A STUDY ON THE SET CHOICE OF MULTIPLE FACTOR GRAPH BELIEF PROPAGATION DECODERS FOR POLAR CODES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY
ŞÜKRÜ CAN AKDOĞAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2018

Approval of the thesis:

A STUDY ON THE SET CHOICE OF MULTIPLE FACTOR GRAPH BELIEF PROPAGATION DECODERS FOR POLAR CODES

submitted by **ŞÜKRÜ CAN AKDOĞAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Melek Diker Yücel
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Yalçın Tanık
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Melek Diker Yücel
Electrical and Electronics Engineering Dept., METU _____

Prof. Dr. Erdal Arıkan
Electrical and Electronics Engineering Dept., BİLKENT _____

Prof. Dr. Ali Özgür Yılmaz
Electrical and Electronics Engineering Dept., METU _____

Assist. Prof. Dr. Gökhan Muzaffer Güvensen
Electrical and Electronics Engineering Dept., METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : ŞÜKRÜ CAN AKDOĞAN

Signature:

ABSTRACT

A STUDY ON THE SET CHOICE OF MULTIPLE FACTOR GRAPH BELIEF PROPAGATION DECODERS FOR POLAR CODES

Akdoğan, Şükrü Can

MSc., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Melek Diker Yücel

July 2018, 93 pages

Polar codes are linear block codes with low encoding and decoding complexity that are proven to achieve the channel capacity for any given binary-input discrete memoryless channel as the codeword length N goes to infinity. The main idea of polar codes is about channel polarization. Special factor graphs are used to represent the encoding and decoding structure of polar codes. These factor graphs consist of n stages for (N, K) codes, where $n = log_2 N$. Each stage contains $N/2$ many Z-shaped connections between specific input-output pairs. Defining the height of a Z connection as the node distance between its input (or output) nodes; Stage-i that occurs only once in a factor graph for each $i = 1, \dots, n$, contains Z-shapes of height $2^{(i-1)}$. Encoding is done with a specific factor graph, whose Z connections are ordered from the largest to the smallest, that we refer to as the "reference factor graph, $n \dots 321$". However, belief propagation decoding can be implemented with one of the $n!$ different factor graphs, by permuting the stages of the reference factor graph.

In this thesis, we study on belief propagation decoding of rate ½ polar codes over a binary erasure channel, using single or multiple factor graphs within the decoder. In order to classify the factor graphs, we propose the "stage order number", in addition to the "number of frozen variables" and "capacity sum" parameters considered by [Doğan, 2015] and [Peker, 2018]. For $n = 6$, single factor graph decoding performances of $n!$ factor graphs are demonstrated and related to the mentioned three parameters. Construction of compatible factor graph sets, in order to get better performance in multiple-factor graph decoding is studied and their performance is found for $n = 6, 7, \dots, 11$ by simulations. Since the sets, consisting of factor graphs with parameters varying in wide ranges seem to have better performance, we propose the "Set Choice" algorithm to compose factor graph sets with ensured variety. Multiple decoders using sets of $\lceil n/2 \rceil$ factor graphs obtained by this algorithm can achieve similar performance to that of the cyclic set of n factor graphs.

Keywords: Polar Codes, Belief Propagation, Multiple Factor Graph Decoder

ÖZ

KUTUPSAL KODLAR İÇİN ÇOK FAKTÖR DİYAGRAMLI İNANÇ YAYILIMI ÇÖZÜCÜLERİNDE KÜME SEÇİMİ ÜZERİNE BİR ÇALIŞMA

Akdoğan, Şükrü Can

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Melek Diker Yücel

Temmuz 2018, 93 sayfa

Kutupsal kodlar, kod sözcüğü uzunluğu N sonsuza giderken verilen herhangi ikili-girişli ayrık hafızasız kanal için kanal kapasitesine ulaştığı kanıtlanan, düşük kodlama ve kod çözme karmaşıklıklı doğrusal blok kodlardır. Kutupsal kodların ana fikri kanal kutuplaşmasıdır. Kutupsal kodların kodlama ve kod çözme yapılarını göstermek için özel faktör diyagramlar kullanılır. Bu faktör diyagramlar, (N, K) kodlar için $n = log_2 N$ kademeden oluşur. Her bir kademe, belirli giriş-çıkış ikil çiftleri arasındaki Z-şekilli bağlantılardan $N/2$ tane içerir. Bir Z bağlantısının yüksekliği, iki giriş (ya da çıkış) noktasının arasındaki nokta uzaklığı olarak tanımlanır. Bir faktör diyagramda her bir $i = 1, \dots, n$ için yalnızca bir tane olan Kademe-i, yüksekliği $2^{(i-1)}$ olan Z-şekilleri içerir. Kodlama işlemi, "referans faktör diyagram, $n \dots 321$" olarak tanımladığımız ve Z bağlantıları en büyükten en küçüğe doğru sıralanmış özel faktör diyagram ile yapılır. Ancak, inanç yayılımlı kod çözme işlemi, referans faktör diyagram kademelerinin sırasını değiştirerek elde edilen $n!$ farklı faktör diyagramdan biriyle yapılabilir.

Bu tezde, ikili silinti kanalında, kod oranı ½ olan kutupsal kodların inanç yayılımlı, tek veya çok faktör diyagramlı kod çözücüleri üzerinde çalışılmış; faktör diyagramlarının sınıflandırılması için, [Doğan, 2015] ve [Peker, 2018] tarafından düşünülen "donuk değişken sayıları" ve "kapasite toplamı" parametrelerine ek olarak "kademe sıra numarası" önerilmiştir. $n!$ faktör diyagramdan her birinin tek faktör diyagramlı kod çözücü başarımları, $n = 6$ için sergilenmiş ve sözü edilen üç parametre ile ilişkilendirilmiştir. Çok faktör diyagramlı kod çözücülerde daha iyi başarım elde etmek için uyumlu faktör diyagram kümeleri oluşturulmuş, ve bu kümelerin başarımları $n = 6, 7, \dots, 11$ için benzetimle elde edilmiştir. Parametreleri geniş aralıkta değişen faktör diyagram kümelerinin daha iyi başarım göstermesi nedeniyle, çeşitlilik sağlayan bir küme seçim algoritması sunulmuştur. Bu algoritma ile elde edilen $\lceil n/2 \rceil$ faktör diyagramlı kümelerin kullanıldığı kod çözücüler, n faktör diyagramlı döngüsel kümelerinki ile benzer başarımlara erişmektedir.

Anahtar Kelimeler: Kutupsal Kodlar, İnanç Yayılımı, Çok Faktör Diyagramlı Kod Çözücü

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Assoc. Prof. Dr. Melek Diker Yücel for her endless help, patience and sharing her knowledge. She always had a supportive approach during this research. I could not have imagined finishing this study without her pearls of wisdom.

Also, I would like to thank my family for their encouragement, support and love all the time.

My sincere thanks also go to my friends and colleagues for their patience, moral support and privilege provided through this study.

Lastly, I am grateful to my company for providing me with the opportunities to finish this thesis.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xii

LIST OF FIGURES xiii

LIST OF ABBREVIATIONS xvii

1 INTRODUCTION 1

1.1 Channel Coding 1

1.2 Belief Propagation Decoder 3

1.3 Aim and Organization of the Thesis 7

2 REVIEW OF POLAR CODES 9

2.1 Preliminaries 9

2.2 Channel Transformation 11

2.2.1 Basic Channel Combining 11

2.2.2 Basic Channel Splitting 12

2.2.3 Recursive Channel Combining and Splitting 14

2.3 Channel Polarization 18

2.4 Capacities of Transformed N Distinct Channels $W_N^{(i)}$ over a BEC 20

2.5 Polar Code Encoding 22

2.6 Belief Propagation Decoding Algorithm 26

3 SIMULATION RESULTS 31

3.1 Number of Frozen Variables (FV), Capacity Sum (CS) and Stage Order Number (SON) Parameters of the Single-FG BP Decoder 32

3.1.1 Capacity Sum (CS) Calculation of Factor Graphs 33

3.1.2 Number of Frozen Variables (FV) Calculation of Factor Graphs 36

3.1.3 Number of Frozen Variable and Capacity Sum Calculation for Reed Muller Codes 41

3.1.4 Interrelation between the Number of Frozen Variables, Capacity Sum and Stage Order Numbers 45

3.1.5 Single-FG Performance Dependence on the Number of Frozen Variables, Capacity Sum and Stage Order Number 48

3.2 Multiple-FG Belief Propagation Decoding 52

3.2.1 Dependent versus Independent *M*-FG BP Decoding 53

3.2.2 Changing the Order of Factor Graphs in Dependent *M*-FG BP Decoding 55

3.2.3 Twice Repeated *M*-FG BP Decoding versus 2*M*-FG Decoding 57

3.2.4 *M* Factor Graphs with Similar/Different FV and CS Values 61

3.2.5 Getting Similar Performance with Less Number of Factor Graphs 64

3.2.6 Relation between the Performance and Stage Order Numbers of the Factor Graph Set 66

3.2.7 Set Choice Algorithm for $M = \lceil n/2 \rceil$ Factor Graphs 68

3.2.8 Performance Comparison of *M*-FG Belief Propagation Decoders with $M = \lceil n/2 \rceil$ and $M = n$ 72

3.2.9 Polar Code Construction and Related Performance for Different Erasure Rates 78

4 CONCLUSION 85

REFERENCES 89

LIST OF TABLES

TABLE 3.1 FROZEN VARIABLE MATRICES OF RFG AND FG WITH LEAST STAGE ORDER NUMBER FOR THE (32, 16) ADAPTIVE POLAR CODE OVER BEC (0.35). 40

TABLE 3.2 DISTRIBUTION OF EQUI-CS AND EQUI-FV SETS FOR THE (32, 16) POLAR CODE OVER BEC(0.35). (REPRODUCED FROM [PEKER, 2018]) 41

TABLE 3.3 INFORMATION BIT INDICES AND ROWS OF GENERATOR MATRICES FOR THE (8, 4) POLAR AND REED-MULLER CODES OVER BEC (0.35). 42

TABLE 3.4 INFORMATION BIT CAPACITIES OF RFG AND HAMMING WEIGHTS OF ROWS FOR THE (32, 16) POLAR AND REED-MULLER CODES OVER BEC (0.35). 43

TABLE 3.5 BIT CAPACITIES AND FROZEN VARIABLE MATRICES OF FACTOR GRAPHS WITH STAGE ORDER NUMBER 54321 AND 12345 FOR THE (32, 16) REED-MULLER CODES OVER BEC (0.35). 44

TABLE 3.6 CORRELATION COEFFICIENTS r_{xy} BETWEEN THE FV-CS, FV-SON AND CS-SON PARAMETERS OF DIFFERENT POLAR CODES OVER BEC(0.35). 48

TABLE 3.7 CODEWORD ERROR RATIO (CER) PERFORMANCE COMPARISON OF MINIMUM AND MAXIMUM EQUI-CS SETS AT SINGLE-FG BP DECODER FOR THE (64, 32) ADAPTIVE POLAR CODES OVER BEC(0.35) 50

TABLE 3.8 STAGE ORDER NUMBERS (SON'S) OF THE THREE SETS USED IN THIS SECTION. 62

TABLE 3.9 STAGE ORDER NUMBERS (SON'S) OF THE 4-FG SETS USED IN THIS SECTION (4-FG SUBSET IN THE SECOND ROW HAS 1% BETTER CER). 66

TABLE 3.10 STAGE ORDER NUMBERS (SON'S) OF THE 4-FG SETS USED IN THIS SECTION (4-FG SUBSET IN THE SECOND ROW HAS 1% BETTER CER). 67

TABLE 3.11 STAGE ORDER NUMBERS (SON'S) OF THE 4-FG SETS USED IN THIS SECTION 70

TABLE 3.12 STAGE ORDER NUMBERS (SON'S) OF THE $\lceil n/2 \rceil$-FG SETS USED IN SECTION 3.2.8. 76

TABLE 3.13 BASIS CODE VECTORS (AND CORRESPONDING HAMMING WEIGHTS) OF THE (128, 64) RM AND ADAPTIVE POLAR CODES FOR VARIOUS ε, THE FIRST COLUMN INDICATING THE ROW NUMBER IN $F^{\otimes 7}$ 79

LIST OF FIGURES

FIGURE 1.1 BLOCK ERROR RATE (BLER) PERFORMANCE OF $(1024, K)$ POLAR CODES UNDER A BINARY ERASURE CHANNEL WITH ERASURE RATE 0.5 (REPRODUCED FROM [CHEN, NIU, & LIN, 2012]).....4
FIGURE 2.1 BINARY ERASURE CHANNEL BEC(ϵ).....11
FIGURE 2.2 BASIC CHANNEL COMBINING.....12
FIGURE 2.3 $W_2^{(1)}$ CHANNEL AFTER BASIC CHANNEL SPLITTING.....13
FIGURE 2.4 $W_2^{(2)}$ CHANNEL AFTER BASIC CHANNEL SPLITTING.....14
FIGURE 2.5 W_4 CHANNEL BY COMBINING TWO INDEPENDENT COPIES OF W_2.....15
FIGURE 2.6 W_N CHANNEL BY COMBINING TWO INDEPENDENT COPIES OF $W_{\frac{N}{2}}$.....17
FIGURE 2.7 THE SYNTHESIZED CHANNEL W_N IS SPLIT INTO N DISTINCT BIT-CHANNELS $W_N^{(i)}$.....18
FIGURE 2.8 THE SYNTHESIZED CHANNEL W_4 21
FIGURE 2.9 THE SYNTHESIZED CHANNEL W_4 IS SPLIT INTO 4 BIT-CHANNELS $W_4^{(i)}$.....21
FIGURE 2.10 CHOOSING THE INFORMATION BITS FOR AN $(8, 4)$ POLAR CODE OVER BEC(0.35), AND ENCODING OF THE WORD [1,1,1,0].....25
FIGURE 2.11 REDRAWN REFERENCE FACTOR GRAPH REPRESENTATIONS OF FIGURE 2.10 WITH Z-SHAPE CONNECTIONS.....26
FIGURE 2.12 REFERENCE FACTOR GRAPH WITH CHECK NODES (SQUARE) AND VARIABLE NODES (CIRCLE) FOR $N = 8$.....27
FIGURE 2.13 DECODING PROCESS DETAILS OF A Z-SHAPE CONNECTIONS.....28
FIGURE 3.1 CHANNEL CAPACITIES OF THE REFERENCE FACTOR GRAPH (3-2-1) FOR THE $(8, K)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....34
FIGURE 3.2 INPUT BIT CAPACITIES AND CS OF ALL FACTOR GRAPHS FOR THE $(8, 3)$ AND $(8, 4)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....35
FIGURE 3.3 Z-SHAPE PROCESS.....36
FIGURE 3.4 INFORMATION AND FROZEN VARIABLES OF ALL FACTOR GRAPHS FOR THE $(8, 4)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....37
FIGURE 3.5 INFORMATION AND FROZEN VARIABLES OF ALL FACTOR GRAPHS FOR THE $(8, 3)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....38
FIGURE 3.6 RELATION BETWEEN FV AND CS VALUES OF FACTOR GRAPHS FOR THE $(64, 32)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....45
FIGURE 3.7 FV VERSUS SON VALUES OF ALL FACTOR GRAPHS FOR THE $(64, 32)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....46
FIGURE 3.8 CS VERSUS SON VALUES OF ALL FACTOR GRAPHS FOR THE $(64, 32)$ ADAPTIVE POLAR CODE OVER BEC (0.35).....47

FIGURE 3.9 CODEWORD ERROR RATIO (CER) PERFORMANCE VERSUS THE NUMBER OF FROZEN VARIABLES FOR SINGLE-FG BP DECODER OF THE (64, 32) ADAPTIVE POLAR CODE OVER BEC(0.35). 49
FIGURE 3.10 CODEWORD ERROR RATIO (CER) PERFORMANCE VERSUS CAPACITY SUM AT SINGLE-FG BP DECODER FOR THE (64, 32) ADAPTIVE POLAR CODE OVER BEC(0.35). 49
FIGURE 3.11 CODEWORD ERROR RATIO (CER) PERFORMANCE VERSUS STAGE ORDER NUMBER (SON) AT SINGLE-FG BP DECODER FOR THE (64, 32) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 51
FIGURE 3.12 CODEWORD ERROR RATIO (CER) COMPARISON OF THE INDEPENDENT AND DEPENDENT 7-FG BELIEF PROPAGATION DECODERS FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 53
FIGURE 3.13 CODEWORD ERROR RATIO (CER) COMPARISON OF THE INDEPENDENT AND DEPENDENT 4-FG BELIEF PROPAGATION DECODERS IN 10 TRIALS FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 54
FIGURE 3.14 CODEWORD ERROR RATIO (CER) COMPARISON OF THE DEPENDENT 7-FG BP DECODERS WITH CYCLICALLY RIGHT AND LEFT-SHIFTED RFG SETS FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 56
FIGURE 3.15 CODEWORD ERROR RATIO (CER) COMPARISON OF THE DEPENDENT 4-FG BP DECODERS WITH A FIXED SET OF 4 FG'S IN 24 DIFFERENT ORDERS FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 57
FIGURE 3.16 CODEWORD ERROR RATIO (CER) COMPARISON OF 6-FG AND TWICE REPEATED 3-FG DEPENDENT BP DECODERS FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 58
FIGURE 3.17 (A) NUMBER OF FROZEN VARIABLES (FV) AND (B) CAPACITY SUM (CS) COMPARISON OF 6-FG'S AND TWICE REPEATED 3-FG'S FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35); LOWER CURVES CORRESPONDING TO 6-FG'S..... 59
FIGURE 3.18 CODEWORD ERROR RATIO (CER) COMPARISON OF 8-FG AND TWICE REPEATED 4-FG DEPENDENT BP DECODERS FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35)..... 60
FIGURE 3.19 CODEWORD ERROR RATIO (CER) PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35) OF THREE 7-FG DECODER SETS, *I)* GREEN CURVE: 7 TOP FG'S OF THE SON-SORTED LIST, *II)* RED CURVE: THE HIGHEST CS-VALUED FG OF 7 HIGHEST EQUI-FV GROUPS, AND *III)* BLUE CURVE: 7 FG'S GENERATED BY CYCLIC SHIFTS; I.E., THE CYCLIC SET. 62
FIGURE 3.20 (A) NUMBER OF FROZEN VARIABLES (FV) AND (B) CAPACITY SUM (CS) COMPARISON OF THE THREE SETS IN FIGURE 3.19 FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35); *I)* GREEN CURVE: 7 TOP FG'S OF THE SON-SORTED LIST, *II)* RED CURVE: THE HIGHEST CS-VALUED FG OF 7 HIGHEST EQUI-FV GROUPS, AND *III)* BLUE CURVE: 7 FG'S GENERATED BY CYCLIC SHIFTS; I.E., THE CYCLIC SET. 63
FIGURE 3.21 CODEWORD ERROR RATIO (CER) PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35) OF TWO DECODER SETS, *I)* RED CURVE: CYCLIC SET WITH 7 FG'S, *II)* PURPLE CURVE: 4 FG'S STARTING WITH ODD-NUMBERED STAGES, CHOSEN FROM THE CYCLIC SET..... 65
FIGURE 3.22 CODEWORD ERROR RATIO (CER) PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35) OF TWO DECODER SETS, *I)* GREEN CURVE: TRIAL SET WITH 4 FG'S "7654321-5764321-

3765421-1765432", *II)* PURPLE CURVE: 4-FG CYCLIC SUBSET STARTING WITH ODD-NUMBERED STAGES, CHOSEN FROM THE CYCLIC SET 68

FIGURE 3.23 CODEWORD ERROR RATIO (CER) PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.35) OF TWO DECODER SETS, *I)* PINK CURVE (UPPER ONE): 4-FG CYCLIC SUBSET, *II)* BLUE CURVE (LOWER ONE): 4-FG SET PRODUCED BY THE SET CHOICE ALGORITHM. 70

FIGURE 3.24 CODEWORD ERROR RATIO (CER) PERFORMANCE FOR THE (64, 32) ADAPTIVE POLAR CODE OVER BEC(0.35) OF TWO DECODER SETS, *I)* BLUE CURVE (LOWER ONE): 4-FG SET PRODUCED BY OUR SET CHOICE ALGORITHM, *II)* RED CURVE (UPPER ONE): GENIE-CHOSEN SET OF [DOĞAN, 2015] 71

FIGURE 3.25 CODEWORD ERROR RATIO PERFORMANCE FOR THE (64, 32) ADAPTIVE POLAR CODE OVER BEC(0.4) OF THREE M-FG DECODER SETS, *I)* RED CURVE: 6 FG'S BY CYCLICALLY LEFT-SHIFTING STAGES OF THE RFG, *II)* PINK CURVE: 3 FG'S SAMPLED FROM THE FIRST SET, *III)* BLUE CURVE: 3 FG'S BY THE SET CHOICE ALGORITHM. 73

FIGURE 3.26 CODEWORD ERROR RATIO PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER BEC(0.4) OF THREE M-FG DECODER SETS, *I)* RED CURVE: 7 FG'S BY CYCLICALLY LEFT-SHIFTING STAGES OF THE RFG, *II)* PINK CURVE: 4 FG'S SAMPLED FROM THE FIRST SET, *III)* BLUE CURVE: 4 FG'S BY THE SET CHOICE ALGORITHM. 73

FIGURE 3.27 CODEWORD ERROR RATIO PERFORMANCE FOR THE (256, 128) ADAPTIVE POLAR CODE OVER BEC(0.4) OF THREE M-FG DECODER SETS, *I)* RED CURVE: 8 FG'S BY CYCLICALLY LEFT-SHIFTING STAGES OF THE RFG, *II)* PINK CURVE: 4 FG'S SAMPLED FROM THE FIRST SET, *III)* BLUE CURVE: 4 FG'S BY THE SET CHOICE ALGORITHM. 74

