行政院國家科學委員會專題研究計畫 成果報告

在低密度奇偶校正碼中運用動態排程以達成不均等的錯誤保護

研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 98-2221-E-216-012-
執 行 期 間 ：98年08月01日至99年07月31日
執 行 單 位 ：中華大學資訊工程學系

計畫主持人：翁文彥
共同主持人：劉懷仁
計畫參與人員：碩士班研究生-兼任助理人員：董致銓
碩士班研究生-兼任助理人員：陳怡光
碩士班研究生-兼任助理人員：葉繕銘
碩士班研究生-兼任助理人員：吳政哲
碩士班研究生-兼任助理人員：蕭仁瑋

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 100年01月01日

行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※　　在低密度奇偶校正碼中運用動態排程　　　　※
※　　　　以達成不均等的錯誤保護　　　　　　　※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：個別型計畫
計畫編號：NSC　98－2221－E－216－012－
執行期間：　98 年　8　月　1 日至　99 年　7 月　31 日

計畫主持人：翁文彥
共同主持人：劉懷仁
計畫參與人員：葉繕銘, 董致銓, 吳政哲, 蕭仁瑋 ·

執行單位：中華大學

中　華　民　國　　99　年　7 月　31　日

Table of Contents

Contents of Report

(1) Preface

Error correction codes usually put the same protection on the information bits. This property is called Equal Error Protection (EEP). However, in some applications, a portion of the information bits are more important than the others and thus require more protection on them. Channel codes that provide this property are called Unequal Error Protection (UEP) codes. For a UEP code, information bits are usually divided into two classes, one called More Important Bits (MIB) and the other called Less Important Bits (LIB). In UEP applications, a UEP code can reduce the total amount of redundancy as compared to EEP codes. This project focused on designing new decoding scheduling algorithms to make a single UEP-LDPC code satisfy variable combinations of MIB/LIB BER requirements.

(2) Objective

For most studies of UEP LDPC codes in the literature, flooding scheduling is used. The simulation results demonstrate different BERs for MIB and LIB of the transmitted message. However, the gap between BERs for MIB and LIB only depend on the code design, i.e. the degree distributions. In our simulation of UEP LDPC codes decoded using SSS and IDS, we do observe lower BER and faster convergence compared to flooding. Still, the gap between BERs for MIB and LIB is fixed once the degree distributions of the variable nodes are chosen.

In some cases, we might want to change the BER requirements for MIB and LIB while the LDPC code is already determined, say from the standards. Then the decoder iteration must be run long enough until both of the requirement is satisfied. The additional number of iterations wastes time and power. So, we are motivated by this observation to design new decoding scheduling algorithms to make a single UEP-LDPC code satisfy variable combinations of MIB/LIB BER requirements.

In this project, we propose to use RBP as the scheduling for UEP-LDPC code to improve the convergence speed. Besides, we set additional rules to the node-selection order such that BERs of MIB and of LIB can be adjusted. The rules include setting LLR ranking thresholds, updating quota, and weighted residues. The intuition is to let MIB and LIB nodes to have different updating priorities which can be controlled by the parameters of our algorithms.

(3) Literature review

Several studies in the literature proposed solutions for the UEP problem. Multilevel coding with multi-stage decoding can provide UEP property. As for single level coding, two independent codes can be used for MIB and LIB independently. This method is called Time-sharing. For MIB, a lower rate code is used for better protection while the LIB uses a higher rate code. Time-sharing is simple but the problem is that the MIB which corresponds to the header part of a packet or image is usually short in length. It is obvious that a short error correction code suffers from serious performance degradation. Another scheme is to use rate-compatible codes with puncturing (RCPC/RCPT codes). In this scheme, a lower-rate mother code can be punctured into a higher rate code to adjust the level of protection. RCPC/RCPT is more flexible in terms of code-rate selection for MIB and LIB if compared with time sharing. However, the performance of RCPC/RCPT codes also suffers from shortened blocklength.

From the above discussion, we can conclude that a single code that can provide UEP property is desirable. It turns out that the irregular LDPC code is a natural fit for UEP applications. Rahnavard et al. proposed a partially regular UEP LDPC code where the variable nodes are divided into three groups, MIB, LIB and Parity Bits (PB).

