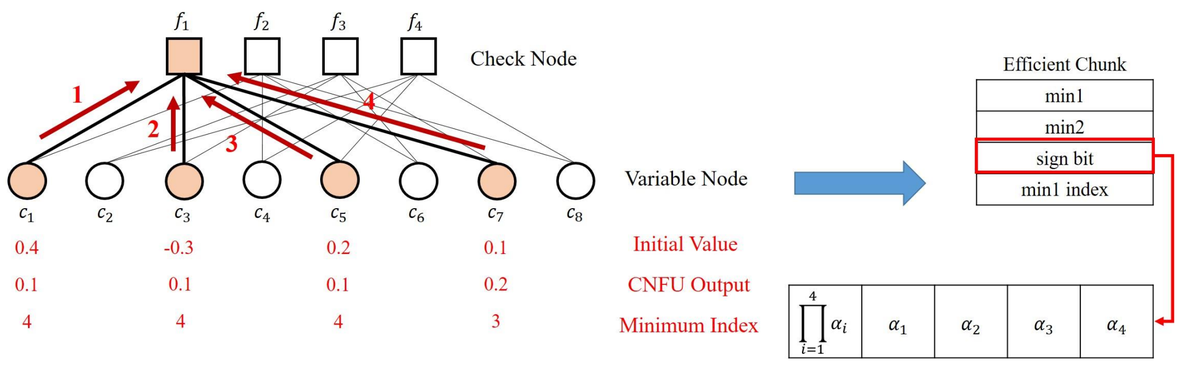
k : Normalization factor ($0.6 \le k \le 0.9$)

$$L(q_{ij}) = L(c_i) + \sum_{j'\in C_{i\setminus j}} L(r_{j'i}) \qquad (3)$$

Column by Column平行處理方式

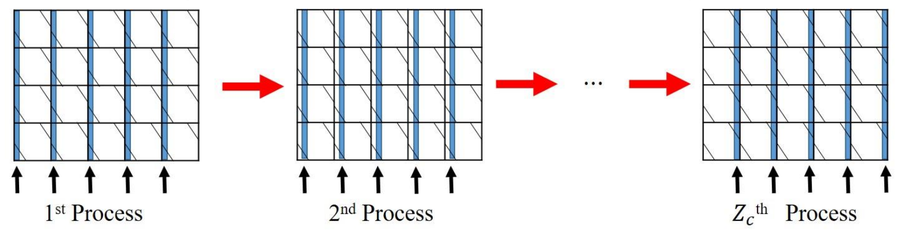
觀察MSA在進行$L(r_{ji})$訊息運算的過程中(圖一)，可發現更新完所有變數節點的值最多只有二種，分別是0.1和0.2，因此可得出二項重要結論:

1. 解碼器只需少量儲存全部變數節點的$L(r_{ji})$訊息最小值(min1)、第二小值(min2)、最小值索引，以及各變數節點之正負符號位元到規劃好的記憶體空間(Efficient Chunk)。
2. 後續在進行$L(q_{ij})$訊息更新時，可同時對下一次迭代計算所使用的$L(r_{ji})$ 進行預測。



圖(一) (8,4) 檢測矩陣H更新檢測節點上的$L(r_{ji})$訊息和簡化儲存之Efficient Chunk

因此NR LDPC Codes在解碼流程上，可使用Column by Column方式進行平行處理(如圖二所示)。在初始化時先更新全部的$L(q_{ij})$訊息，並且將計算完成的$L(r_{ji})$訊息簡化儲存至記憶體。接著進行$L(q_{ij})$訊息運算時，可從Efficient Chunk還原出正確的$L(r_{ji})$數值，再更新記憶體。經由Z_C次的處理後，所有Efficient Chunk皆更新完成，提供給下一次的迭代運算使用。



圖(二) Column by Column平行處理方式

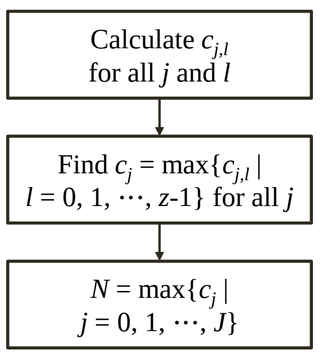
避免資料存取衝突之資料結構

然而在NR LDPC Codes的檢測矩陣內，同一列偏移值會有相同的情況出現，這會導致資料存取發生衝突。因此將這些Column平均分配成多組的Column Set，不僅能避免記憶體衝突所產生的解碼延遲問題，還可提高解碼器的處理速度。

針對NR LDPC Codes尋找平行處理之Column Set的流程如下:

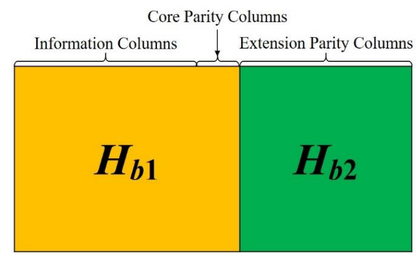
Step 1. Find the number of column sets (chunks) N

- The block row index j = 1, 2, , J.
- The offset value l = 0, 1, ⋯, z-1.
- $c_{j,l}$: the number of sub-matrices with the offset l in the jth block row.
- c_j: the maximum number of sub-matrices with the same offset in the jth block row.
- N: the number of column sets in the parity check matrix H .