FIGURE 3.28 CODEWORD ERROR RATIO PERFORMANCE FOR THE (512, 256) ADAPTIVE POLAR CODE OVER BEC(0.4) OF THREE M-FG DECODER SETS, *I)* RED CURVE: 9 FG'S BY CYCLICALLY LEFT-SHIFTING STAGES OF THE RFG, *II)* PINK CURVE: 5 FG'S SAMPLED FROM THE FIRST SET, *III)* BLUE CURVE: 5 FG'S BY THE SET CHOICE ALGORITHM. 74

FIGURE 3.29 CODEWORD ERROR RATIO PERFORMANCE FOR THE (1024, 512) ADAPTIVE POLAR CODE OVER BEC(0.4) OF THREE M-FG DECODER SETS, *I)* RED CURVE: 10 FG'S BY CYCLICALLY LEFT-SHIFTING STAGES OF THE RFG, *II)* PINK CURVE: 5 FG'S SAMPLED FROM THE FIRST SET, *III)* BLUE CURVE: 5 FG'S BY THE SET CHOICE ALGORITHM. 75

FIGURE 3.30 CODEWORD ERROR RATIO PERFORMANCE FOR THE (2048, 1024) ADAPTIVE POLAR CODE OVER BEC(0.4) OF THREE M-FG DECODER SETS, I) RED CURVE: 11 FG'S BY CYCLICALLY LEFT-SHIFTING STAGES OF THE RFG, II) PINK CURVE: 6 FG'S SAMPLED FROM THE FIRST SET, III) BLUE CURVE: 6 FG'S BY THE SET CHOICE ALGORITHM. 75

FIGURE 3.31 CODEWORD ERROR RATIO PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER DIFFERENT BEC(ε) OF FOUR DECODERS, *I*) ORANGE CURVE: SINGLE-FG WITH SON=1234567, *II*) GREEN CURVE: THE RFG (7654321), *III*) PINK CURVE: 4-FG CYCLIC SUBSET, *IV*) BLUE CURVE: 4-FG SET PRODUCED BY THE SET CHOICE ALGORITHM. 81

FIGURE 3.32 BIT ERROR RATIO PERFORMANCE FOR THE (128, 64) ADAPTIVE POLAR CODE OVER DIFFERENT BEC(ε) OF FOUR DECODERS, *I*) ORANGE CURVE: SINGLE-FG WITH SON=1234567, *II*) GREEN CURVE: THE

RFG (7654321), *III*) PINK CURVE: 4-FG CYCLIC SUBSET, *IV*) BLUE CURVE: 4-FG SET PRODUCED BY THE SET CHOICE ALGORITHM. 81

FIGURE 3.33 AVERAGE NUMBER OF BP DECODING ITERATIONS OVER 1000 CODEWORDS WITH PERFECT KNOWLEDGE-BASED STOPPING CONDITION FOR THE (128, 64) ADAPTIVE POLAR CODE OVER DIFFERENT BEC(E) OF FOUR DECODERS, *I*) ORANGE CURVE: SINGLE-FG WITH SON=1234567, *II*) GREEN CURVE: THE RFG (7654321), *III*) PINK CURVE: 4-FG CYCLIC SUBSET, *IV*) BLUE CURVE: 4-FG SET PRODUCED BY THE SET CHOICE ALGORITHM. 83

LIST OF ABBREVIATIONS

B-DMC	……	Binary-input Discrete Memoryless Channel
BEC	……	Binary Erasure Channel
BER	……	Bit Error Ratio
BLER	……	Block Error Ratio
BP	……	Belief Propagation
BPL	……	Belief Propagation List
BSC	……	Binary Symmetric Channel
CER	……	Codeword Error Ratio
CRC	……	Cyclic Redundancy Check
CS	……	Capacity Sum
FG	……	Factor Graph
FV	……	Number of Frozen Variables
LDPC	……	Low Density Parity Check
M-FG	……	Multiple Decoder That Uses M Factor Graphs
MPA	……	Message Passing Algorithm
MS	……	Min-Sum
PC	……	Polar Codes
RFG	……	Reference Factor Graph
RM	……	Reed-Muller
SC	……	Successive Cancellation
SCL	……	Successive Cancellation List
SMS	……	Scaled Min-Sum
SON	……	Stage Order Number

CHAPTER 1

INTRODUCTION

1.1 Channel Coding

In the transmission of information, reliable data delivery from source to destination is one of the most fundamental aims. In accordance with this purpose, it is essential to down the channel noise occurs in the unreliable communication channels. Error detection techniques can determine the errors caused by the noise and error correction techniques can reconstruct these errors in data. The developments of error detection and correction are inspired from the seminal work of Claude Shannon's A Mathematical Theory of Communication [Shannon, 1948].

In 1948, Shannon stated the relationship between channel capacity as C and transmission rate as R, in his two principal theorem; source coding and noisy channel coding. Source coding, can also be stated as data compression, explains the wrapping of source data in order to remove uncontrolled redundancy. On the other hand, channel coding uses these redundant bits properly during the transmission, and at the receiver, decompress the data with checking these redundant bits. According to his theorems, if the transmission rate R is below the capacity C; i.e., $R < C$, it is possible that the receiver has arbitrarily small error rate with appropriate coding. Besides, if the opposite occurs, reliable transmission may not be achievable since the probability of error is greater than a certain minimal level.

Since then, by using Shannon’s work, finding the linear codes with good algebraic qualifications was an imperative research area for coding theory. Algebraic codes such as Hamming [Hamming, 1950], Golay [Golay, 1949], Reed-Muller [Muller,

1954], [Reed, 1954], Reed-Solomon [Reed & Solomon, 1960] and BCH codes [Bose & Ray-Chaudhuri, 1960] have been studied and expressed in algebraic terms. On the other hand, there are probabilistic codes including convolutional codes [Elias, 1955], LDPC codes [Gallager, 1963] and Turbo codes [Berrou, Glavieux, & Thitimajshima, 1993]. With these studies, more reliable data transmission has become possible. But none of these codes have performance as close to the Shannon's capacity as that of the LDPC and Turbo codes.

Low-Density Parity-Check (LDPC) codes were invented by Gallager in 1962 [Gallager, 1962]. However, due to technological inadequacies like the lack of a practical decoding algorithm, practical implementations could not be possible until 1996. In 1996, MacKay and Neal presented a special form of Gallager's LDPC codes with a low complexity iterative decoder [MacKay & Neal, 1996]. A few years earlier, in 1993, *"Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes"* were presented [Berrou, Glavieux, & Thitimajshima, 1993]. Developments in computer technology have also supported Berrou's study about bipartite graph-based decoding algorithms. Turbo codes have been revolutionary since their transmission rate is quite close to the capacity limit given by Shannon's theory. Other Turbo code schemes, which need half power for decoding, were also proposed [Berrou & Glavieux, 1996].

In 2009, Arıkan introduced polar codes [Arıkan, 2009] as the first "provably capacity achieving" linear block codes. In the area of source coding and secrecy coding, polar codes have almost optimal performance [Hassani, Korada, & Urbanke, 2009] [Mahdavifar & Vardy, 2011]. Another thing that makes this theory so unique is that the encoding and decoding complexities of polar codes are low, $O(N \log N)$, where N is the blocklength of the codes.

Polar codes are constructed by channel polarization, which consists of channel combining and channel splitting steps. Channel combining step generates a vector channel from N copies of binary-input discrete memoryless channel (B-DMCs). Channel splitting step obtains N polarized new channels, whose capacities vary in the range [0, 1], from a vector channel produced with the channel combining step

[Arıkan, 2009]. While the blocklength is getting larger, capacities of polarized new channels converge to 0 or 1 with the channel transformation operations. The main idea in constructing an (N, K) polar code for blocklength $N = 2^n$ is choosing K channels with higher capacities to send data, and freezing the remaining $N - K$ channels. Then, inputs are mapped to codewords by using a $K \times N$ generator matrix, which is composed of some rows of $F^{\otimes n}$, the n^{th}-Kronecker product of the base matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Arıkan also presented a formula for recursive computation of Bhattacharyya parameters over binary erasure channels [Arıkan, 2009]. Zhao, Shi and Wang worked on the computation of Bhattacharyya parameters [Zhao, Shi, & Wang, 2011], presenting the recursive formula with initial values for four conventional channels to provide polar codes with converging capacities.

1.2 Belief Propagation Decoder

Polar codes are usually decoded by successive cancellation (SC) and belief propagation (BP) decoding algorithms. Arıkan has introduced the SC decoding algorithm for polar codes [Arıkan, 2009]. However; due to the serial decoding scheme of the SC algorithm, it may suffer from high decoding latency. Forney has shown that Reed-Muller (RM) codes can be regarded as codes on graphs [Forney, 2001]. (N, K) polar codes where $N = 2^n$ are subcodes of the full RM(n, n) codes, so they can also be seen as codes on graphs. Hence, they can be decoded by iterative BP decoders, which may be implemented with lower decoding latency than SC decoders because of their parallelizability. Arıkan uses the BP decoding algorithm to show the performance advantages of polar codes over Reed-Muller codes [Arıkan, 2008]. Hussami, Korada and Urbanke report that polar codes perform better with the BP algorithm than the SC algorithm [Hussami, Korada, & Urbanke, 2009]. Korada also observes that using the BP decoder with a multiple trellis, which is obtained by $n = \log_2 N$ cyclic stage permutations of the encoding factor graph, provides significant improvement on the performance of polar codes [Korada, 2009].

Today's popular decoding algorithm is list decoding; proposed by Dumer and Shabunov for the Reed-Muller codes [Dumer & Shabunov, 2006]. Tal and Vardy have properly applied successive cancellation list decoding to polar codes [Tal & Vardy, 2011]. They describe it as the generalization of the classic SC decoder of Arıkan. Also; Chen, Niu and Lin worked with list decoding [Chen, Niu, & Lin, 2012]. They compare five different decoding algorithms for polar codes with $N = 1024$ and different code rates over a binary erasure channel. They use successive cancelation (SC), belief propagation (BP), linear programming (LP), maximum likelihood (ML) and successive cancellation list (SCL) decoding. SCL is performed with $L = 20$ and $T = 0.1$. As seen in Figure 1.1, performance of the SCL (or equivalently LSC; i.e., "list SC") is almost as good as the maximum-likelihood (ML) decoding performance.

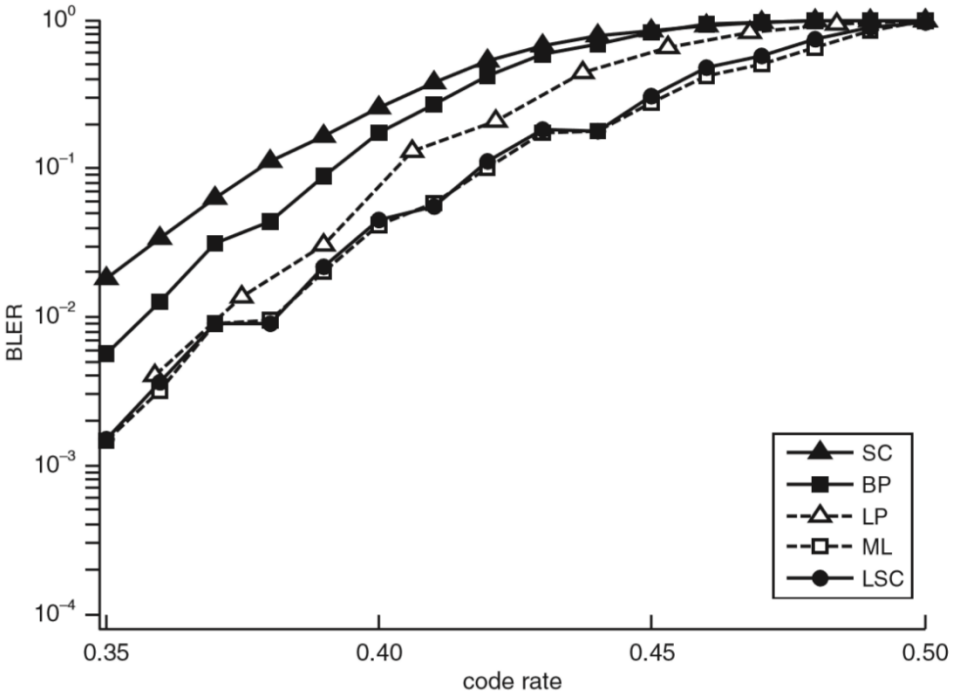


Figure 1.1 Block error rate (BLER) performance of (1024, *K*) polar codes under a binary erasure channel with erasure rate 0.5 (reproduced from [Chen, Niu, & Lin, 2012]).

Eslami and Pishro-Nik have performed simulations that indicate polar codes, as compared to LDPC codes, have inferior bit error rate but superior error floor

performance. They have suggested a guessing algorithm to improve the performance of BP decoding by 0.2 dB at BER= 10^{-5} over the Gaussian channel with polar code length 2^{13} [Eslami & Pishro-Nik, 2010]. In 2013, the same authors have proposed a concatenated polar-LDPC code of length around 2^{15}, which uses a soft BP decoder with 9 bit quantization, that closes the gap to Shannon's capacity while avoiding the error floor and maintaining low complexity [Eslami & Pishro-Nik, 2013].

Pamuk has stated that calculations for message-passing algorithm can be approximated to a min-sum (MS) method at the cost of some degradation at the performance [Pamuk, 2011]. Yuan and Parhi have examined the min-sum approximation and proposed scaled min-sum (SMS) BP decoding algorithm to eliminate the loss at performance of polar codes due to min-sum (MS) approximation [Yuan & Parhi, 2013]. Furthermore; Xu, Che and Choi have proposed a novel round-trip message passing scheduling method by simplifying the scaled min-sum approximation at BP decoding to reduce the computational complexity [Xu, Che, & Choi, 2015]. They have suggested that instead of updating nodes with left and right messages in each stage, firstly pass all left-directed messages and then all right-directed messages to update nodes. Their algorithm provides the same performance as the scaled min-sum (SMS) BP decoding algorithm with lower complexity and less number of iterations. Doğan has studied on the required iteration numbers for belief propagation [Doğan, 2015] and suggested maximum iteration numbers for different blocklengths to optimize the complexity and performance. Şimşek and Türk have recommended early stopping methods for BP decoders to reduce the computational complexity of successful decoding and the number of operations during the decoding process [Şimşek & Türk, 2016]. A soft-output decoder for polar codes, called SCAN, has been offered by Fayyaz and Barry, to improve the performance with lower complexity and memory requirements [Fayyaz & Barry, 2014]. Lin, Xiong and Yan have presented their reduced complexity soft cancellation decoding algorithm, called the RCSC. Their decoder architecture consumes less energy on updating log-likelihood-ratio (LLRs) [Lin, Xiong, & Yan, 2015]. However; both SCAN and RCSC decoding algorithms have limited throughput due to their serial decoding schemes. For higher throughput, Lin, Sha, Li, Xiong, Yan and Wang have suggested an improved decoding algorithm [Lin, et al., 2016]. In another study, the memory

efficient BP decoder algorithm for polar codes has been proposed, by combining two adjacent stages into one stage and offering new belief message updating rules, in order to get lower latency and memory requirements [Sha, Liu, Wang, & Zeng, 2015]. Other researchers have offered a modified BP decoding algorithm for polar codes [Zhang, Liu, Pan, Ye, & Gong, 2014]. They have used extra check nodes to multiply the messages, and shown that the reliability of propagated messages are enhanced while the performance is improved.

As in [Korada, 2009], Doğan and Peker also study on the subject of permuted factor graphs (FG's) under BP decoding, and attempt to find suitable FG sets [Doğan, 2015] [Peker, 2018]. In [Elkelesh, Ebada, Cammerer, & Brink, 2018-1], the BP decoder randomly permutes the FG stages (to arrive as many as 1000 different FG's) of the (2048, 1024) polar codes. Using a perfect knowledge-based stopping criterion, the proposed multiple trellis decoder slightly outperforms the SCL decoder with list size 32, but it cannot catch the performance of the SCL-CRC decoder with cyclic redundancy check (CRC) aided stopping criterion. In another paper, the same researchers investigate the BP decoding performance of L (=32 or 64) permuted FG's, which decode independently in a parallel decoding system [Elkelesh, Ebada, Cammerer, & Brink, 2018-2]. With a stopping criterion based on the generator matrix G, parallel BP list decoder (i.e., BPL decoder) reaches almost the same performance as the SCL decoder with a lower latency; but again, it cannot catch the performance of the SCL+CRC-16 that uses the cyclic redundancy check.

It is also interesting that the (2048, 1024) hybrid RM-polar code (with minimum distance 32) suggested in [Li, Shen, & Tse, 2014] outperforms the (2048, 1024) polar code (with minimum distance 16). BPL performances of both codes are below their SCL performances [Elkelesh, Ebada, Cammerer, & Brink, 2018-2].

Choice of factor graphs for the decoding set among all possible permutations are substantial, especially if one prefers to deal with small number of permuted FG's. In [Doan, Hashemi, Mondelli, & Gross, 2018], an empirical method is suggested to compose a small sized set of FG's among $k!$ different factor graphs, which are produced by permuting the last k stages of the reference factor graph. Also, by using the relation between stage permutations and bit permutations of the binary row

representations, they claim that a single decoding scheduling can be applied to different FG's without making changes in the hardware implementations. Permuted factor graph decoding is also investigated by [Hashemi, Doan, Mondelli, & Gross, 2018]. They discuss that successive factor graph permutations improve the decoding performance of Reed-Muller (RM) and polar codes under SC and list decoding if suitable FG permutations are chosen.

There are additional methods to improve the performance of polar codes such as concatenating polar codes with other codes, for example with chosen optimal linear block codes in [Trifonov, 2012] and outer LDPC codes in [Guo, Qin, Fabregas, & Siegel, 2014]. A scattered EXIT chart-based design method is presented to develop the performance of the concatenated polar and LDPC codes in [Elkelesh, Ebada, Cammerer, & Brink, 2016]. Concatenating the belief propagation decoding algorithm with other algorithms to generate a hybrid decoder, is another way to improve the performance of polar codes as explained in [Yuan & Parhi, 2014] and [Cammerer, Leible, Stahl, Hoydis, & Brink, 2017]. By pruning some check and variable nodes of the decoder, size of the parity check matrix is reduced in order to interpret a polar code as an LDPC code with an underlying sparse decoding graph in [Cammerer, Ebada, Elkelesh, & Brink, 2018]. This method provides reduced memory requirements and complexity with negligible performance loss, thus it is suitable for hardware implementations.

1.3 Aim and Organization of the Thesis

In this thesis, single factor graph and multiple factor graph performances of (N, K) polar codes under the belief propagation decoding algorithm are examined. The effects of permuting the stages of the decoder factor graph over the capacity sum (sum of capacities of the information bit channels), the number of frozen variables and over the performance of polar codes are studied. Among $(\log_2 N)! = n!$ possible stage permutations of the factor graphs, suitable sets of stage-permuted factor graphs

are sought experimentally, to improve the performance of multiple trellis BP decoding by keeping the set size as small as $\lceil n/2 \rceil$. All simulations are carried on a binary erasure channel BEC(ε), with a given erasure rate ε; and a perfect knowledge-based stopping criterion is used in all decoders.

In Chapter 2; a summary of polar codes is given. Main ideas of polar codes, channel polarization, channel combining and splitting, are reviewed. Procedure for capacity calculation over a given binary erasure channel is shown. Relation between bit capacities and Bhattacharyya parameters is examined. To differentiate the factor graphs, Z-shaped stages of $n!$ diagrams are named as 1 to $n = log_2 N$, where N is the block length. At the end of the chapter, belief propagation decoding algorithm is explained.

In the first part of Chapter 3; calculation of the "capacity sum - CS" and the "number of frozen variables - FV" parameters for polar codes and Reed-Muller codes are explained. The difference of polar and Reed-Muller codes in terms of these parameters, and the change of these parameters for both codes under stage permutations, are examined. Moreover; variation of CS and FV parameters versus the stage order numbers (SON) is explored for $n!$ factor graphs. Performances of single factor graph polar code BP decoders are obtained depending on their CS, FV and SON values. In the second part of Chapter 3; multiple factor graph (M-FG, or multi-trellis) decoding methods are examined. Difference between dependent and independent M-FG decoding methods is discussed. Selection of FG sets for a better-performing M-FG decoder is explored, where the SON values of the factor graphs are also considered, in search for a systematic rule. Moreover, the answer for the question of getting similar performance with less number of factor graphs is sought. An algorithm that generates an FG set with $M = \lceil n/2 \rceil$ elements is presented; and the performance comparison of $\lceil n/2 \rceil$ FG's generated by this algorithm, with Korada's n cyclically shifted FG's is given for polar codes of different lengths. Dependence of the construction and performance of polar codes upon the channel erasure rate is investigated.

In Chapter 4, main contributions of this thesis are discussed and summarized.

CHAPTER 2

REVIEW OF POLAR CODES

In this chapter, we give a review of polar code construction for the binary-input discrete memoryless channel (B-DMCs) based on the work of Arıkan [Arıkan, 2009]. Firstly, we present some preliminary information. Recursive channel transformation to get N distinct channels from N copies of a given binary-input discrete memoryless channel (B-DMC) is explained after basic channel combining and splitting operations are summarized in Section 2.2. In Section 2.3, channel polarization is described. Method of calculation for capacities of each polarized channel by using the channel erasure rate is clarified in Section 2.4. In Section 2.5, the process of determining the information channels that send data, and constructing the generator matrix depending on the information transmission channels is described in detail. Also; we provide the simple factor graph representation that is used both in encoding and decoding operations. Finally, in Section 2.6, Belief Propagation (BP) decoding with a modification that we employ, is described briefly.