LDPC codes are usually decoded by running an iterative message-passing algorithm over the underlying bi-partite graph of the code. This iterative algorithm is also called Belief Propagation (BP). The problem of determining the order that the variable nodes and the check nodes update their message is called *scheduling*. Traditional message-passing schedule adopts the flooding strategy. Although flooding has almost become the standard LDPC decoder scheduling, several studies show that different scheduling strategies can outperform flooding in terms of convergence speed and frame error rate (FER). These strategies include Standard Sequential Scheduling (SSS) and Informed Dynamic Scheduling (IDS).

The first IDS algorithm was proposed by Elidean et al. as the Residual Belief Propagation (RBP) algorithm. RBP updates the node that has the largest difference between its message in the current iteration and its message in the previous iteration. This is a greedy algorithm with the intuition of finding the node whose LLR is increasing the fastest such that the iteration can be speed up. RBP was found to converge even faster than SSS. However, its performance for a large number of iterations becomes worse since the greedy algorithm sometimes converges to the wrong codeword. Node-wise RBP (NW-RBP) updates the variable nodes in sequence as in RBP but updates the check nodes in parallel. The NW-RBP is less greedy than the RBP and turns out to be a good trade-off between convergence speed and error performance.

(4) Methodology

We use C++ to implement a software simulation environment for UEP-LDPC codes. The simulator can take the input of any arbitrary LDPC parity check matrix and run simulation over AWGN. The simulation environment can use flooding or RBP in the message update and the three proposed algorithms will then be implemented.

The system consists of a binary LDPC encoder followed by a signal mapper and then the coded bits are sent over the channel. The channel is additive white Gaussian Noise channel (AWGN). At the receiving side, first the received symbols are de-mapped assuming that perfect channel side information can be obtained at the receiver. Then the LLRs are passed to the iterative LDPC decoder running according to the proposed IDS algorithm and stopping criteria.

We generated the optimal degree distribution using UEP density evolution (DE). With this degree distribution, we will construct the UEP-LDPC codes using random construction with graph-conditioning methods. Graph conditioning aims to avoid short cycles and small stopping sets such that the resulting BER of the LDPC codes have low error floors. The block length will be medium length between 1,000 and 3,000 bits for practical reasons. Several different code rates will be selected.

We propose three scheduling algorithms to make the bit error rates of UEP-LDPC flexible. The intuition is to change the preference of the message update order of MIB and LIB bits. The algorithms are described as follows:

(a) LLR Ranking Threshold (LRT)

In this method, in addition to the residue queue, we create two queues, QMIB and QLIB, for MIB and LIB respectively. QMIB and QLIB are sorted according to the LLR of the variable nodes in non-decreasing order. When performing RBP, the message with the largest residue is updated if LLR of the corresponding variable node is below the pre-determined ranking thresholds, LRTMIB or LRTLIB. Otherwise, that message is not updated and second message is examined to see if it satisfies LRT and so forth. The threshold is a ranking percentage inside the queue. For example, LRTMIB=0.2 means that the variable nodes with LLR in the smaller 20% of QMIB are considered for updates while the larger 80% of QMIB are skipped. In other words, LRT constrains the more reliable (high LLR) nodes from updating their messages even if that message has the largest residue.

(b) Node Quota (NQ)

In this method, we set two quotas, TMIB and TLIB, for the edges that point to MIB and LIB respectively. When performing RBP, after the message with the largest residue is updated, the corresponding quota decreases by one. If the corresponding quota is zero, that message must not be chosen for message update. The next message in the residue queue will be chosen and checked for its quota. When both TMIB and TLIB are become zero, we will reset the quotas to start the next round. Through the NQ algorithm, we can adjust the number of updates of MIB and LIB and hence change their BERs.

(c) Weighted Residue (WR)

In this method, we set different weighting, WMIB and WLIB, for the edges that point to MIB and LIB respectively. Similar to RBP, the message with the largest residue is updated only that the residue is multiplied by its weighting. For example, we may apply a large weighting on MIB edges, then MIB edges will have higher priority than LIB edges. Through the WR algorithm, we can adjust the number of updates of MIB and LIB and hence change their BERs.