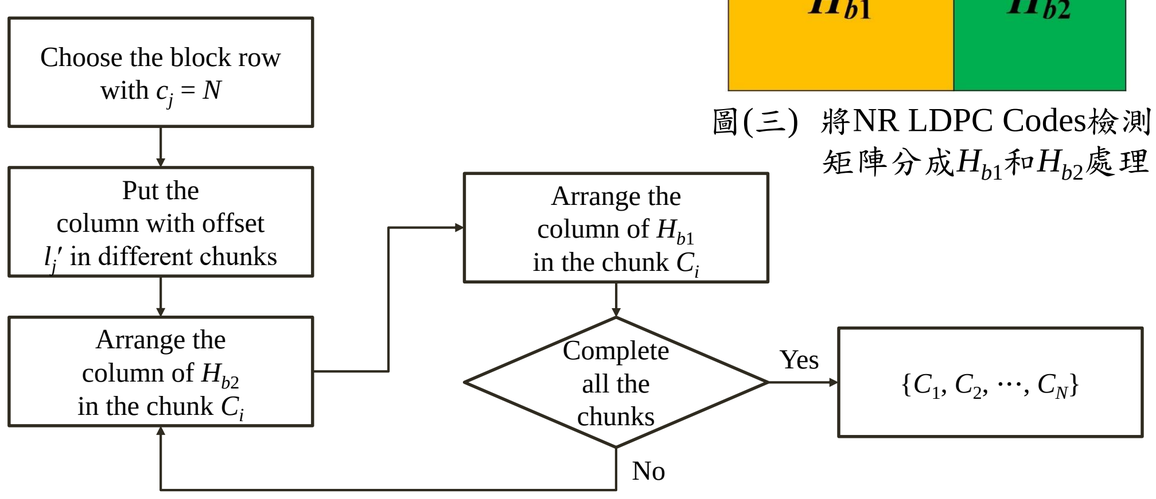


Step 2. Construction of column sets $\{C_1, C_2, \ldots, C_N\}$

- The offset value $l_j' = \arg\max_l \{c_{j,l}\}$.
- The index of column set (chunk) i = 1, 2, ⋯, N.

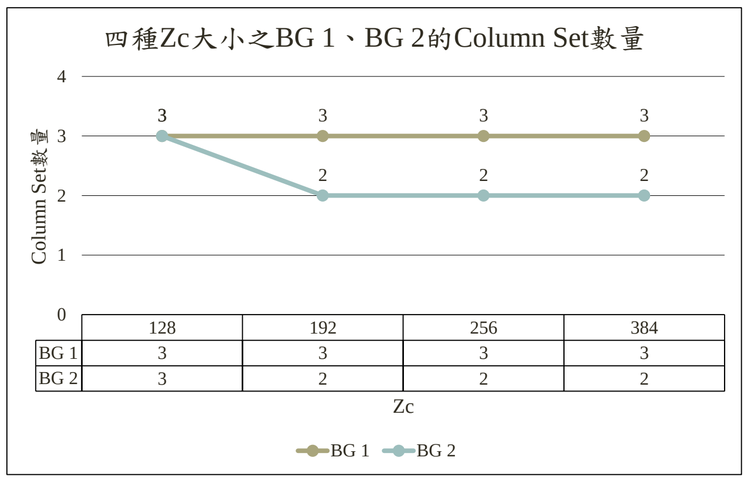


圖(三) 將NR LDPC Codes檢測矩陣分成H_{b1}和H_{b2}處理



實驗結果

以下數據顯示兩種矩陣針對不同Z_C大小的平行化處理程度，首先BG 1在四種Z_C的條件下進行Column Set分配，得知其平行解碼加速皆可提升為原本的三倍；而BG 2除了Z_C = 128外，其餘三種Z_C的處理速度則加快二倍。



	128	192	256	384
BG 1	3	3	3	3
BG 2	3	2	2	2

結論

本專題針對 5G NR LDPC Codes，使用Column by Column平行處理的解碼方式。考量檢測矩陣中同一個Row之偏移值重複問題，藉由分配多組Column Set來避免資料存取衝突發生，並且提高平行處理速度，以滿足5G高速、低延遲的通訊要求。未來希望能使用此解碼方式在非二元類循環LDPC Codes的解碼器。