2.1 Preliminaries

Representation of $W: X \rightarrow Y$ is used for a B-DMC, where X is input alphabet, Y is output alphabet and $W(y|x)$ is channel transition probability for $x \in X, y \in Y$. We assume the input alphabet set X to be $\{0, 1\}$. $W_N: X^N \rightarrow Y^N$ with transition probability $W_N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i)$ represents the channel, which is N times employments of W. A row vector $(u_i, u_{i+1}, \ldots, u_j)$ is described as u_i^j for $1 \leq i \leq j \leq N$. Also; all calculations are over $GF(2)$.

The symmetric mutual information of a B-DMC $W: X \rightarrow Y$ is

$$I(W) = \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}\left(W(y|0)W(y|1)\right)} \tag{2.1}$$

The Bhattacharya parameter is used for measurement of the reliability of the channel and defined as [Arıkan, 2009]

$$Z(W) = \sum_{y \in Y} \sqrt{W(y|0)W(y|1)} \tag{2.2}$$

The Kronecker product of an $m \times n$ matrix $A = [A_{ij}]$ and a $k \times l$ matrix $B = [B_{ij}]$ is defined as

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix} \tag{2.3}$$

$A^{\otimes n}$ is used to denote Kronecker product of the matrix A by itself n times.

As known in the literature, there are many different channels but, we will use only the binary erasure channel (BEC) in this work. BEC is one of the simplest channel to the analysis. This channel is binary input channel that means only one (0 or 1) input symbol can be transmitted. Since the channel is not perfect channel, there may be erased bits through the transmission. As shown in **Figure 2.1**, the bits can be either erased with the probability ϵ or correctly transmitted with probability $1 - \epsilon$. Abbreviation of the channel is BEC(ϵ) and capacity of it is $1 - \epsilon$.

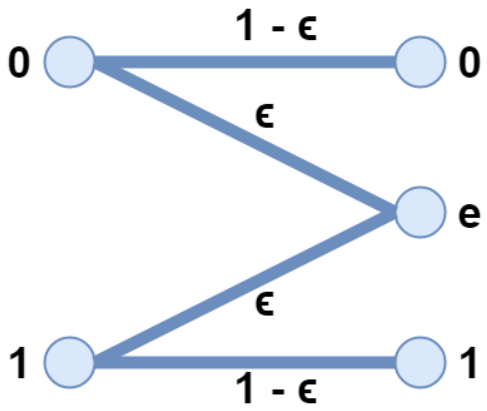


Figure 2.1 Binary erasure channel BEC(ϵ).

2.2 Channel Transformation

Polar codes are constructed by a channel transformation method. This method has two parts, which are channel combining and channel splitting. N copies of a given binary-input discrete memoryless channel (B-DMC) W are transformed into N binary input channels ($W_N^{(i)}: 1 \leq i \leq N$). $W_N^{(i)}$ is defined for the i^{th} bit-channel. Parts of channel transformation will be studied in detail in the following subsections.

2.2.1 Basic Channel Combining

Channel combining has recursive steps in order to produce a vector channel called W_N, where N is the blocklength. In the first step, two independent B-DMCs W are combined to obtain $W_2: X^2 \rightarrow Y^2$ by applying the base matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. This is the basic channel combining in the transformation. A row vector $X = (x_1, x_2)$ is input and $Y = (y_1, y_2)$ is output for two independent B-DMCs W. It can be defined as $x_1^2 = u_1^2 F^{\otimes 1}$. This first step is shown in **Figure 2.2**.

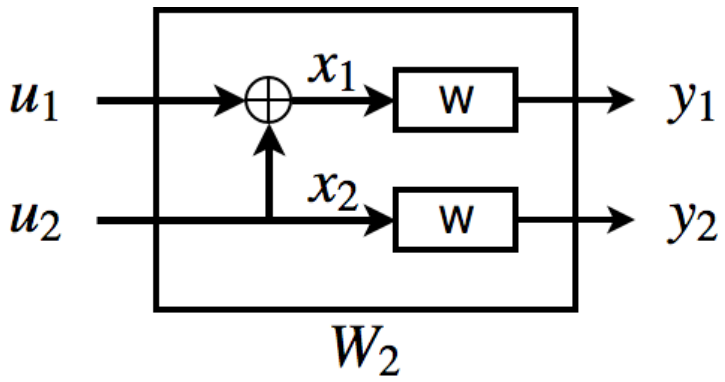


Figure 2.2 Basic Channel Combining.

The transition probability of the vector channel W_2 is

$$W_2(y_1^2|x_1^2) = \prod_{i=1}^{2} W(y_i|x_i) = W(y_1|u_1 \oplus u_2)W(y_2|u_2) \qquad 2.4$$

Since linear transform between u_1^2 and x_1^2 has one-to-one mapping property, when u_i is identically independent distributed, x_i is also identically independent distributed. Thus; there is no loss in total channel capacity as described in Equation 2.5.

$$I(W_2) = I(U_1, U_2; Y_1, Y_2) = I(X_1, X_2; Y_1, Y_2) = I(X_1; Y_1) + I(X_2; Y_2) \qquad 2.5$$
$$= 2I(W)$$

2.2.2 Basic Channel Splitting

Channel splitting has also recursive steps in order to get polarized channels from a vector channel called W_N, where N is the blocklength. Basic channel splitting method is explained for B-DMCs W_2, which has two independent B-DMCs W. As mentioned in Equation 2.5, the mutual information between the input and the output of the W_2 channel is $I(W_2) = I(U_1, U_2; Y_1, Y_2)$. This mutual information can be split as

$$I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2, U_1) \quad 2.6$$

A new channel can be defined by assuming u_1 as input, y_1, y_2 as outputs and u_2 as noise. This can be shown as $W_2^{(1)}: X \to Y^2$ whose mutual information is $I(U_1; Y_1, Y_2)$. The transition probability of the vector channel $W_2^{(1)}$ is indicated in Equation 2.7 and the channel is demonstrated in **Figure 2.3**.

$$W_2^{(1)}(y_1, y_2|u_1) = \sum_{u_2} \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2) \quad 2.7$$

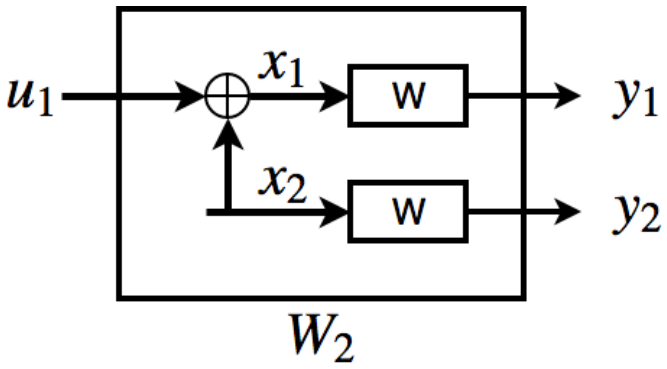


Figure 2.3 $W_2^{(1)}$ channel after basic channel splitting.

Another new channel can also be defined by using u_1, y_1, y_2 as outputs and u_2 as input. This can be indicated with $W_2^{(2)}: X \to X \times Y^2$ whose mutual information is $I(U_2; Y_1, Y_2, U_1)$. The transition probability of the vector channel $W_2^{(2)}$ is indicated in Equation 2.8 and the channel is shown in **Figure 2.4**.

$$W_2^{(2)}(y_1, y_2, u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2) \quad 2.8$$

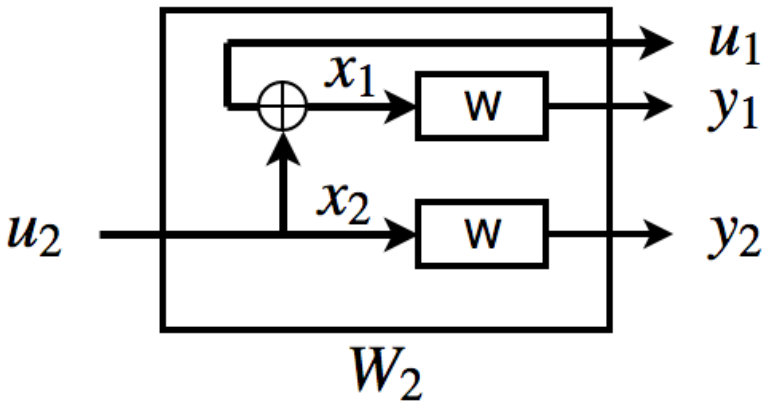


Figure 2.4 $W_2^{(2)}$ channel after basic channel splitting.

The mutual information of the new channels $W_2^{(1)}$ and $W_2^{(2)}$ satisfy Equation 2.9 and equality is achieved if and only if mutual information of $W$is 0 or 1. Mutual information of these new channels converges to 0 or 1, while total capacity is conserved. That is; while blocklength is getting larger, polarization becomes more.

$$I(W_2^{(1)}) \leq I(W) \leq I(W_2^{(2)}) \qquad 2.9$$

2.2.3 Recursive Channel Combining and Splitting

So far, we have mentioned about basic channel combining and splitting methods over the B-DMCs W_2. In this section, for larger N values where $N = 2^n$, necessary steps to obtain the distinct bit channels $W_N^{(i)}$will be discussed by using channel combining and splitting methods recursively.

In the second step of the recursion, for $N = 4$, a $W_4: X^4 \rightarrow Y^4$ channel with four inputs and four outputs is produced by combining two independent copies of B-DMCs W_2. The W_4 channel is indicated in Figure 2.5, and its transition probability is shown in Equation 2.10.

$$W_4(y_1^4|u_1^4) = \prod_{i=1}^{4} W(y_i|x_i) \tag{2.10}$$
$$= W_2(y_1, y_2|u_1 \oplus u_3, u_2 \oplus u_4) W_2(y_3, y_4|u_3, u_4)$$
$$= W(y_1|u_1 \oplus u_2 \oplus u_3 \oplus u_4) W(y_2|u_2 \oplus u_4) W(y_3|u_3 \oplus u_4) W(y_4|u_4)$$

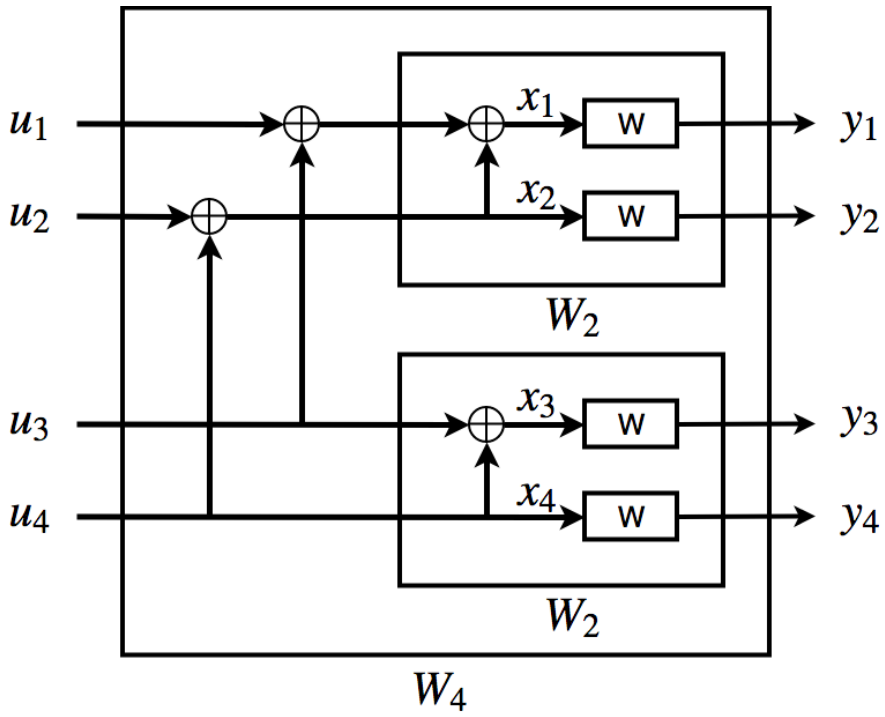


Figure 2.5 W_4 channel by combining two independent copies of W_2.

Two independent B-DMCs W_2 are combined to obtain W_4: $X^4 \rightarrow Y^4$ by applying the 2nd-Kronecker product ($F^{\otimes 2}$) of the base matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, where $N = 4 = 2^n = 2^2$. This is the second step of the channel combining process in the transformation. The mapping over W_4 channel can be represented as $x_1^4 = u_1^4 F^{\otimes 2}$ where $F^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$.

For larger $N = 2^n$ values, two independent copies of B-DMCs $W_{\frac{N}{2}}$ are recursively combined to produce $W_N: X^N \rightarrow Y^N$ with N inputs and N outputs. The combined general W_N channel is indicated in **Figure 2.6** and its general transition probability is shown in Equation 2.11.

$$W_N(y_1^N|u_1^N) = W_{\frac{N}{2}}\left(y_1^{\frac{N}{2}}\middle|u_1^{\frac{N}{2}} \oplus u_{\frac{N}{2}+1}^N\right) W_{\frac{N}{2}}\left(y_{\frac{N}{2}+1}^N\middle|u_{\frac{N}{2}+1}^N\right) \qquad 2.11$$

$$where\ u_1^{\frac{N}{2}} = \left\{u_1, u_2, \dots, u_{\frac{N}{2}}\right\}, u_{\frac{N}{2}+1}^N = \left\{u_{\frac{N}{2}+1}, u_{\frac{N}{2}+2}, \dots, u_N\right\}$$

The overall mapping, which is between inputs u_1^N of W_N channel and inputs x_1^N of W channel, can be defined as $x_1^N = u_1^N F^{\otimes n}$ where $F^{\otimes n}$ is the n^{th}-Kronecker product of the base matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $n = \log_2 N$.

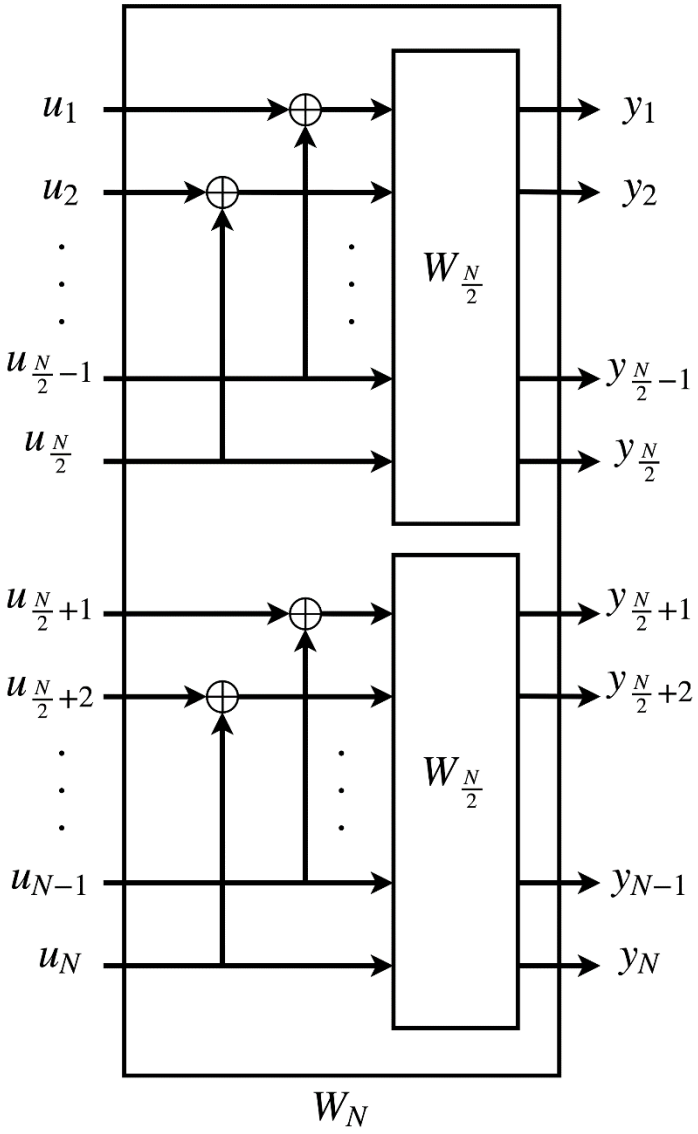


Figure 2.6 W_N channel by combining two independent copies of $W_{\frac{N}{2}}$.

After producing the vector channel W_N, the next step is channel splitting in recursive manner. This means that the synthesized channel W_N is split into N distinct bit-channels $W_N^{(i)}: X \to X^{i-1} \times Y^N$, $1 \leq i \leq N$ as seen in **Figure 2.7**. Each split channel is defined as $W_N^{(i)}: u_i \to (y_1^N, u_1^{i-1})$. The transition probability of the channel $W_N^{(i)}$ is indicated in Equation 2.12.

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) = \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N) \qquad 2.12$$

where u_i is the input and (y_1^N, u_1^{i-1}) is the outputs of the channel $W_N^{(i)}$.

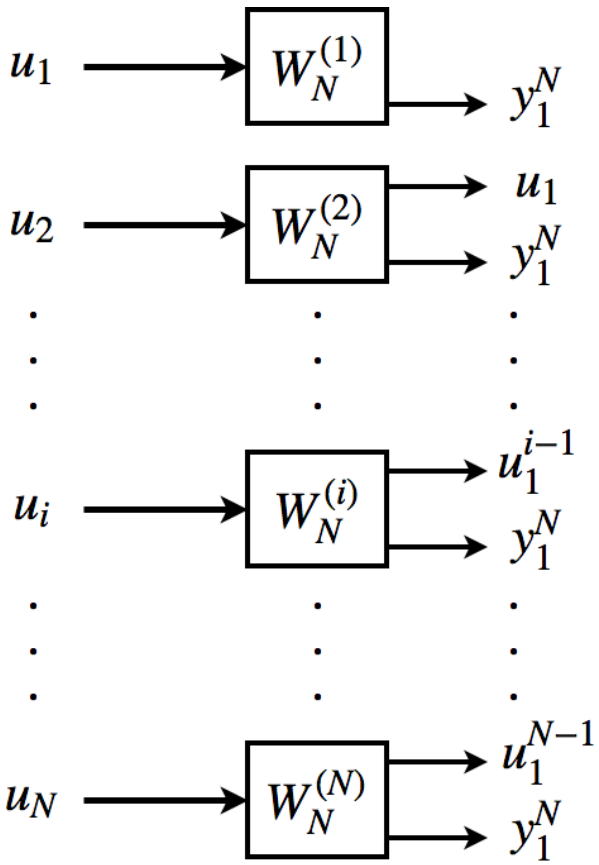


Figure 2.7 The synthesized channel W_N is split into N distinct bit-channels $W_N^{(i)}$.

2.3 Channel Polarization

In this section, we will study on the fact of channel polarization, which shows us that polar codes have the symmetric mutual information on the binary-input discrete memoryless channel. Also; overall symmetric mutual information is conserved through the channel transformation process. According to the chain rule for the

mutual information, the mutual information of the synthesized channel W_N can be written as

$$I(Y_1^N; X_1^N, U_1^N) = I(Y_1^N; X_1^N) + I(Y_1^N; U_1^N | X_1^N) \\ = I(Y_1^N; U_1^N) + I(Y_1^N; X_1^N | U_1^N) \tag{2.13}$$

Since $$I(Y_1^N; U_1^N | X_1^N) = I(Y_1^N; X_1^N | U_1^N) = 0 \tag{2.14}$$

Then $$I(Y_1^N; X_1^N) = I(Y_1^N; U_1^N) \tag{2.15}$$

Therefore $$I(W_N) = I(Y_1^N; X_1^N) = I(Y_1^N; U_1^N) = NI(W) \tag{2.16}$$

Equation 2.16 shows that while combining N copies of the original binary-input discrete memoryless channel W, $NI(W)$ is conserved.

According to the chain rule for the mutual information, the mutual information of the channel W_N can be split as

$$NI(W) = I(U_1^N; Y_1^N) = \sum_{i=1}^{N} I(U_i; Y_1^N, U_1^{i-1}) \tag{2.17}$$

After channel combining and splitting process are operated in recursive manner, N copies of the original binary-input discrete memoryless channel W transform into N distinct channels $W_N^{(i)}$, $i \in \{1, \dots, N\}$. However; the symmetric mutual information of each distinct channel $W_N^{(i)}$ is different than that of the original channel W. Theorem 2.1, presented by Arıkan in his influential paper published in 2009, shows that while N goes to infinity, $W_N^{(i)}$ converges to either 0 or 1. This phenomenon is called "channel polarization".

Theorem 2.1 [Arıkan, 2009]: For any B-DMC W, the channels $\{W_N^{(i)}\}$ polarize in the sense that, for any fixed $\delta \in (0,1)$, as N goes to infinity through powers of two, the fraction of indices $i \in \{1, \dots, N\}$ for which $I\left(W_N^{(i)}\right) \in (1-\delta, 1]$ goes to $I(W)$ and the fraction for which $I\left(W_N^{(i)}\right) \in [0, \delta)$ goes to $1 - I(W)$. That is,

$$\lim_{N\to\infty} \frac{number\ of\ channels\ with\ I\left(W_N^{(i)}\right) \in (1-\delta, 1]}{N} = I(W) \qquad 2.18$$

$$\lim_{N\to\infty} \frac{number\ of\ channels\ with\ I\left(W_N^{(i)}\right) \in [0, \delta)}{N} = 1 - I(W) \qquad 2.19$$

2.4 Capacities of Transformed N Distinct Channels $W_N^{(i)}$ over a BEC

After the channel transformation operation, N distinct channels $W_N^{(i)}$, $i \in \{1, \dots, N\}$ are produced. In this section, capacity of each of these channels will be computed for a given binary erasure channel W with erasure probability ϵ, abbreviated as BEC(ϵ).