(5) Findings and discussions

Figure 1 shows a set of simulation of a UEP LDPC code using two different LRT thresholds. The LDPC code used is rate-1/2 and the code length is 1944 bits. Among the 972 information, 140 bits are the MIB and the other 832 bits are the LIB. MIB has degree 20 and LIB has degree 2 and 3. In LRT case A, we choose the thresholds as LRTMIB=0.25 and LRTLIB=1. In LRT case B, LRTMIB=0.15 and LRTLIB=1. From Figure 1, it is observed that by skipping the high LLR MIB updates, we are able to maneuver the gap between the BERs of MIB and LIB.

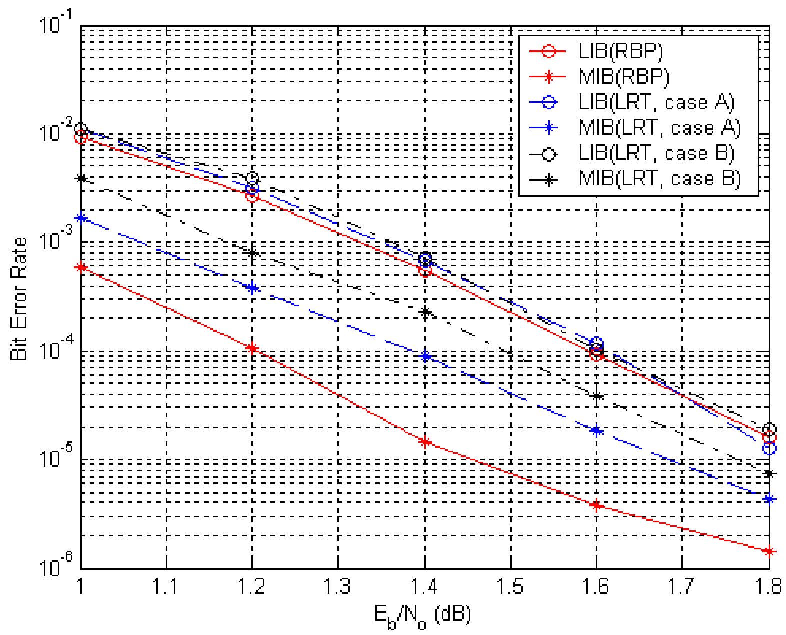

Figure 1. With the LRT algorithm, we can adjust the BER curves of MIB by setting different ranking threshold. The BER curves of LIB virtually remain the same. Therefore, the gap between LIB and MIB can be maneuvered.

Figure 2 shows the effect of the NQ method on the same UEP LDPC code in Figure 1. The Eb/No is 1.4dB. We choose the quotas as TMIB=1822, TLIB=712. It is found that so far the NQ algorithm can only slightly change the gap between the BERs of MIB and LIB. However, it could slow down the convergence. We think that the NQ algorithm worth more studying since it may play a role in the future if we consider a combination of the three algorithms.

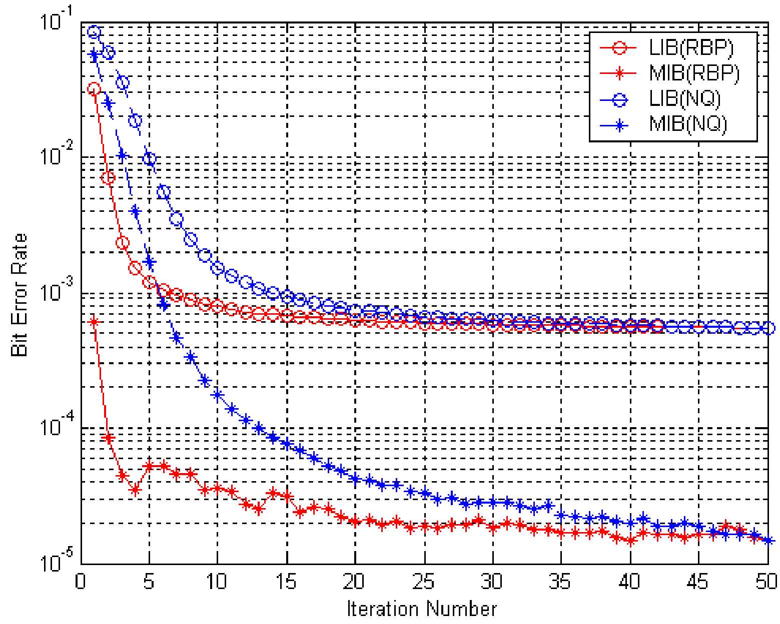

Figure 2. With the NQ algorithm, the BER of MIB can move upwards while the BER of LIB virtually remains the same. Therefore, the gap between LIB and MIB is slightly reduced.