For $N = 2$, split channels have been indicated as $W_2^{(1)}$ and $W_2^{(2)}$ in Section 2.2.2. To be clear throughout operations, first and second split channels for $N = 2$ will be shown as $W_2^{(1)} = W^-$ and $W_2^{(2)} = W^+$. For $N = 4$, when ancestry channel $W_2^{(1)} = W^-$ splits into two channels, they will be $W_4^{(1)} = W^{--}$ and $W_4^{(3)} = W^{-+}$. Similarly; $W_2^{(2)} = W^+$ splits into $W_4^{(2)} = W^{+-}$ and $W_4^{(4)} = W^{++}$. Synthesized channel W_4 from 4 copies of the original B-DMCs W is demonstrated in **Figure 2.8** and split channels $W_4^{(i)}$ are shown in **Figure 2.9**.

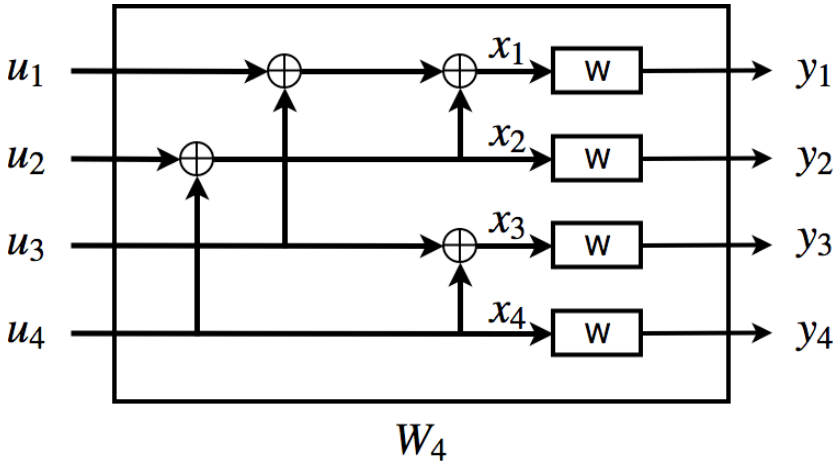


Figure 2.8 The synthesized channel W_4.

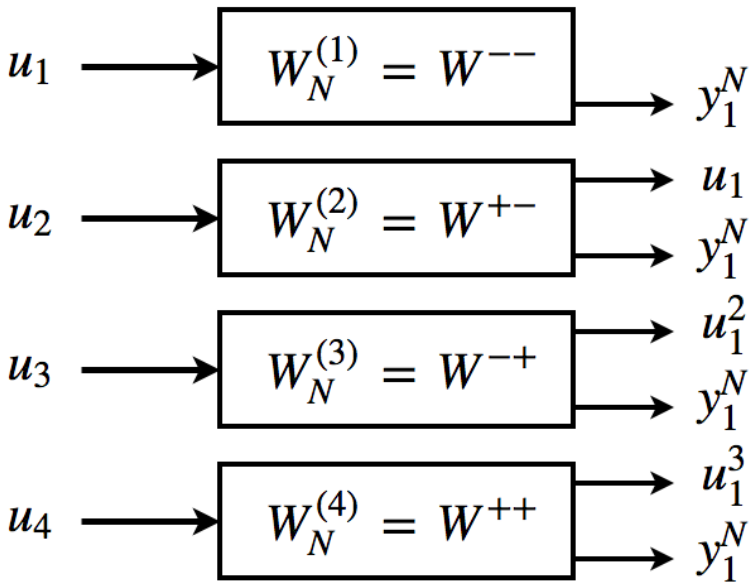


Figure 2.9 The synthesized channel W_4 is split into 4 bit-channels $W_4^{(i)}$.

Channel capacity of a binary erasure channel with erasure probability ϵ equals to $I(W) = 1 - \epsilon$. Since each split channel is also a binary erasure channel, it has its own erasure probability. For $N = 2$, W^- channel is shown in Figure 2.3 on page 13 and $u_1 = x_1 + x_2$. Since x_1 and x_2 are necessary to estimate u_1, erasure probability of W^- equals to $\epsilon^- = 1 - (1 - \epsilon)^2$. Also; W^+ channel is shown in Figure 2.4 on

page 14 and $u_2 = x_2$, $u_2 = x_1 + u_1$. One of these equations is enough to estimate u_2, and u_1 is also assumed as known from W^-; so, the erasure probability of W^+ equals to $\epsilon^+ = \epsilon^2$.

For $N = 4$, erasure probabilities of each split channel $W_4^{(i)}$ can be computed by using ϵ^- and ϵ^+. Thus; the erasure probabilities of the channels:

W^{--} is $\epsilon^{--} = 1 - (1 - \epsilon^-)^2$,

W^{-+} is $\epsilon^{-+} = (\epsilon^-)^2$,

W^{+-} is $\epsilon^{+-} = 1 - (1 - \epsilon^+)^2$ and

W^{++} is $\epsilon^{++} = (\epsilon^+)^2$.

This process can be followed to calculate erasure probabilities of each split channel $W_N^{(i)}$ and their capacities ($1 -$ erasure probability) over BEC(ϵ) for larger N values.

2.5 Polar Code Encoding

In the previous parts, N distinct channels $W_N^{(i)}$, $i \in \{1, \dots, N\}$ have been obtained by channel transformation method. The symmetric capacities of each of them are polarizing and each one converges to 0 or 1, while N becomes larger. The main idea at constructing a (N, K) polar code for blocklength $N = 2^n, n \geq 0$ is choosing K of channels $W_N^{(i)}$, $i \in \{1, \dots, N\}$ with higher capacities to send data and freezing $N - K$ of channels with lower ones. Since frozen bits do not have the capability to transmit data bits reliably, they are not chosen to send data. Thus; frozen bits have predetermined values known by the encoder and the decoder. Then, inputs are mapped to codewords through the linear encoder, $x_1^N = u_1^N F^{\otimes n}$ where $F^{\otimes n}$ is the n^{th}-Kronecker product of the base matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

After obtaining N distinct channels $W_N^{(i)}$, $i \in \{1, \dots, N\}$, we will use A to denote the set of frozen bit channels, which have converging symmetric capacities to 0. Size of this set is $N - K$, where blocklength $N = 2^n$. Their corresponding input bits will be frozen and indicated as u_A. Frozen bits will be predetermined as 0 for simplicity in this thesis. Also; information bits are denoted as complement of frozen bits, u_{A^c}. The set of K information bit channels will be shown as A^c and they are chosen according to higher symmetric capacity values.

The Bhattacharyya parameters are used by Arıkan, while choosing the information bits [Arıkan, 2009]. The Bhattacharyya parameters $Z(W)$ are defined as in Equation 2.20 by Arıkan. After Bhattacharyya parameters of N distinct channels $W_N^{(i)}$, $i \in \{1, \dots, N\}$ are computed, the set of $N - K$ channels with largest $Z(W_N^{(i)})$ is called frozen set as explained in Equation 2.21.

$$Z(W) = \sum_y \sqrt{W(y|0)W(y|1)} \tag{2.20}$$

$$Z\left(W_N^{(i)}\right) \geq Z\left(W_N^{(j)}\right), \forall i \in F, \forall j \in F^c \tag{2.21}$$

Each row of $F^{\otimes n}$ matrix is indexed as $i \in \{1, \dots, N\}$. Then, Bhattacharya parameters of N distinct channels $W_N^{(i)}$ are calculated for BEC(ϵ) according to Equation 2.22.

$$Z\left(W_N^{(2i-1)}\right) = 2Z(W_{\frac{N}{2}}^{(i)}) - Z(W_{\frac{N}{2}}^{(i)})^2 \tag{2.22}$$

$$Z(W_N^{(2i)}) = Z(W_{\frac{N}{2}}^{(i)})^2$$

When initial condition $Z(W_1^{(1)})$ is chosen as the erasure probability ϵ of the channel, the corresponding code is called an adaptive polar code [Arıkan, 2009]. After computation of all Bhattacharya parameters, indices of the K lowest of them are used to create the generator matrix. The $K \times N$ generator matrix, G_P, is constructed from

rows of $F^{\otimes n}$ matrix, whose indices are corresponding to the indices of the K lowest Bhattacharya parameters.

Moreover; indices of the K lowest Bhattacharya parameters equal to indices of the K highest capacities of channels for the reference factor graph $(n - \cdots - 1)$ for BEC(ϵ) [Arıkan, 2009]. However; as also mentioned by Peker, others than the reference factor graph do not satisfy this equality [Peker, 2018]. Capacity of each channel $W_N^{(i)}$ of the reference factor graph (RFG) equals to $I\left(W_N^{(i)}\right) = 1 - Z(W_N^{(i)})$, where $i \in \{1, \dots, N\}$.

For example; the process of choosing the information bits for an (8, 4) polar code over BEC(0.35), and encoding of the word [1,1,1,0] are shown in **Figure 2.10**. There are 4 information and 4 frozen bits. The first left column shows channel capacities of the RFG from each input. Second column indicates Bhattacharya parameter values of each indices. As it can be seen that for each indices, each summation of channel capacities and Bhattacharya parameters equals to 1. According to the 4 highest channel capacities or the 4 lowest Bhattacharya parameters, indices of information bits, green ones, are $A^c = \{4,6,7,8\}$ whereas frozen bits, red ones, $A = \{1,2,3,5\}$. Frozen bit values are predetermined to 0. Word to send is $[1,1,1,0]$ and each bit will convey over inputs $u_{A^c} = \{u_4, u_6, u_7, u_8\}$. Moreover; since the 4 highest channel capacities are at $\{4,6,7,8\}$, the generator matrix G_P of (8, 4) Polar code consists of 4$^{\text{th}}$, 6$^{\text{th}}$, 7$^{\text{th}}$ and 8$^{\text{th}}$ rows of $F^{\otimes 3}$ matrix as seen in Equation 2.23. When we calculate the codeword at encoder output by using formula $x_1^8 = u_1^8 G_P$, we can say that the codeword x_1^8 equals to the result of the operation in Figure 2.10.

$$G_P = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} for\ (8,4)\ Polar\ code \tag{2.23}$$

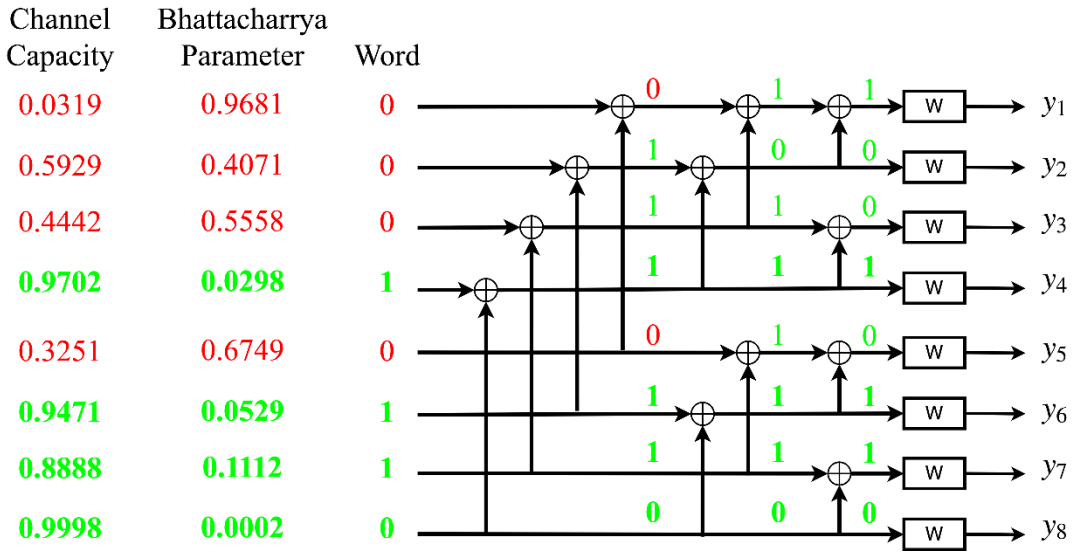


Figure 2.10 Choosing the information bits for an (8, 4) polar code over BEC(0.35), and encoding of the word [1,1,1,0].

As an example given in **Figure 2.10**, encoding and decoding process of polar codes can be represented by factor graphs (FG's). Connections in these factor graph representations can be simplified as Z-shape connections as presented by Korada [Korada, 2009]. **Figure 2.11** shows the redrawn version of **Figure 2.10** with Z-shape connections. Each Z-shape connection connects two inputs and two outputs. Also; these Z connections' heights can differ and it depends on difference between indices of two inputs. For example, if height of a Z-shape is 1, difference between indices of two inputs is 1 and this is the smallest Z-shape. Height of a Z-shape can be $2^{(n-1)}$ at maximum. These height values also represent the stage numbers [Doğan, 2015]. Order of these stage numbers will be used to name factor graphs in the next chapters. Moreover; each stage consists of $\frac{N}{2}$ number of Z-shape connections with same height. Each factor graph has n stages and these stages can be permuted with each other in $n!$ different ways. So; there are $n!$ different factor graphs for a code. This will be studied on next chapter in detail.

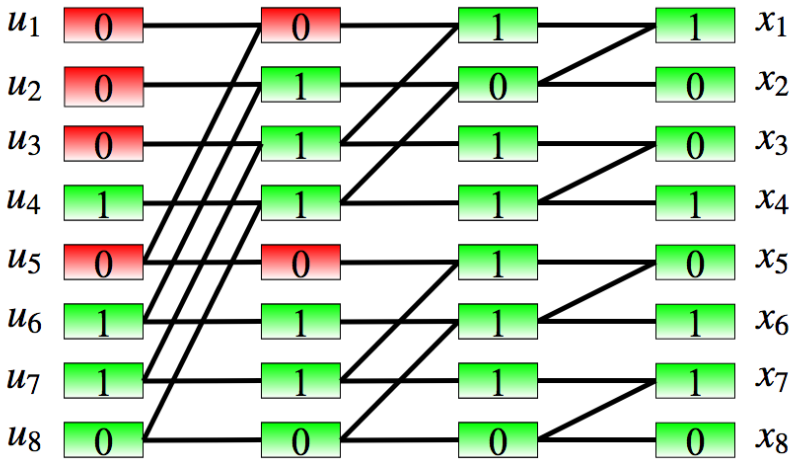


Figure 2.11 Redrawn reference factor graph representations of **Figure 2.10** with Z-shape connections.

2.6 Belief Propagation Decoding Algorithm

Since polar codes are close relatives with Reed Muller (RM) codes, as Forney applied for RM codes [Forney, 2001], Belief Propagation (BP) decoding algorithm can be used for polar codes as well. According to [Korada, 2009], performance of BP decoding algorithm is worse than MAP, but better than successive cancellation (SC) decoding algorithm. Arıkan used SC decoding algorithm in [Arıkan, 2009]. However; SC decoding algorithm has high decoding latency and low throughput due to its serial processing scheme [Zhang, Liu, Pan, Ye, & Gong, 2014]. Therefore; after this point, we will use Belief Propagation (BP) decoding algorithm in our simulations.

Belief Propagation (BP) decoding algorithm is a kind of message passing algorithm (MPA). This kind of algorithms are applied on codes with graphical representations to decode. As mentioned in Section 2.5, codes can be represented with factor graphs. Each FG consists of n stages and each stage has $\frac{N}{2}$ number of Z-shape connections. Also; each stage contains N inputs and N outputs nodes. Input nodes of first stage are input bits and output nodes of last stage are output bits. Inputs of each stage are

outputs of previous stage. Thus; there are $(n+1)N$ variable nodes in a factor graph and except nodes of input bits, there are nN nodes, which are called check nodes. For example; in **Figure 2.12**, for $N = 8$, reference factor graph (321) is indicated with 24 check nodes as square and 32 variable nodes as circle.

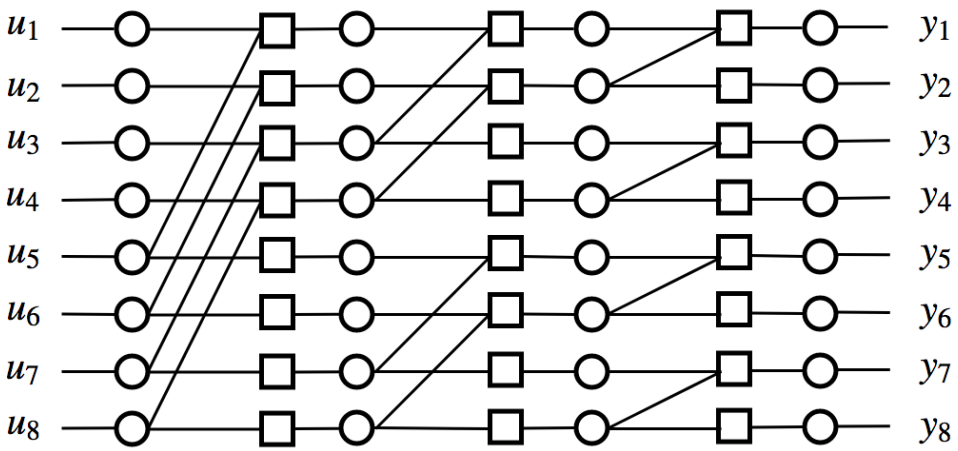


Figure 2.12 Reference factor graph with check nodes (square) and variable nodes (circle) for $N = 8$.

As it can be seen in **Figure 2.12**, there are 4 Z-shape connections in each stage. Processing details of each one is demonstrated in **Figure 2.13**. There are two inputs and two outputs at each Z-shape connection. Input and output nodes are shown with v_I and v_O, respectively. Stage numbers are indicated with i, where $i = 1 \dots n$. The $i = 1$ means that nodes of first stage are in process. The j and k indicators are used for rows of the diagram, where $j = k - 2^{i-1} = 1 \dots (N-1)$. Iteration numbers are represented by t. Messages with $t-1$ as exponential indicator carry previous iterations' information. Messages, whose directions are towards to left, are indicated with L, whereas R for messages with right directions. That is; in **Figure 2.13**, $L^t_{v_I(i,j)}$ and $L^t_{v_I(i,k)}$ have the probability messages, which are going to $v_I(i,j)$ and $v_I(i,k)$ input nodes at the t^{th} iteration. Also; $L^{t-1}_{v_O(i,j)}$ and $L^{t-1}_{v_O(i,k)}$ the probability messages, which come out of $v_O(i,j)$ and $v_O(i,k)$ output nodes at the previous $(t-1)^{th}$ iteration. These can also be said similarly for messages with right direction.

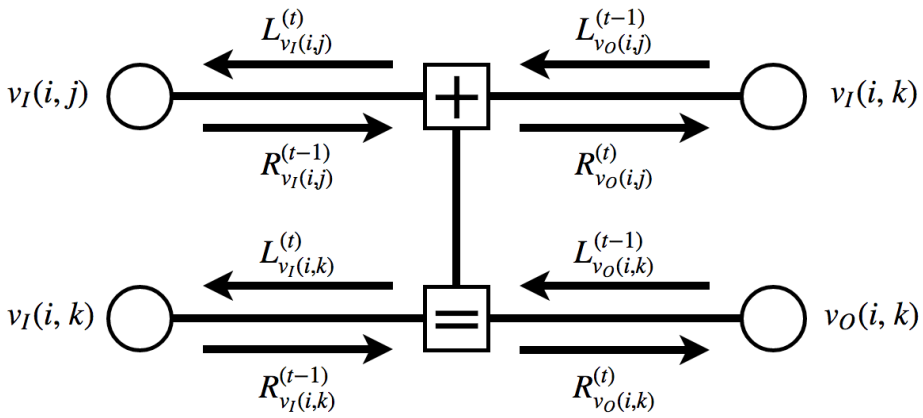


Figure 2.13 Decoding process details of a Z-shape connections.

The Belief Propagation (BP) decoding algorithm that we have used in this thesis is different than the common version. This altered version has also been used in [Xu, Che, & Choi, 2015] and [Doğan, 2015]. In our version, iteration starts from the right most nodes, which are the outputs of the last stage and received bits at the same time. Then, nodes are updated only by messages directed to the left until reaching the leftmost nodes, which are inputs of the first stage. After updating the leftmost nodes, messages merely with right direction are used till the rightmost nodes. This process is corresponding to one iteration.

Decoding process should firstly start with defining which nodes are the frozen variables. As it will be explained in Chapter 3, frozen bits at the input of the diagram may cause frozen variable nodes at inner nodes of the diagram.

After assigning all frozen variables, one should fill the output variable nodes of the diagram, $v_O(n,j)$ & $v_O(n,k)$, by using 0's and 1's of the received word and 0.5 for erased bits for the BEC in order to start the decoding process from the rightmost nodes of the diagram. Then; operations in Equation 2.24 are employed to shift the probability messages from the rightmost nodes to the leftmost nodes of the diagram.

$$L^{t}_{v_I(i,j)} = L^{t-1}_{v_O(i,j)} \otimes [L^{t-1}_{v_O(i,k)} \odot R^{t-1}_{v_I(i,k)}] \quad 2.24$$

$$L^{t}_{v_I(i,k)} = [R^{t-1}_{v_I(i,j)} \otimes L^{t-1}_{v_O(i,j)}] \odot L^{t-1}_{v_O(i,k)}$$

Also; shifting the probability messages from the left most nodes to the right most nodes can be done with operations in Equation 2.25 [Zhang, Zhang, Pan, Ye, & Gong, 2014]. That is; these two equations are used in sequence to complete one iteration.

$$R^{t}_{v_O(i,j)} = R^{t-1}_{v_I(i,j)} \otimes [L^{t-1}_{v_O(i,k)} \odot R^{t-1}_{v_I(i,k)}] \quad 2.25$$

$$R^{t}_{v_O(i,k)} = [R^{t-1}_{v_I(i,j)} \otimes L^{t-1}_{v_O(i,j)}] \odot R^{t-1}_{v_I(i,k)}$$

Two operations $\otimes$ and $\odot$ are defined in Equation 2.26. The probability that x equals 0 or 1 is indicated as $p_x(0)$ or $p_x(1)$, respectively. For messages with left direction, $p_x(0)$ refers to $L_v(0)$, whereas $p_x(1)$ refers to $L_v(1)$. It can also be written for messages with right direction as $p_x(0)$ for $R_v(0)$ and $p_x(1)$ for $R_v(1)$.

$$(p_x \otimes p_y)(0) = p_x(0)p_y(0) + p_x(1)p_y(1)$$

$$(p_x \otimes p_y)(1) = p_x(0)p_y(1) + p_x(1)p_y(0) \quad 2.26$$

$$(p_x \odot p_y)(0) = p_x(0)p_y(0)$$

$$(p_x \odot p_y)(1) = p_x(1)p_y(1)$$

To start updating the nodes in the first iteration by using messages, which come from the rightmost nodes, initial probabilities can be taken as written in Equation 2.27.

$$L^{0}_{v_O(,j)} = \begin{cases} 0 & if\ rw(i) = 0 \\ 1 & if\ rw(i) = 1 \\ 0.5 & if\ rw(i) = e \end{cases} \quad 2.27$$

After completing each iteration, all output nodes of the diagram, $v_O(n,j)$ & $v_O(n,k)$, are checked. If there is no erasure and all nodes are filled with 0's and 1's, the algorithm stops; but if there is still an erased bit, the next iteration of the decoding process starts. The maximum number of iterations to stop the decoding algorithm is predetermined. The optimum number iterations are suggested by Doğan in his thesis [Doğan, 2015], which depend on the blocklength of the code and the erasure rate of the binary erasure channel. In this thesis, we set the maximum number of iterations to 20 for all codes and erasure rates.

CHAPTER 3

SIMULATION RESULTS

In this chapter, we discuss some practical results about the single-FG and multiple-FG (M-FG) belief propagation (BP) decoding performance of polar codes. Single-FG decoder employs a single factor graph, whereas M-FG decoder works with $M > 1$ arbitrarily selected factor graphs, which decode sequentially one after the other. Keeping M around $\lceil n/2 \rceil$, that is the smallest integer greater than equal to half of the number of stages $n = log_2 N$, we try to find matching sets of M factor graphs that improve the decoding performance.

For the classification of $n!$ FG's, two parameters are used in [Doğan, 2015] and [Peker, 2018]: the number of frozen variables (FV), and the capacity-sum (CS). We propose a third and more simply evaluated parameter that helps the classification of FG's, and call it the "stage order number (SON)". In Section 3.1, relations between FV, CS and SON values are explored, and the effects of these parameters on the single-FG BP decoding performance of polar codes are studied.

Section 3.2 is devoted to multiple-FG BP decoding, and it starts by comparing the performance of independent and dependent M-FG BP decoders in Section 3.2.1. Because of its better performance, we prefer to continue with dependent decoding in the remaining part of this work. Section 3.2.2 deals with the effect of changing the order of FG's in dependent M-FG decoding. Section 3.2.3 is about the decoding performances of a twice repeated set of M different FG's, versus that of $2M$ different FG's. In Section 3.2.4, the selection of FG sets for a better-performing M-FG decoder is explored, where the SON values of the factor graphs are also considered in search for a systematic rule. Question of finding similarly performing FG sets with smaller size is asked in Section 3.2.5, SON's of compatible sets are explored in Section 3.2.6 and an algorithm that generates an FG set with $M = \lceil n/2 \rceil$ elements is

given in Section 3.2.7. The performance comparison of $\lceil n/2 \rceil$ FG's generated by this algorithm, with Korada's n cyclically shifted FG's is given for polar codes of different lengths in Section 3.2.8. Finally, dependence of polar code construction on the erasure rate and related performance results are examined in Section 3.2.9.

3.1 Number of Frozen Variables (FV), Capacity Sum (CS) and Stage Order Number (SON) Parameters of the Single-FG BP Decoder

Special factor graphs (FG's) are used to represent the encoding and decoding structure of polar codes. These factor graphs consist of n stages for (N, K) codes, where $N = 2^n$. Each stage contains $N/2$ many Z-shaped connections between two specific 'stage input and corresponding stage output' points. If the positions of the Z-connected stage inputs are such that there is no other input node between them, vertical positions of the corresponding stage outputs are also adjacent; so, the height of the corresponding Z-shaped connection is 1, and it is named as 'Stage-1' in [Doğan, 2015]. The smallest height of a Z-shaped connection is 1, it can only be a power of 2, such as $2^0, 2^1, 2^2, \ldots$ and at maximum $2^{(n-1)}$; the corresponding stages are called $1, 2, 3, \ldots, n$ respectively [Doğan, 2015]; i.e., Stage-i contains $N/2$ many Z-shapes of height $2^{(i-1)}$, and it occurs only once in a factor graph for each $i = 1, \ldots, n$. Since there are n different stages, they can be ordered in $n!$ different ways, so there are $n!$ different factor graphs.

As mentioned in Section 2.4, the capacity of each channel (from an input bit of the FG to all outputs of the FG) can be calculated by using the erasure rate of the BEC. Information bit indices of the polar code computed according to the ascending order of Bhattacharya parameters are in harmony with the descending order of channel capacities [Arıkan, 2009] of the reference factor graph (RFG), whose stage order is from the largest to the smallest, n-…-2-1. In other words, highest K channel capacities of the RFG determine the information bit indices, and they are the same indices determined by the Bhattacharya parameters over a BEC. The factor graph

used for encoding is the RFG structure; however, decoding can be done using other FG's as well, provided that the positions of the information bit indices are kept fixed; i.e., the same as those of the RFG.

As discussed in [Doğan, 2015] and [Peker, 2018], the capacity sum (CS) of the K information channels and the number of frozen variables (FV) of a factor graph may vary from one FG to the other, since always the same information bit indices are used regardless of the factor graph. The reference FG, whose stage order is n-…-2-1, has the highest CS and FV values among $n!$ factor graphs, which may be shared by other FG's as well.

In this work, we suggest another classification parameter for these $n!$ factor graphs, in addition to their CS and FV values, that we call the "stage order number-SON". The SON is defined as the "n-digit integer obtained by writing the stage names of the FG from left to right", like $123\ldots n$, $321\ldots n$, or $n\ldots 231$. Since the SON is unique for each FG, there are $n!$ different SON values for $n!$ different FG's. The SON of the reference FG is the highest of all; i.e., $n(n-1)$ ...321. Effects of the CS, FV and SON parameters over single-FG belief propagation decoding performance will be investigated in Section 3.1.5.

3.1.1 Capacity Sum (CS) Calculation of Factor Graphs

While the stage order of factor graphs are permuting, channel capacity on each information bit index may also permute. In other words, although the set of N channel capacities remains the same, their capacity order may change. This capacity permutation depends on changes in connection paths between each input and output bit. Since information is transmitted using always the same K input bits, capacities of these K channels may differ resulting in different CS values from one factor graph to another one.

For example, channel capacities of the (8, K) adaptive polar codes over BEC (0.35) for the reference factor graph with SON=321 [Peker, 2018] are computed as in **Figure 3.1**. Also; channel capacities of all $3 = log_2 8$ different factor graphs are shown in **Figure 3.2** for $N = 8$; where the arrows connecting equi-capacities are added only to indicate the change of the capacity order.

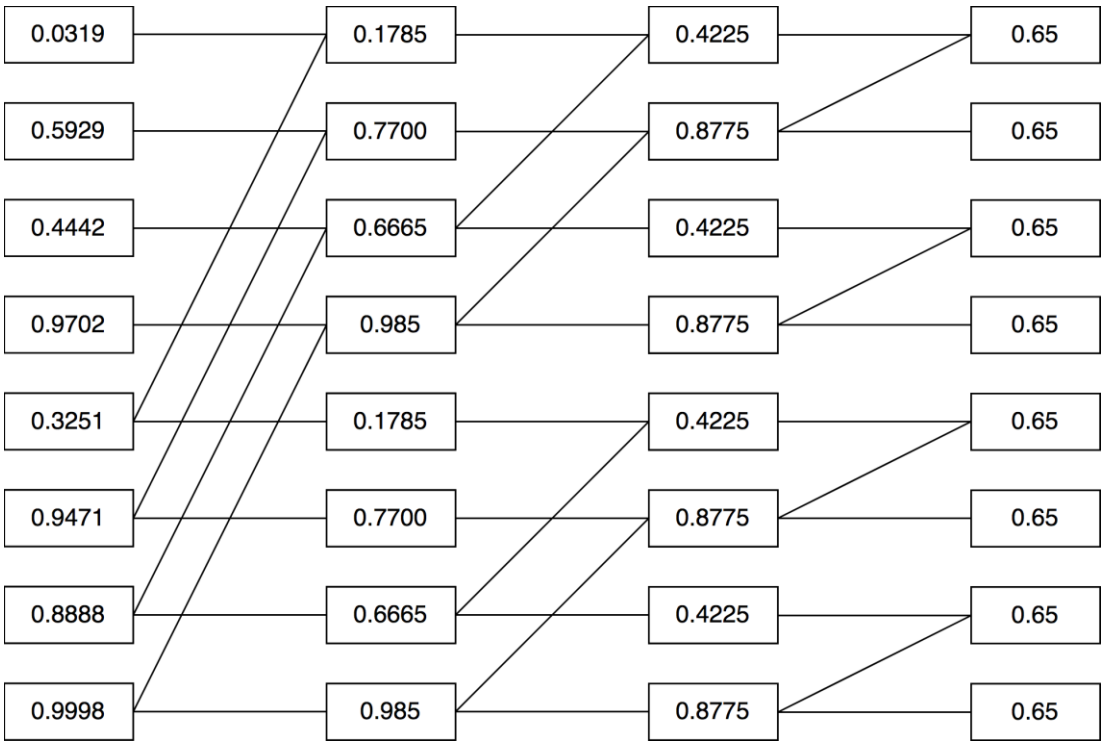


Figure 3.1 Channel capacities of the reference factor graph (3-2-1) for the (8, K) adaptive polar code over BEC (0.35).

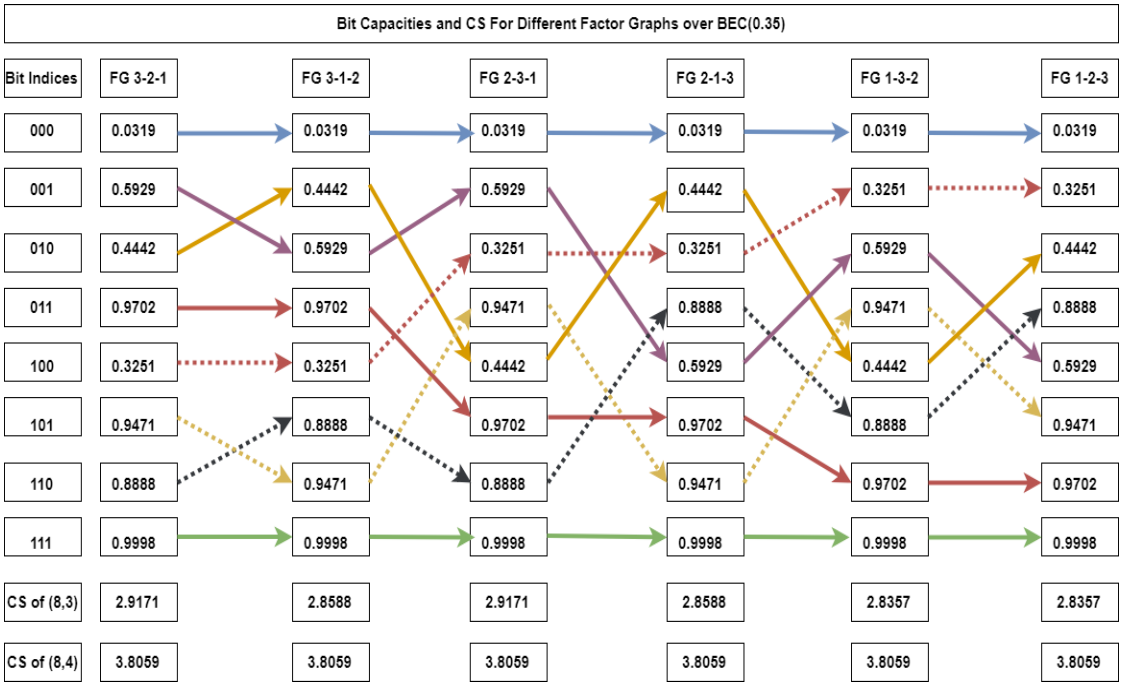


Figure 3.2 Input bit capacities and CS of all factor graphs for the (8, 3) and (8, 4) adaptive polar code over BEC (0.35).

As expected, the reference factor graph 3-2-1 has the maximum CS value for both codes, (8, 3) and (8, 4). "CS of (8, 3)" row in **Figure 3.2**, shows that different factor graphs may have different CS values because of the permuting channel capacities. One can observe that for the (8, 4) codes, CS values of all factor graphs are the same. Because, while stage orders of factor graphs permute, bit capacities of the highest-capacity 4 channels permute among the same 4 information bit indices: 011, 101 and 110. So, the capacity sum of these 4 information channels remain the same for all FG's. Invariation of CS versus FG occurs when the polar code is also a Reed-Muller (RM) code.

The capacity sum of a factor graph is shown to be related to the 1-FG BP decoding performance of the related decoder [Doğan, 2015], [Peker, 2018]. CS grouping of FG's is done and each group of factor graphs with the same CS value is called an "equi-CS set" [Doğan, 2015]. In Figure 3.2, one may observe 3 different equi-CS sets of size 2 for the (8, 3) code. Corresponding CS values are 2.91, 2.85 and 2.83.

3.1.2 Number of Frozen Variables (FV) Calculation of Factor Graphs

There are K "information bits", which are used to convey information and have fixed indices regardless of the stage order; the remaining input bits of the FG are called "frozen bits". There are $N - K$ frozen bits in an (N, K) code. During encoding or decoding, other nodes within the inner stages can be employed for information conduction or they may be frozen. If a node is used for information transmission, it is called an "information variable" otherwise, it is called a "frozen variable".

Except the first stage in the FG, each stage uses N outputs of the previous stage as its inputs. Input pairs of each stage are processed in its own Z-shapes and results are shown as outputs of that stage. To determine whether a node is an information variable or a frozen variable (**Figure 3.3**), one needs to consider the following 4 cases:

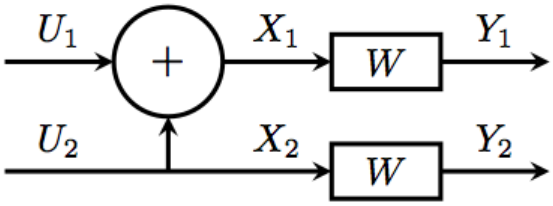


Figure 3.3 Z-shape process.

- If U_1 and U_2 are both frozen variables, then X_1 and X_2 are also frozen variables and no node is an information variable.
- If U_1 is an information variable and U_2 is a frozen variable, then X_1 is an information variable and X_2 is a frozen variable.
- If U_1 is a frozen variable and U_2 is an information variable, then X_1 and X_2 are both information variables.
- If U_1 and U_2 are both information variables, then X_1 and X_2 are also information variables and no node is a frozen variable.

After determining all frozen nodes of the diagram, corresponding 'number of frozen variables' is called FV [Doğan, 2015]. In an (N, K) code, the number of frozen variables, FV, is lower bounded by the number of frozen input bits $N - K$ and it may change by differing the stage order.

As investigated similarly in [Peker, 2018]; when we calculate the number of frozen variables of the (8, 4) and (8, 3) adaptive polar codes over BEC (0.35) for all factor graphs, we can draw a frozen variable diagram of each factor graph. Frozen variable diagrams of (8, 4) and (8, 3) adaptive polar codes are demonstrated in **Figure 3.4** and **Figure 3.5**, respectively.

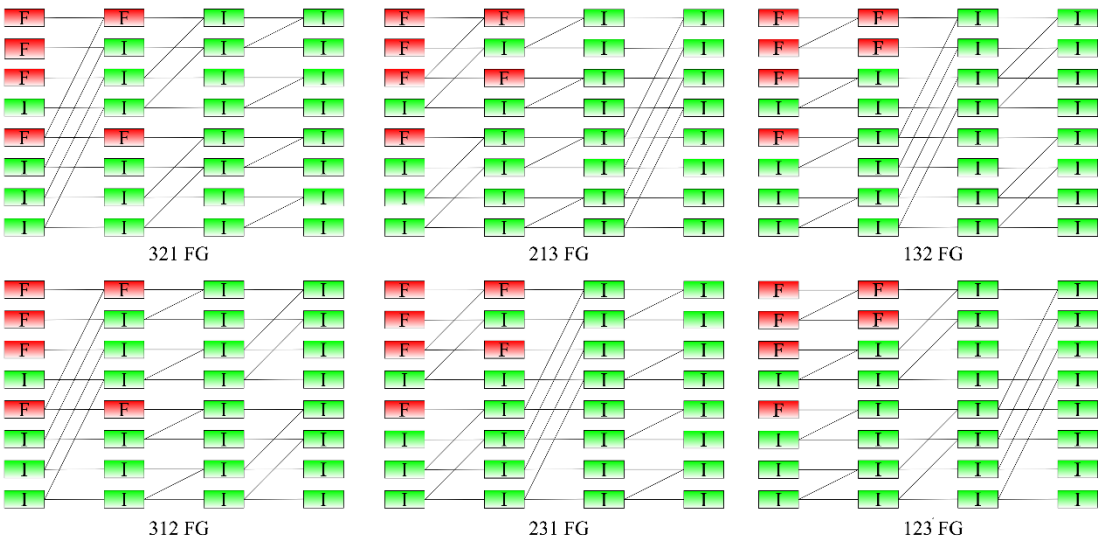


Figure 3.4 Information and frozen variables of all factor graphs for the (8, 4) adaptive polar code over BEC (0.35).

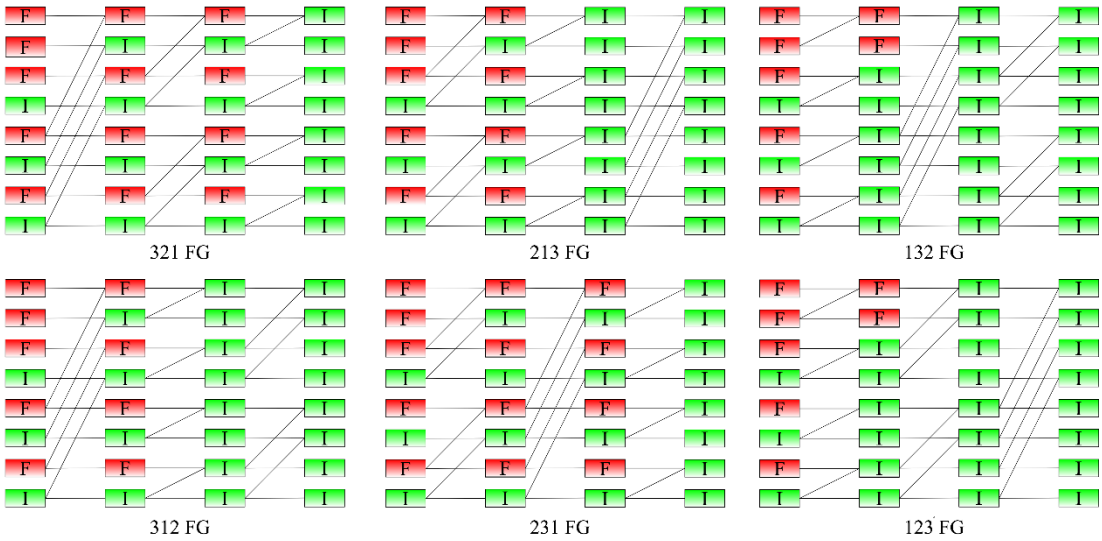


Figure 3.5 Information and frozen variables of all factor graphs for the (8, 3) adaptive polar code over BEC (0.35).

Nodes, which are filled with red and marked with "F", represent frozen variables. Nodes, which are filled with green and marked with "I", represent information variables. One can calculate the number of frozen variables by counting the red nodes. In **Figure 3.4** drawn for the (8, 4) polar (or RM) code, FV is 6 and it remains constant for all factor graphs. As mentioned previously, if the (N, K) polar code is the same as the (N, K) Reed-Muller code, FV and CS values are independent of the stage order of the factor graph.

In **Figure 3.5**, given for the (8, 3) adaptive polar code, one can observe that the FV value may change according to the stage order of factor graph. The minimum FV value is 7 and it belongs to the FG with the least stage order number (123). The maximum FV value is 13 and it belongs to the reference factor graph, which has the biggest stage order number (321). According to our observations, the factor graph with minimum stage order number (SON =12…n), has always the minimum FV and CS values, whereas the reference factor graph, whose stage order number is the maximum (SON = n…21), has the maximum FV and CS values. As expected, since the number of frozen bits at the input increases when the code rate decreases, FV's of the factor graphs also increase.

To observe the FV values of (N, K) codes with larger blocklengths, one can calculate a "frozen variable matrix", which consists of 0's and 1's [Doğan, 2015]; where frozen and information variables are shown with "0" and "1", respectively. Each node in the diagram is matched to one location in the matrix. For instance, elements of the first column of the matrix show the input bit states, while ones at the last column indicate the output bit states. Since there are N input bits and n stages, the matrix size is $N \times (n + 1)$.