Figure 3 shows the potential of the WR method on the same UEP LDPC code in Figure 1. The Eb/No is 1.6dB. We choose the weightings as WMIB=300, WLIB=1. Basically the parameters make the MIB much more preferred in the residue queue than the LIB. It is interesting to find that this choice of parameters brings the MLB and LIB BER curves very close to each other. The UEP LDPC code behaves almost like an EEP LDPC code. We have not fully understood how to set the appropriate weightings to control the gap and that is what we will work on.

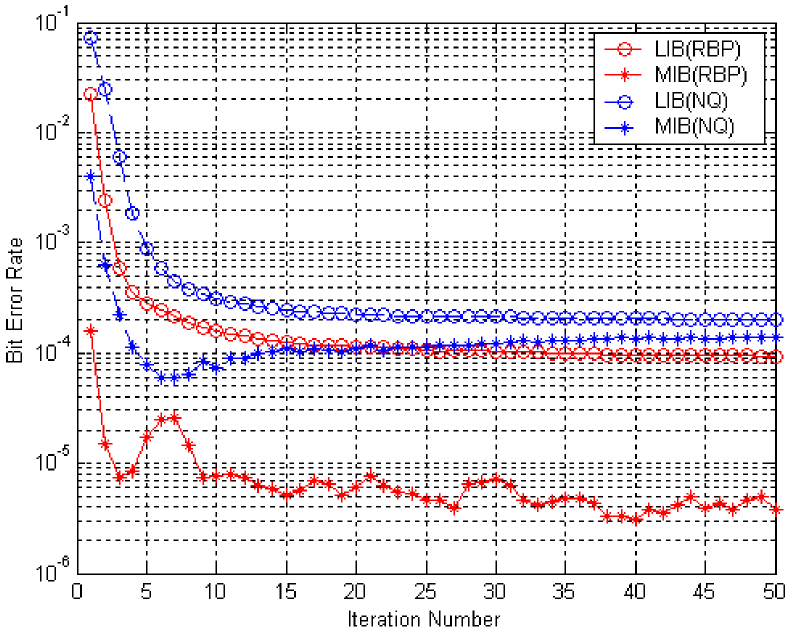

Figure 3. With the WR algorithm, the BER of MIB can move upwards and become very close to the LIB curve. Therefore, the gap between LIB and MIB is significantly reduced and the UEP LDPC code behaves almost like an EEP LDPC code.

Self Evaluation

We successfully demonstrated that the proposed LRT, NQ and WR algorithms can be used to change the BER gap of MIB and LIB. However, how to quantitatively choose the appropriate parameters to achieve the design goal is still open. We will need more simulation results and mathematical derivations to solve the optimization problem.

Future work can be study on combining the three algorithms to further improve the performance of the designed UEP-LDPC code. Also we might work on dynamic thresholds as iteration proceeds. How to design a good algorithm while keeping the computational complexity affordable is also an issue.

無研發成果推廣資料

98 年度專題研究計畫研究成果彙整表

計畫主持人：翁文彥	計畫編號：98-2221-E-216-012-
計畫名稱：在低密度奇偶校正碼中運用動態排程以達成不均等的錯誤保護	

成果項目			量化：實際已達成數（被接受或已發表）	量化：預期總達成數(含實際已達成數)	量化：本計畫實際貢獻百分比	單位	備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）
國內	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	0	0	100%		
		專書	0	0	100%		
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（本國籍）	碩士生	5	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		
國外	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	0	0	100%		
		專書	0	0	100%	章/本	
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（外國籍）	碩士生	0	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		

其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)	無.

	成果項目	量化	名稱或內容性質簡述
科教處計畫加填項目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與（閱聽）人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

■達成目標

□未達成目標（請說明，以 100 字為限）

□實驗失敗

□因故實驗中斷

□其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文：□已發表 □未發表之文稿 ■撰寫中 □無

專利：□已獲得 □申請中 ■無

技轉：□已技轉 □洽談中 ■無

其他：（以 100 字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

學術成就方面, 雖有初步成果, 但對於效能之改進程度並不顯著, 故尚未具備高的應用價值及社會影響. 有待繼續發展, 更深入了解實驗結果之意義及改進方式.