For example; for the (32, 16) code, there are $5! = 120$ different factor graphs. In **Table 3.1**, we show the frozen variable matrices of the FG's with SON's 54321 and 12345 by indicating the frozen and information variables in different colors. Number of zeros; i.e, FV of RFG is 44 and that of the FG with minimum SON =12345 is 24. Since the (32, 16) adaptive polar code is different from the (32, 16) Reed-Muller code, one may get different FV values for different factor graphs. FV's of all other factor graphs differ between 24 and 44.

Table 3.1 Frozen variable matrices of RFG and FG with least stage order number for the (32, 16) adaptive polar code over BEC (0.35).

PC N=32 K=16 (54321)						PC N=32 K=16 (12345)					
0	0	0	0	1	1	0	0	1	1	1	1
0	0	1	1	1	1	0	0	1	1	1	1
0	0	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	0	1	1	1	1
0	1	1	1	1	1	0	0	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	0	1	1	1	1
0	1	1	1	1	1	0	0	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	0	1	1	1	1
0	0	1	1	1	1	0	0	1	1	1	1
0	0	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
FV = 44						**FV = 24**					

FV values can be used as a sorting feature for factor graphs. Factor graphs with the same FV values are called “equi-FV sets” [Doğan, 2015]. Peker studies on equi-FV and equi-CS groups of (32, 16), (64, 32) and (128, 64) in detail and their performances over BEC(0.35) [Peker, 2018]. According to **Table 3.2** reproduced from [Peker, 2018]; for the (32, 16) adaptive polar code over BEC (0.35), 5 different equi-FV sets and 10 different equi-CS sets are created from 120 different factor graphs. Each equi-CS set consists of 12 factor graphs. The factor graphs with the least FV and CS values are in the same group. Similarly, highest FV and CS-valued sets are the same. The number of equi-FV sets is less than the number of equi-CS sets; that is, equi-CS sets are the subsets of equi-FV sets. In the following sections, we will discuss that FG’s in equi-FV or equi-CS sets have similar codeword error ratio (CER) performances in single-FG BP decoding.

Table 3.2 Distribution of equi-CS and equi-FV sets for the (32, 16) polar code over BEC(0.35). (Reproduced from [Peker, 2018])

Equi-FV set no.	FV	CS	Number of FG's in the equi-CS set	Number of equi-CS sets in the equi-FV set
1	44	15.67	12	1
2	36	15.48	12	2
	36	15.23	12	
3	32	15.32	12	3
	32	15.07	12	
	32	14.88	12	
4	28	15.18	12	3
	28	14.94	12	
	28	14.78	12	
5	24	14.7	12	1
5 Equi-FV sets	**Total**		**120 FG's**	**10 Equi-CS sets**

3.1.3 Number of Frozen Variable and Capacity Sum Calculation for Reed Muller Codes

As mentioned in Section 2.1, n-Kronecker power matrix $F^{\otimes n}$ is calculated by taking the self Kronecker product of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, n times. The size of $F^{\otimes n}$ is $2^n \times 2^n$. Polar codes are close relatives of Reed-Muller (RM) codes, and generator matrices of both codes are chosen from the same n-Kronecker power matrix. So, the blocklength is $N = 2^n$ for both codes; but the information word length for polar codes can be any integer $K \leq N$, whereas for an r^{th} order Reed-Muller code, RM (r, n), $K = \sum_{i=0}^{r} \binom{n}{i}$. In Chapter 2.5, it is stated how the highest-capacity K input bits are used to select K rows of $F^{\otimes n}$ in order to arrange the generator matrix G_P of the polar code.

On the other hand; the generator matrix G_{RM} of a Reed-Muller code is produced with rows of $F^{\otimes n}$ whose Hamming weights are greater than or equal to 2^{n-r}.

The generator matrix of an (N, K) Reed-Muller code can be exactly the same as that of an (N, K) polar code for some N and K values. For example, the generator matrices of (8, 4) polar and RM codes, whose rows are selected from $F^{\otimes 3}$ depending on bit capacities for the polar, and row Hamming weights for the RM, are the same (see **Table 3.3**). Generator matrices of both codes are formed by the rows of $F^{\otimes 3}$, whose indices are 4, 6, 7 and 8.

Table 3.3 Information bit indices and rows of generator matrices for the (8, 4) polar and Reed-Muller codes over BEC (0.35).

Bit Capacities	**3-Kronecker Power Matrix**								**Hamming Weights of Rows**
0.0318	1	0	0	0	0	0	0	0	1
0.5929	1	1	0	0	0	0	0	0	2
0.4442	1	0	1	0	0	0	0	0	2
0.9702	1	1	1	1	0	0	0	0	4
0.3251	1	0	0	0	1	0	0	0	2
0.9471	1	1	0	0	1	1	0	0	4
0.8887	1	0	1	0	1	0	1	0	4
0.9997	1	1	1	1	1	1	1	1	8

Although (8, 4) polar and RM codes are the same, generator matrices of $(N,\ N/2)$ codes start to differ for $N = 32$; hence (32, 16) polar and RM codes differ by one row. As the code length increases, the number of different rows in the generator matrices G_P and G_{RM} also increases. For $N = 32$ and $K = 16$, RM (2, 5) code chooses all rows of the 5-Kronecker power matrix whose Hamming weights are 32, 16 and 8 in order to arrange a 16 by 32 generator matrix G_{RM}. However; (32, 16) polar code selects rows of the 5-Kronecker power matrix depending on the highest 16 channel capacities in the reference FG; and picks up a weight-4 row for G_P, instead of a weight-8 row, which is included in G_{RM}. As can be seen in **Table 3.4**, the channel capacity of the green colored weight-4 row is greater than that of the pink

colored weight-8 row. These two rows are the only differing rows of the generator matrices G_P and G_{RM} of the (32, 16) polar and RM codes.

Table 3.4 Information bit capacities of RFG and Hamming weights of rows for the (32, 16) polar and Reed-Muller codes over BEC (0.35).

Information Bit Capacities of (32,16) PC (54321)	Information Bit Hamming Weights of (32,16) PC	Information Bit Capacities of (32,16) RM (54321)	Information Bit Hamming Weights of (32,16) RM
0.8860695	4	0.9990995	8
0.9990995	8	0.9982262	8
0.9982262	8	0.9944116	8
0.9944116	8	0.9754104	8
0.9754104	8	0.9999998	16
0.9999998	16	0.9965557	8
0.9965557	8	0.9893917	8
0.9893917	8	0.9558657	8
0.9558657	8	0.9999997	16
0.9999997	16	0.9725360	8
0.9725360	8	0.9045819	8
0.9045819	8	0.9999992	16
0.9999992	16	0.7925878	8
0.9999921	16	0.9999921	16
0.9998469	16	0.9998469	16
0.9999999	32	0.9999999	32

For the (N, K) Reed-Muller codes, although bit capacities are permuted and frozen variable matrices may be different for each factor graph, CS and FV values remain always the same as indicated in **Table 3.5**.

Table 3.5 Bit capacities and frozen variable matrices of factor graphs with stage order number 54321 and 12345 for the (32, 16) Reed-Muller codes over BEC (0.35).

Bit Capacities (54321)	Frozen Variable Matrix (54321)						Bit Capacities (12345)	Frozen Variable Matrix (12345)					
0.0000010	0	0	0	1	1	1	0.0000010	0	0	0	1	1	1
0.1235816	0	0	1	1	1	1	0.0020296	0	0	0	1	1	1
0.0389375	0	0	1	1	1	1	0.0039329	0	0	0	1	1	1
0.8860695	0	1	1	1	1	1	0.1214942	0	0	0	1	1	1
0.0111769	0	0	1	1	1	1	0.0111769	0	0	1	1	1	1
0.8046159	0	1	1	1	1	1	0.2002654	0	0	1	1	1	1
0.6239712	0	1	1	1	1	1	0.2965617	0	1	1	1	1	1
0.9990995	1	1	1	1	1	1	0.7925878	1	1	1	1	1	1
0.0039329	0	0	0	1	1	1	0.0389375	0	0	1	1	1	1
0.6960188	0	1	1	1	1	1	0.3557144	0	0	1	1	1	1
0.4776217	0	1	1	1	1	1	0.4776217	0	1	1	1	1	1
0.9982262	1	1	1	1	1	1	0.9045819	1	1	1	1	1	1
0.2965617	0	1	1	1	1	1	0.6239712	0	1	1	1	1	1
0.9944116	1	1	1	1	1	1	0.9558657	1	1	1	1	1	1
0.9754104	1	1	1	1	1	1	0.9754104	1	1	1	1	1	1
0.9999998	1	1	1	1	1	1	0.9998469	1	1	1	1	1	1
0.0020296	0	0	0	1	1	1	0.1235816	0	0	1	1	1	1
0.5795019	0	0	1	1	1	1	0.5795019	0	0	1	1	1	1
0.3557144	0	0	1	1	1	1	0.6960188	0	1	1	1	1	1
0.9965557	1	1	1	1	1	1	0.9725360	1	1	1	1	1	1
0.2002654	0	0	1	1	1	1	0.8046159	0	1	1	1	1	1
0.9893917	1	1	1	1	1	1	0.9893917	1	1	1	1	1	1
0.9558657	1	1	1	1	1	1	0.9944116	1	1	1	1	1	1
0.9999997	1	1	1	1	1	1	0.9999921	1	1	1	1	1	1
0.1214942	0	0	0	1	1	1	0.8860695	0	1	1	1	1	1
0.9725360	1	1	1	1	1	1	0.9965557	1	1	1	1	1	1
0.9045819	1	1	1	1	1	1	0.9982262	1	1	1	1	1	1
0.9999992	1	1	1	1	1	1	0.9999992	1	1	1	1	1	1
0.7925878	1	1	1	1	1	1	0.9990995	1	1	1	1	1	1
0.9999921	1	1	1	1	1	1	0.9999997	1	1	1	1	1	1
0.9998469	1	1	1	1	1	1	0.9999998	1	1	1	1	1	1
0.9999999	1	1	1	1	1	1	0.9999999	1	1	1	1	1	1
CS = 15.5785050	FV = 30						CS = 15.5785050	FV = 30					

3.1.4 Interrelation between the Number of Frozen Variables, Capacity Sum and Stage Order Numbers

As studied in previous sections, each factor graph with different stage order has a "capacity sum-CS", and a "number of frozen variables-FV". CS and FV values of an arbitrary factor graph can be seen as parameters that predict the BP decoding performance of that factor graph, single-FG BP decoding is more successful with FG's with high CS and FV values. In **Figure 3.6**, we sketch CS values of $6! = 720$ different FG's versus their FV values for the (64, 32) adaptive polar code over BEC (0.35). One can say that although equi-FV groups have a somewhat wide range of CS values, FV and CS parameters have a positive correlation in general.

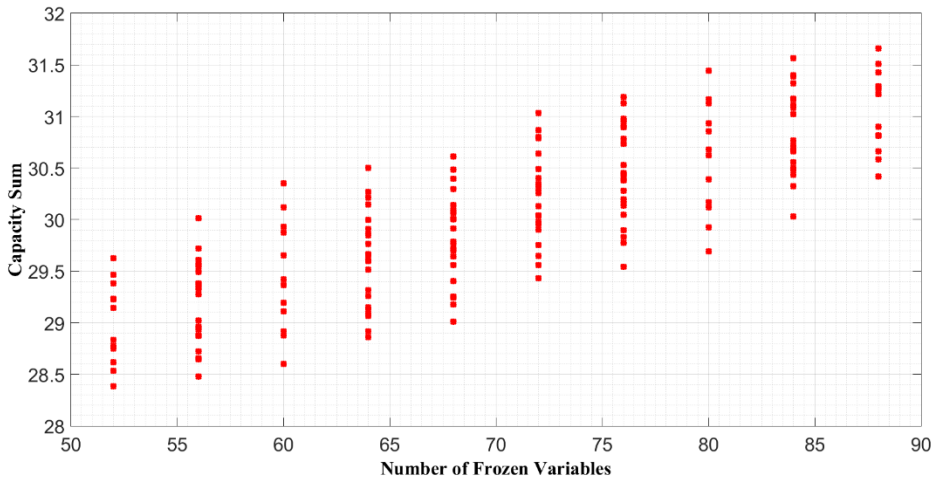


Figure 3.6 Relation between FV and CS values of factor graphs for the (64, 32) adaptive polar code over BEC (0.35).

It was mentioned that the "stage order number-SON" of a factor graph is a number with n digits. For example; if the stage order is 1-2-3-…-7, then the corresponding SON = 1234567. SON values of FG's are unique, so they can be used as another parameter to classify FG's.

In **Figure 3.7**, the relation between FV and SON values is studied for the (64, 32) adaptive polar codes over BEC (0.35). Factor graphs, which are sorted by ascending order of their SON values, are placed in the x-axis and their corresponding FV values are indicated in the y-axis. As shown in **Figure 3.7**, FV values range between 52 and 88 and one observes that although FV is not linearly related to SON, these parameters have parallel increment-decrement trends. Moreover; when we repeat a similar experiment to compare CS with SON values of factor graphs in **Figure 3.8**, CS values are in the interval of [28.3858, 31.6595] and they have a rising tendency while SON values of factor graphs are increasing. As expected, since the greatest SON value (654321) belongs to the RFG, it has the maximum FV and CS values. Other FG's with maximum FV and CS values are 654231, 645321 and 645231. Also; in both figures, there are grouped data points, which are shown in different colors and each of them belongs to factor graphs, which start with the same stage number. For instance, the orange colored group of the smallest SON-valued factor graphs, all start with the stage number '1'.

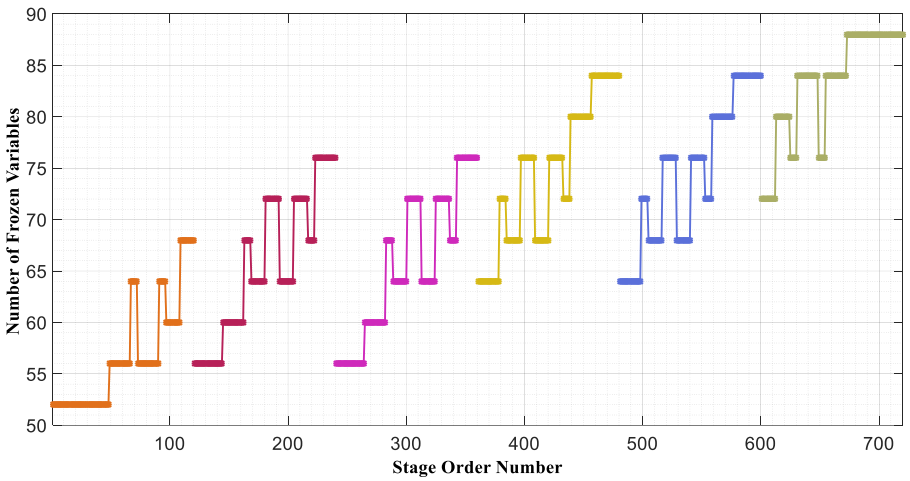


Figure 3.7 FV versus SON values of all factor graphs for the (64, 32) adaptive polar code over BEC (0.35).

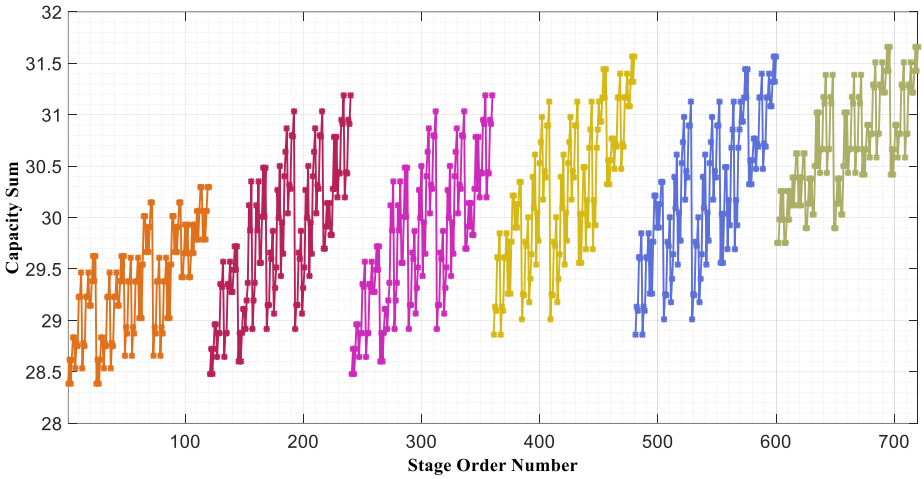


Figure 3.8 CS versus SON values of all factor graphs for the (64, 32) adaptive polar code over BEC (0.35).

As expected, both **Figure 3.7** and **Figure 3.8** reach their maximum values on the y-axis (maximum FV and CS values respectively) for the largest SON on the x-axis, which corresponds to the reference FG.

In order to observe the relations between these parameters for larger blocklengths; in **Table 3.6** we tabulate the correlation coefficients $r_{\text{FV,CS}}$, $r_{\text{FV,SON}}$ and $r_{\text{CS,SON}}$ calculated by

$$r_{xy} = \frac{\sum_{i=1}^{n!}(x_i - (\bar{x}))(y_i - (\bar{y}))}{\sqrt[2]{\sum_{i=1}^{n!}[x_i - (\bar{x})]^2 \sum_{j=1}^{n!}[y_j - (\bar{y})]^2}} \qquad 3.1$$

where $(\bar{x}) = \sum_{i=1}^{n!} x_i/n!$ and $(\bar{y}) = \sum_{i=1}^{n!} y_i/n!$ denote the average values. During the computation of correlation coefficients r_{xy} related to SON, corresponding indices in the SON-sorted list are used instead of SON's. One can say from **Table 3.6** that all correlation coefficients between FV-CS, FV-SON and CS-SON are positive for different polar codes; but they tend to decrease while the blocklength gets larger.

Table 3.6 Correlation coefficients r_{xy} between the FV-CS, FV-SON and CS-SON parameters of different polar codes over BEC(0.35).

x - y	**(16,8) Polar Code**	**(32,16) Polar Code**	**(64,32) Polar Code**	**(128,64) Polar Code**	**(256,128) Polar Code**	**(512,256) Polar Code**
FV - CS	0.91	0.86	0.82	0.81	0.79	0.77
FV - SON	0.74	0.74	0.85	0.77	0.75	0.72
CS - SON	0.66	0.64	0.63	0.53	0.49	0.44

3.1.5 Single-FG Performance Dependence on the Number of Frozen Variables, Capacity Sum and Stage Order Number

Our aim is to observe the effects of the number of frozen variables (FV), capacity sum (CS) and stage order number (SON) values on the single-FG performance. For the (64, 32) adaptive polar code, there are $6! = 720$ different factor graphs and arbitrarily chosen factor graphs can be used in this experiment. BP decoding performance of the (64, 32) adaptive polar code over BEC (0.35) is examined by using a perfect knowledge-based stopping criterion and setting the maximum number of iterations to 20. Codeword error ratio (CER) is measured, after 1000 codewords with erased bits are decoded.

To observe the effect of the number of frozen variables for all possible factor graphs on the single-FG performances, single-FG performance of all FG's are calculated and put in ascending order of FV values in **Figure 3.9**. Although there are some jump off points, one observes that single-FG performance is getting more successful while factor graphs with higher FV values, are employed. Factor graph with the highest FV value is approximately 30% superior to the one with the lowest FV value. The best single-FG performance belongs to the factor graph with maximum FV value.

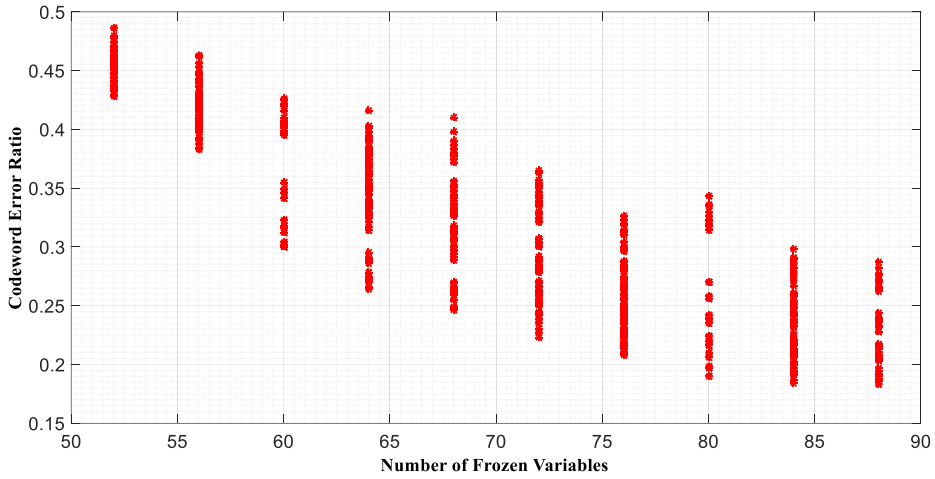


Figure 3.9 Codeword Error Ratio (CER) performance versus the Number of Frozen Variables for Single-FG BP decoder of the (64, 32) adaptive polar code over BEC(0.35).

To observe the effect of the capacity sum for all possible factor graphs on the single-FG performances, single-FG performance of all FG's are calculated and put in ascending order of CS values in **Figure 3.10**. It can be said that while CS values are increasing, even there is no exact linearity, single-FG performances have an average trend of improving. Single FG performances change between CER values of 0.486 and 0.183 depending on the factor graph used in decoding. Also; the factor graph with the maximum CS value has the most successful performance.

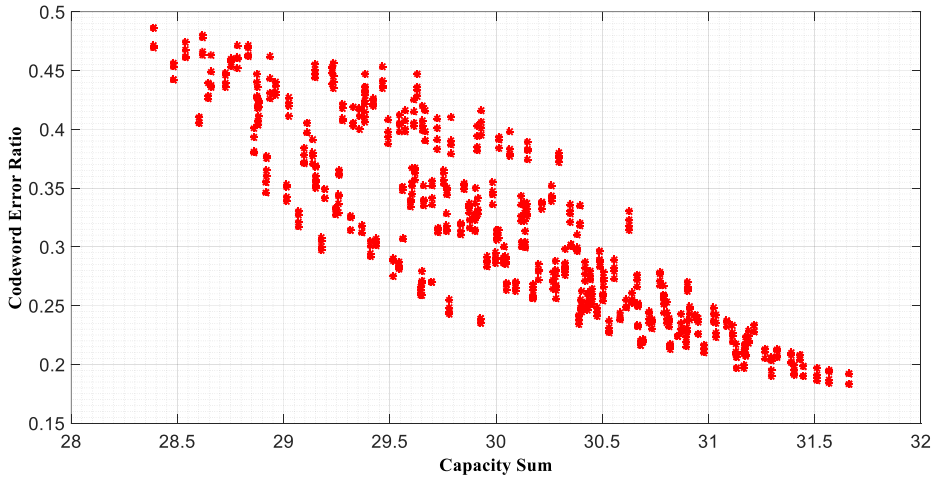


Figure 3.10 Codeword Error Ratio (CER) performance versus Capacity Sum at Single-FG BP decoder for the (64, 32) adaptive polar code over BEC(0.35).

In **Table 3.7**, we show the elements of two equi-CS sets, which are the highest and lowest ones. These eight factor graphs of the highest and lowest equi-CS sets also belong to the highest and lowest equi-FV sets, respectively.

Table 3.7 Codeword error ratio (CER) performance comparison of minimum and maximum equi-CS sets at single-FG BP decoder for the (64, 32) adaptive polar codes over BEC(0.35).

	FV	CS	SON	CER
MinFV&MinCS	52	28.3858	123456	0.486
	52	28.3858	123546	0.486
	52	28.3858	132456	0.469
	52	28.3858	132546	0.471
MaxFV&MaxCS	88	31.6594	645231	0.183
	88	31.6594	645321	0.192
	88	31.6594	654231	0.183
	88	31.6594	654321	0.192

In **Table 3.7**, 654321 and 645321 FG's from highest equi-CS sets have similar performances (CER=0.192) with each other. Also; 654231 and 645231 FG's have parallel performances (CER=0.183) with each other. In fact, some of them have slightly better CER performance than the reference factor graph. Moreover; performance of 132546-132456 FG's and CER performance of 123546-123456 FG's are parallel as pairs and approximately equal to 0.47 and 0.48, respectively. Though, these are lower than ones of FG's from highest equi-CS set. That is; the factor graphs with higher FV and CS values seem to have more successful single-FG performance.

Finally, we observe the single-FG performances of all possible factor graphs, which are sorted according to their stage order numbers in **Figure 3.11**. It can be seen that although there are some jump off points, while SON values are increasing, single-FG performances of factor graphs are improving in general. They achieve CER values between 0.486 and 0.183. Moreover; there are grouped data points, which are shown in different colors and each group belongs to factor graphs which start with same stage number. This shows that the first stage number of FG's is quite important to determine those factor graphs' performance.

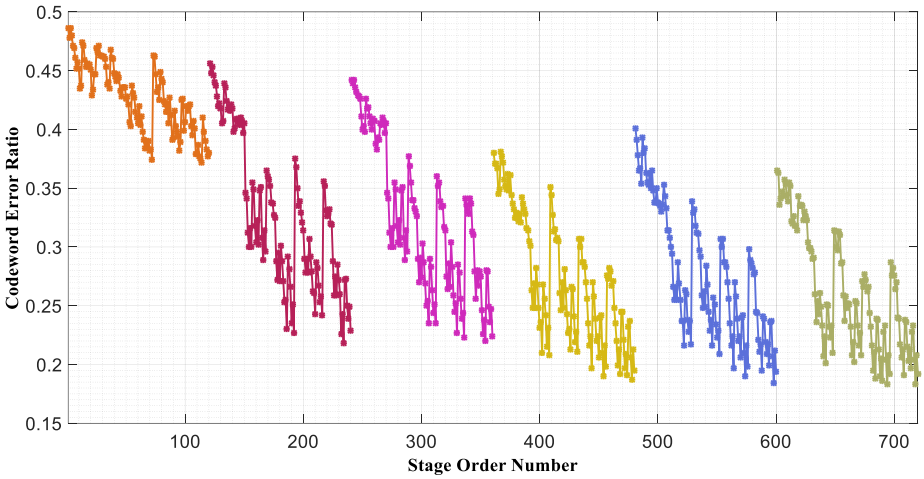


Figure 3.11 Codeword Error Ratio (CER) performance versus Stage Order Number (SON) at single-FG BP decoder for the (64, 32) adaptive polar code over BEC(0.35).

We also attempt to compute the correlation coefficients between codeword error ratio (CER) performances and three parameters (CS, FV and SON) of factor graphs. As it has been done in Section 3.1.4, we use Equation 3.1, while calculating the correlation coefficients r_{xy}. For a (64, 32) polar code over BEC(0.35), correlation coefficients between "CER and the number of frozen variables FV", "CER and the capacity sum CS", and "CER and the stage order number SON" are found as -0.8767, -0.8732 and -0.7269, respectively. As expected, all correlations are highly negative; and one observes that among the three parameters CS, FV and SON, the CER performance seems to be more correlated with the FV and CS values.

Since the codeword error ratio performance simulation for each one of the $n!$ factor graphs is quite burdensome for larger blocklengths, we conduct this experiment only for the (64, 32) polar code.

3.2 Multiple-FG Belief Propagation Decoding

A multiple M-FG decoder is composed of $M > 1$ cascaded factor graphs (FG's). Each FG attempts to decode the N-bit words, which can not be decoded by the previous FG's decoding process. So, the first FG of the M-FG decoder attempts to decode all received words; whereas the second factor graph only decodes the words remaining from the first FG. Since we study the performance over binary erasure channels, "successful decoding" means that all erased bits of the received N-bit word are correctly filled. However, "unsuccessful decoding" can also fill some of the erased bits, correctly or incorrectly. There arises the question of how to submit the first FG's output to the second one. Should it be in its original received form or should the estimated bits by the first FG be included in the second FG's input?

In order to make a clear distinction between these two cases, we propose using the words "independent" and "dependent". In "independent M-FG decoding", each FG decoder passes its undecoded words to the successive FG in their original received forms; whereas, in "dependent M-FG decoding", undecoded words are passed to next factor graph with its (correctly or incorrectly) estimated bits from the previous FG.

In this section, BP decoding performance of the (128, 64) adaptive polar code over a BEC(0.35) is examined except for Section 3.2.8, where other block lengths $N = 2^6, 2^7, \dots, 2^{10}, 2^{11}$ are considered; and Section 3.2.9, where the erasure rates are changed as $\varepsilon = 0.5, 0.45, \dots, 0.15$. Codeword error ratio-CER (or sometimes the bit error ratio- BER) for $\varepsilon \geq 0.35$ is measured over 1000 codewords, but for smaller ε's of Section 3.2.9, either 10,000 or 100,000 codewords are used. In all decoding simulations, "perfect knowledge-based stopping condition" is employed to stop the iterations, whose number can at most be equal to the preset value of 20 iterations. Independent M-FG decoding is used only in Section 3.2.1 for comparison with dependent M-FG decoding; all other simulations are made using dependent M-FG decoding.

3.2.1 Dependent versus Independent M-FG BP Decoding

For the (128, 64) adaptive polar code, there are $7! = 5040$ different factor graphs. To observe the performance difference between dependent/independent M-FG decoders, arbitrarily chosen M factor graphs can be used. In **Figure 3.12**, we choose seven FG's as cyclically left-shifted forms of the reference factor graph, RFG, and indicate the codeword error ratio (CER) performances of the independent and dependent M-FG BP decoders, for $1 < M < 7$.

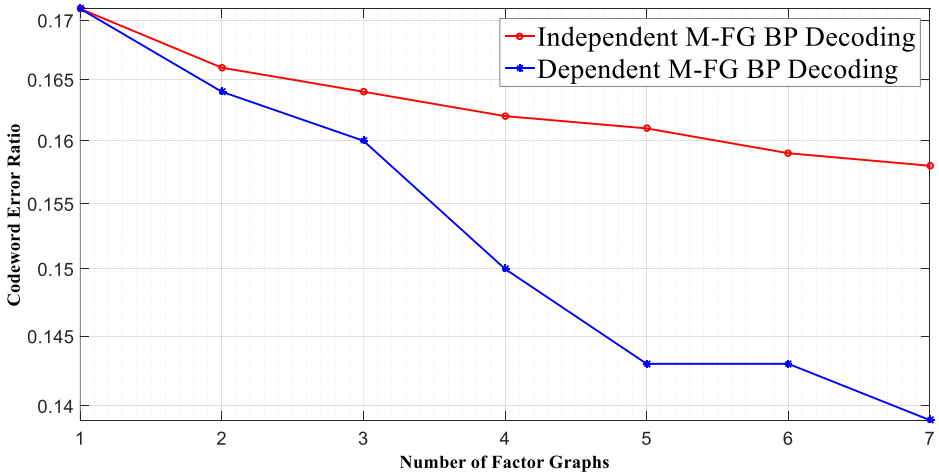


Figure 3.12 Codeword error ratio (CER) comparison of the independent and dependent 7-FG belief propagation decoders for the (128, 64) adaptive polar code over BEC(0.35).

As can be seen in **Figure 3.12**, CER performance of the first FG is the same for both the independent and dependent 7-FG BP decoders, because they start with the same FG (that is the RFG). The second, third and further FG's of the dependent M-FG BP decoder have better CER performances than those of the independent one. At the output of the 7^{th} FG, performance advantage of the dependent 7-FG BP decoder over the independent 7-FG BP decoder is approximately 2%. Although this is not a remarkable gain, dependent M-FG decoders are observed to keep this advantage over the independent case with different sets of FG's and erasure patterns, as well.

To observe the effect of the erasure pattern on both dependent and independent decoder performances, we select four factor graphs (7654321, 5432176, 3217654, and 1765432 from the set of cyclically shifted forms of the reference factor graph. 4-FG BP decoding of the (128, 64) adaptive polar code is repeated 10 times with 10 different erasure patterns, each containing 1000 codewords, over BEC(0.35). Resulting performances for 10 trials are given in **Figure 3.13**. Dependent decoders are shown with dashed lines and independent ones with solid lines, both in the same color for each trial.

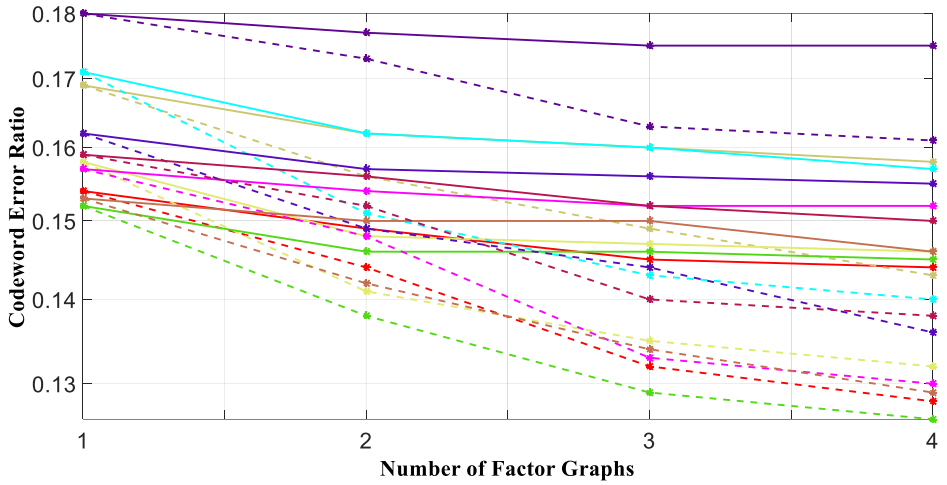


Figure 3.13 Codeword error ratio (CER) comparison of the independent and dependent 4-FG belief propagation decoders in 10 trials for the (128, 64) adaptive polar code over BEC(0.35).

As observed in **Figure 3.13**, first factor graph's performances are the same for both the independent and dependent *M*-FG BP decoders over same erasure pattern, since they both start with the RFG. However; because of the variability of the erasure pattern, difference between the CER performances of the first FG's is 2.8%; whereas the performance of the most successful dependent 4-FG decoder also changes by approximately 3.5% among all erasure patterns. As observed previously in Figure 3.12, CER performance gain of the independent and dependent decoding gets larger as *M* increases. Differences of 4-FG independent/dependent decoder performances are at least 1.2% and maximum 2.2% in **Figure 3.13**. That is; the difference between two decoders can change according to the employed the erasure pattern. However,

the dependent M-FG BP decoder seems consistently performing slightly better than the independent one in all experiments. This observation also implies that partially filled erasures of the undecoded word by the dependent decoder are mostly correct.

Additionally, in dependent decoding, since the undecoded words in estimated forms are passed with less number of erasures, the next factor graph attempts to decode less number of bits. Hence, the time spent for dependent M-FG BP decoding is less than that of the independent one. For instance; in **Figure 3.12**, dependent M-FG BP decoding processing time (181 sec) is approximately 3 times shorter than that of the independent one (578 sec) on an Intel, Core i5-5200U, with CPU @2.20GHz and 6 GB RAM. Therefore, we continue with dependent *M*-FG BP decoding in the rest of this work.

3.2.2 Changing the Order of Factor Graphs in Dependent *M*-FG BP Decoding

As we see in the previous section, in dependent M-FG BP decoding, previous factor graphs' estimations positively affect the performance of the next FG. We now ask the question of how one should arrange positions of the M given factor graphs of the dependent decoder with respect to each other for better performance.

For the (128, 64) adaptive polar code, there are $7! = 5040$ different factor graphs. To observe the effect of the difference in orders of factor graphs, an arbitrarily chosen set of 7 FG's in two different orders can be used in dependent 7-FG BP decoding. We choose the set as the RFG and its cyclically shifted forms, and its different orderings are produced by cyclically left or cyclically right shifting. Codeword error ratio (CER) performances of dependent M-FG BP decoders with two differently ordered FG sets are indicated for $1 < M < 7$, in **Figure 3.14**.

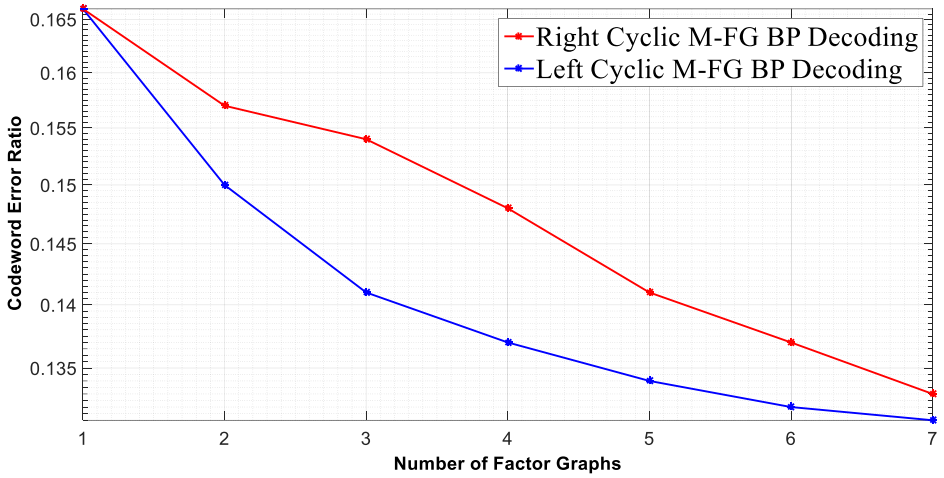


Figure 3.14 Codeword error ratio (CER) comparison of the dependent 7-FG BP decoders with cyclically right and left-shifted RFG sets for the (128, 64) adaptive polar code over BEC(0.35).

As can be seen in **Figure 3.14**; CER of both sets start with the 16.6% performance of the reference factor graph, which is the first FG for both sets; and then starts to differ in the 2^{nd} FG, where the FG with SON=6543217 outperforms the one with SON=1765432. For this simulation, the cyclically left-shifted set has slightly better performance (approximately 0.2% at the 7^{th} FG) than the cyclically right-shifted set.

When we repeat this experiment 24 times for a 4-FG set (with SON's 7654321, 5764321, 3765421 and 1765432) ordered in $4! = 24$ different ways, we get the CER performances of the 24 dependent 4-FG BP decoders, depicted in **Figure 3.15**.

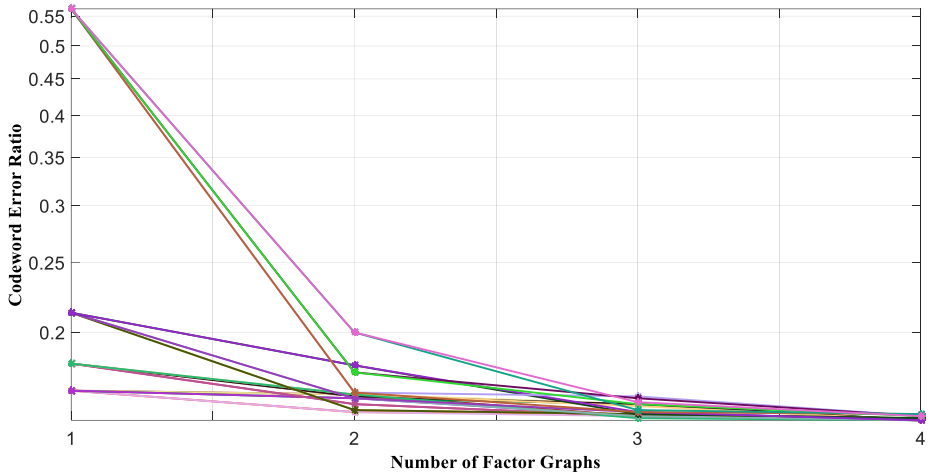


Figure 3.15 Codeword error ratio (CER) comparison of the dependent 4-FG BP decoders with a fixed set of 4 FG's in 24 different orders for the (128, 64) adaptive polar code over BEC(0.35).

In 24 permutations of 4 factor graphs, there are 4 different starting points (RFG being the best) and $3! = 6$ different sets in each group. One observes that there is no noticeable difference in CER performances, hence the order of the FG's in the set does not seem having a considerable effect. It is also observed that 4-FG performances given in Figure 3.15 are better than the 7-FG performances of Figure 3.14. This can only be the result of the difference in respective erasure patterns. So, it is also motivating to compare their performance under the same erasure pattern, as we have attempted in Section 3.2.6. Although, there is no significant effect of the order of FG's on the performance, we will sort factor graphs by descending order of their stage order numbers in M-FG BP decoders.

3.2.3 Twice Repeated M-FG BP Decoding versus $2M$-FG Decoding

We now compare the dependent decoding performance of a twice repeated set of M different factor graphs cascaded as $FG_1FG_2\ldots FG_MFG_1FG_2\ldots FG_M$, with the performance of $2M$ different FG's connected as $FG_1FG_2\ldots FG_MFG_{M+1}\ldots FG_{2M}$;

where the repeated M-FG employs the same M factor graphs twice and but the $2M$-FG decoder has its last M FG's different from the first M FG's.

To observe the effect of twice repeated M-FG BP decoding, arbitrarily chosen M factor graphs can be used. We choose $M = 3$ and in the first set, use the RFG, 7654321, and its two cyclically left-shifted forms 6543217 and 5432176 for the repeated 3-FG BP decoding. For comparison with the 6-FG decoder, 3 more FG's from the cyclically left-shifted forms of RFG; i.e., 4321765, 3217654, 2176543, are added to the first set.

In **Figure 3.16**, codeword error ratio (CER) performances of two sets of factor graphs are indicated for $M = 3$. One observes that CER performance of 6-FG BP decoding is better than twice repeated 3-FG BP decoding.

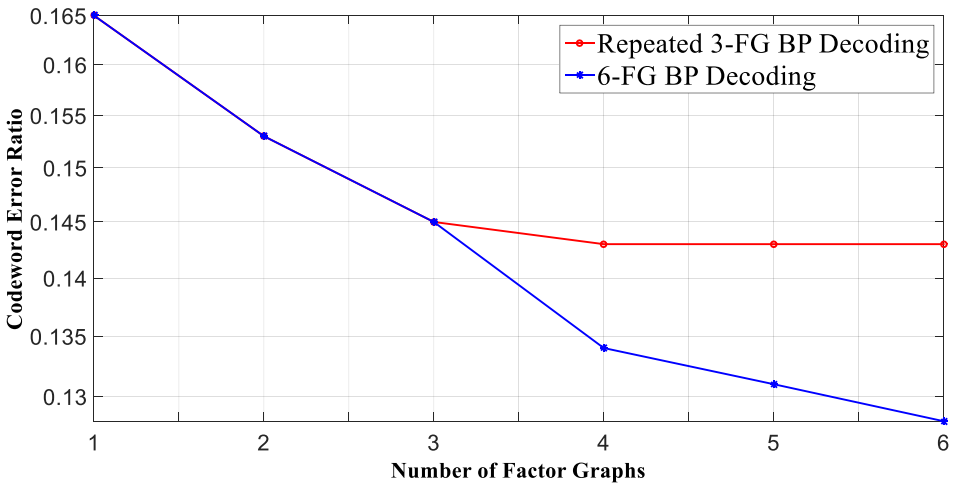
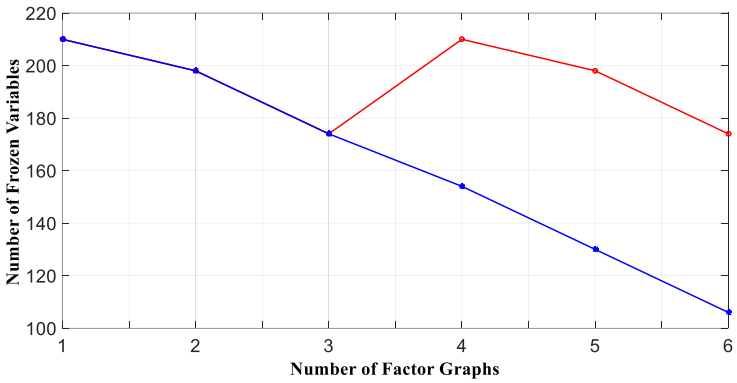


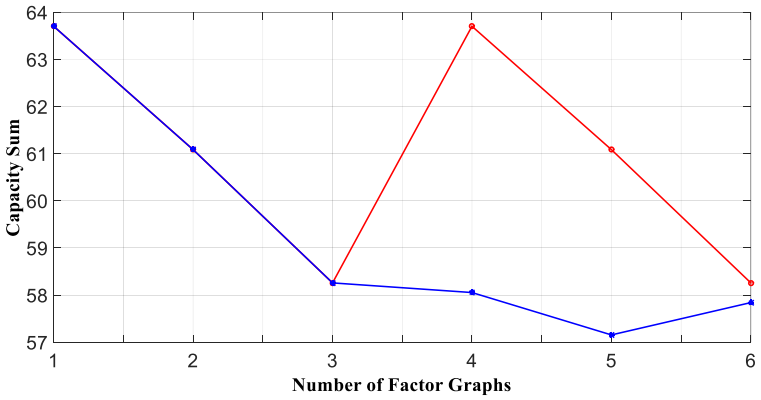
Figure 3.16 Codeword error ratio (CER) comparison of 6-FG and twice repeated 3-FG dependent BP decoders for the (128, 64) adaptive polar code over BEC(0.35).

At the beginning, CER performances are the same because the first 3 FG's of both sets are the same. Repeated 3-FG shows a little progress when the RFG is used as the fourth decoder, but the 5^{th} and the 6^{th} FG's do not make any contribution to CER performance. However; 6-FG decoder has a gradual performance improvement on each FG decoder. Performance advantage of 6-FG BP decoding over repeated 3-FG

BP decoding is approximately 1.5% for this specific set of FG's and erasure pattern. This value may vary with different sets of factor graphs.



(a)



(b)

Figure 3.17 (a) Number of frozen variables (FV) and **(b)** capacity sum (CS) comparison of 6-FG's and twice repeated 3-FG's for the (128, 64) adaptive polar code over BEC(0.35); lower curves corresponding to 6-FG's.

Figure 3.17 demonstrates the number of frozen variables (FV) and capacity sums (CS) of each FG. One can observe that after the third FG, FV and CS values of the repeated 3-FG and 6-FG decoders start to differ. FV values of the repeated 3-FG decoder vary in [174, 210]; whereas those of the 6-FG decoder occupy a larger interval, [106, 210]. Similarly, CS values of the repeated 3-FG decoder differ between 58.3 and 63.7; and those of the 6-FG decoder vary in a larger range,

between 57.2 and 63.7. We observe that the set of 6 factor graphs with larger range of FV and CS values is more successful for dependent M-FG BP decoding.

When we repeat this experiment with different sets for $M = 4$, we pick the RFG, 7654321, and 3 more FG's (6543217, 5432176 and 4321765) for the first set. We arbitrarily select 4 more FG's (3217654, 2176543, 1765432 and 1234567) for the second set. In **Figure 3.18**, CER performances of these two sets of FG's are indicated. One observes that CER of 8-FG BP decoding is better than twice repeated 4-FG BP decoding.

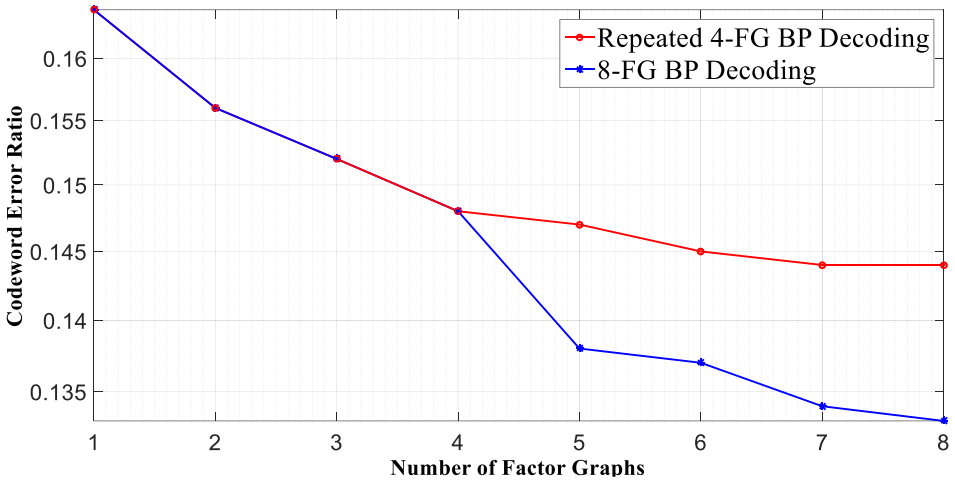


Figure 3.18 Codeword error ratio (CER) comparison of 8-FG and twice repeated 4-FG dependent BP decoders for the (128, 64) adaptive polar code over BEC(0.35).

In **Figure 3.18**, two sets naturally have the same CER's in their first 4 identical FG's. Difference starts at the 5th FG of the 8-FG decoder, which makes larger contribution to the decoding process than that of the twice repeated 4-FG decoder. Similarly with previous experiments, FG's of second set have more different FV and CS values than ones of the first set. Thus; performance of 8-FG BP decoding is approximately 1.2% better than twice repeated 4-FG BP decoding (for this specific set of FG's and erasure pattern).

Above experiments imply that using different FG's with different FV and CS values may lead to better performance.

3.2.4 M Factor Graphs with Similar/Different FV and CS Values

Choosing a well-performing set of factor graphs is important in M-FG BP decoding. Because, an M-FG BP decoder consists of M cascaded FG's, where each FG's input is the output of the previous one. Since each FG sequentially tries to decode the remaining undecoded words from the previous one, and each FG has a different amount of contribution to the total decoding process depending on the specific erasure pattern; the choice of a compatible set of FG's is significant in M-FG BP decoding.

As it is mentioned in previous sections, while choosing a factor graph set for an M-FG BP decoder, FV and CS values of these FG's are noteworthy parameters. As we know from previous sections, single-FG decoders with high FV and CS values have better performance than those with low FV and CS values. However; when a set of M factor graphs is selected for M-FG BP decoding, the best ones may not perform well together. Since the factor graphs should work as a team, choosing factor graphs that complement each other may contribute more to the decoding process.

To observe the effect of different sets of factor graphs on the codeword error ratio of the 7-FG decoder, we choose 3 different sets of factor graphs:

i) The first set consists of 7 factor graphs, which are the first 7 elements of the stage order number-sorted (SON-sorted) FG list. As we have seen in previous sections, SON and single-FG performance have positive correlation. For the (128, 64) adaptive polar code, there are 7 stages. Thus; $7! = 5040$ different SON's can be sorted by descending order, to have a SON-sorted FG list. The first element of this list corresponds to 7654321, which is the reference factor graph (RFG) used for encoding.

ii) The second set consists of 7 factor graphs, which are the ones with the highest CS values in 7 different highest equi-FV groups. For the (128, 64) adaptive polar code over BEC(0.35), we get 28 different equi-FV groups and each of them contains different number of factor graphs.

iii) The third set consists of factor graphs which are all cyclically left-shifted forms of the RFG, which may be called the "cyclic set".

SON values corresponding to these three sets are given in **Table 3.8**

Table 3.8 Stage order numbers (SON's) of the three sets used in this section.

Set (*i*)	7654321	7654312	7654231	7654213	7654132	7654123	7653421
Set (*ii*)	7654321	6475321	4675321	6547321	4657321	4567321	3674521
Set (*iii*) Cyclic Set	7654321	6543217	5432176	4321765	3217654	2176543	1765432

In **Figure 3.19**, codeword error ratio (CER) performances of these three sets of factor graphs are indicated, for $1 < M < 7$. Also; in **Figure 3.20**, the number of frozen variables (FV) and capacity sum (CS) values of each factor graph are demonstrated.

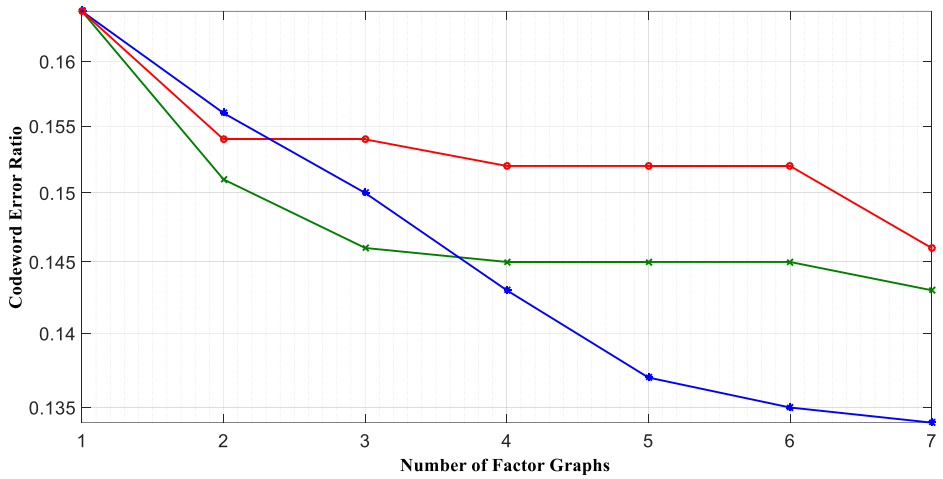
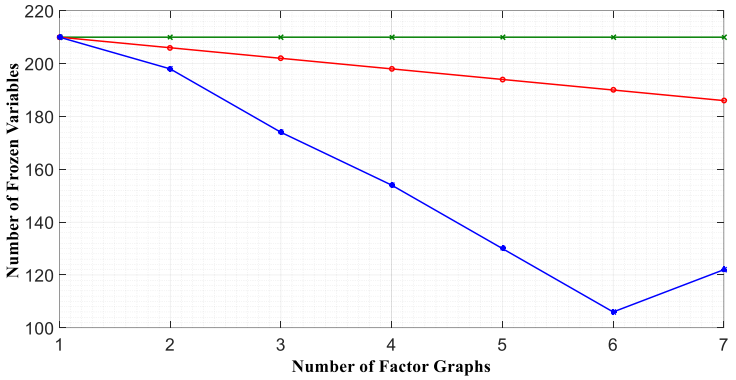


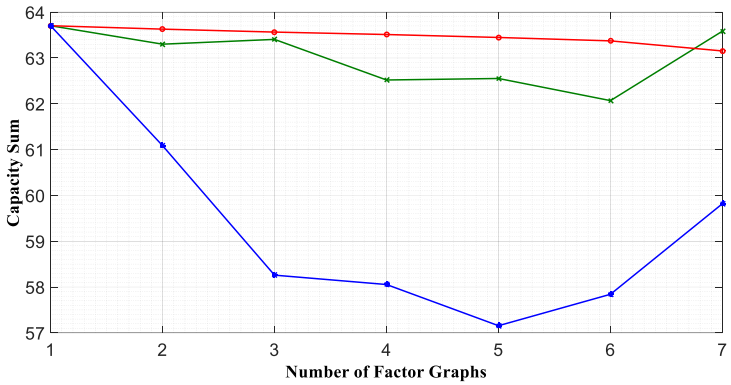
Figure 3.19 Codeword error ratio (CER) performance for the (128, 64) adaptive polar code over BEC(0.35) of three 7-FG decoder sets, *i)* green curve: 7 top FG's of the SON-sorted list, *ii)* red curve: the highest CS-valued FG of 7 highest equi-FV groups, and *iii)* blue curve: 7 FG's generated by cyclic shifts; i.e., the cyclic set.

Figure 3.19 shows that 7-FG CER performances of the first and second sets are similar, but the third one is more successful than others. Also; one can observe that

their starting CER performances are the same (almost 16.5%); because, the first element of all sets is the reference factor graph, RFG. CER performances of the first, second and third sets are approximately 14.6%, 14.4% and 13.4% respectively, at the output of the 7^{th} FG. The performance of the third set seems to be more than 1% superior to the first and second sets.



(a)



(b)

Figure 3.20 (a) Number of frozen variables (FV) and **(b)** Capacity sum (CS) comparison of the three sets in Figure 3.19 for the (128, 64) adaptive polar code over BEC(0.35); *i)* green curve: 7 top FG's of the SON-sorted list, *ii)* red curve: the highest CS-valued FG of 7 highest equi-FV groups, and *iii)* blue curve: 7 FG's generated by cyclic shifts; i.e., the cyclic set.

Figure 3.20 indicates that factor graphs of the first and second sets at each step have similar FV and CS values; whereas those of the third set are quite different. As explained in Section 3.1, FV and CS values of factor graphs are important parameters that determine their single-FG BP decoding performances. High FV and high CS

valued single-FG decoders are superior in performance to those with low FV and CS values. On the other hand, **Figure 3.19** and **Figure 3.20** show that the CER performance of the 7-FG decoder is better within the third set that has distributed FV and CS values; i.e, small CS & FV valued FG's as well, in addition to the high (FV & CS) valued FG's. It seems that in team work, the factor graphs of the third set with distinct FV and CS values are more compatible with each other. So, while choosing the sets of $M > 1$ factor graphs for M-FG decoders, the experiments of this part suggest FG sets with wide intervals of FV and CS values. However, we conjecture that in the attempt of forming sets of M FG's with better performance, it may be more practical to employ the SON values of the FG's rather than computing the related CS and FV parameters. We present a set-choice algorithm depending on SON values for $M = \lceil n/2 \rceil$; i.e., the smallest integer greater than $n/2$ in Section 3.2.7.

3.2.5 Getting Similar Performance with Less Number of Factor Graphs

As we have mentioned in Section 3.2.4, choosing a compatible set of factor graphs increases the performance of M-FG BP decoding. Cyclical shifting of the RFG stages, as used in [Korada, 2009], seems to be one of the successful methods to generate a compatible M-FG set, with $M = n$. We call the corresponding set of n FG's, the "cyclic set". We have observed in previous sections that compatible sets like the cyclic set have wide range of CS and FV values.

At this point, we ask the question of "Can we get similar performance with less number of compatible factor graphs?" Thus; in this section, we study on getting similar performance with the cyclic set by using less number of FG's.

For the (128,64) polar code, we generate a set of 4 FG's, as the subset of the cyclic set, by picking up only the FG's starting with odd-numbered stages; i.e., {7654321, 5432176, 3217654, 1765432}. Corresponding range of CS and FV values is still wide in this 4-FG subset. In **Figure 3.21**, codeword error ratio performance of the

cyclic set, for $1 < M < 7$, and that of the 4-FG subset, for $1 < M < 4$, are indicated.

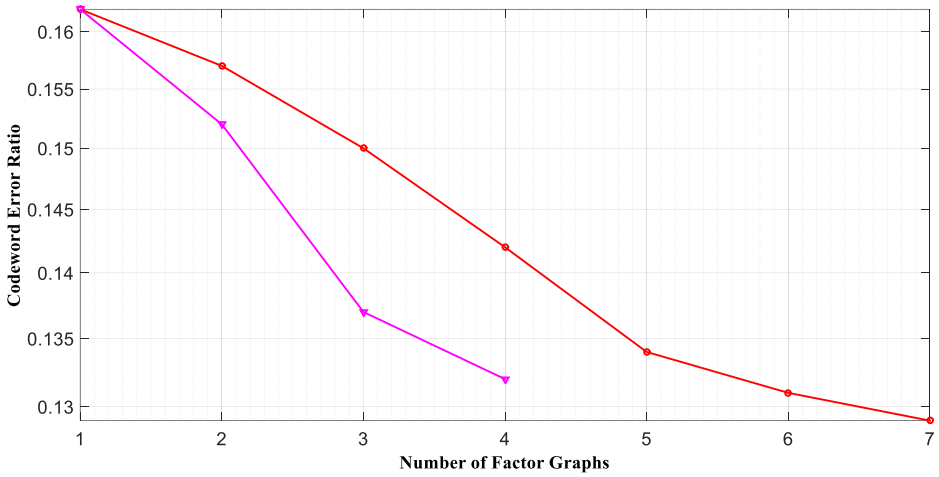


Figure 3.21 Codeword error ratio (CER) performance for the (128, 64) adaptive polar code over BEC(0.35) of two decoder sets, *i)* red curve: cyclic set with 7 FG's, *ii)* purple curve: 4 FG's starting with odd-numbered stages, chosen from the cyclic set.

In **Figure 3.21**, since the first elements of both sets are the RFG, two sets of decoders start with the same CER of approximately 16.2%. The third and fifth FG's of the cyclic set are the same as the second and third FG's of the 4-FG subset, but their CER performances differ by 0.2% & 0.3%, respectively, in favor of the cyclic set because of its intermediate FG's starting with even-numbered stages. Although two sets have different number of factor graphs, CER performances are so close at the last FG decoders of both sets. Their performance difference is approximately 0.3% at the last FG.

However; if we have arranged the second set by choosing the first four factor graphs of the first set instead of the above selection, as can be seen in **Figure 3.21**, its performance at the output of the 4'th FG would be approximately 1.3% worse than the output of the 7-FG cyclic set, which could be reduced four times (to 0.3%) with the 4-FG subset of the above selection. So, keeping the range of CS and FV values wide enough, it seems possible to decrease the size of the M-FG decoder from n to $\lceil n/2 \rceil$ without any significant loss in performance.

In Table 3.9, we also tabulate the SON's of the first 4 FG's of the cyclic set, together with those of the 4-FG subset having the performance shown in **Figure 3.21**; for further reference.

Table 3.9 Stage order numbers (SON's) of the 4-FG sets used in this section (4-FG subset in the second row has 1% better CER).

First 4 FG's of the Cyclic Set	7654321	6543217	5432176	4321765
4-FG Subset Starting with Odd-Numbered Stages	7654321	5432176	3217654	1765432

It is observed in **Table 3.9** that stage order numbers of the subset in the second row are varying in a wider range than those in the first row (SON range in the second row $= 7654321-1765432 > 765431- 4321765 =$ SON range in the first row). One may also conjecture that, in case of sets with the same SON range, variety at individual stage positions may provide more successful M-FG decoder performance as will be discussed in Section 3.2.6.

3.2.6 Relation between the Performance and Stage Order Numbers of the Factor Graph Set

In order to form a compatible FG set for *M*-FG decoding, we have seen that FG variety is needed rather than similar FG's. Previously, we have related the required variety with the FV and CS values of factor graphs within the *M*-FG set. If the compatible set selection can also be related to the stage order numbers of the FG's as we have mentioned at the end of Section 3.2.5, a more practical method can be derived.

For another evaluation, we compare the performance of the 4-FG cyclic subset considered in Section 3.2.5, with a trial 4-FG set having almost fixed output stages. Input stages of the trial set are also 7, 5, 3, 1 as those of the 4-FG cyclic subset, but output stages are fixed in three elements of the trial set, as emphasized in **Table 3.10** by bold letters. One may observe that the first and last elements of the two 4-FG sets are the same, and the difference is only in the second and third FG's.

Table 3.10 Stage order numbers (SON's) of the 4-FG sets used in this section (4-FG subset in the second row has 1% better CER).

Trial 4-FG Set	7654**321**	5764**321**	37654**21**	1765432
4-FG Cyclic Subset Starting with Odd-Numbered Stages	7654321	5432176	3217654	1765432

In **Figure 3.22**, codeword error ratio performances of the two sets of factor graphs are indicated, for $1 < M < 4$. One observes that the new 4-FG set performs approximately 1.3% worse than the 4-FG cyclic subset. So, we understand that while arranging a compatible set for the M-FG decoder, variety only in the first stage positions is not enough. When all stage positions are occupied by a variety of stages, corresponding set can be inferred to have more successful CER performance.

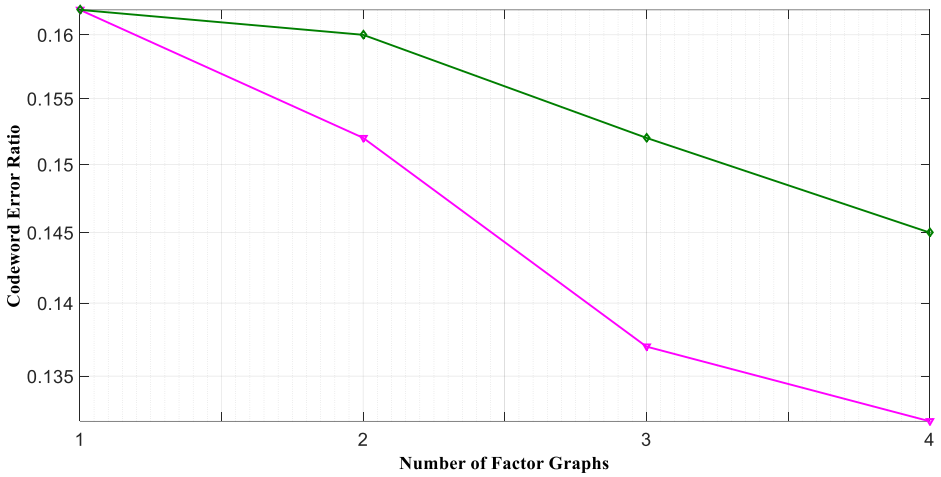


Figure 3.22 Codeword error ratio (CER) performance for the (128, 64) adaptive polar code over BEC(0.35) of two decoder sets, *i)* green curve: trial set with 4 FG's "7654321-5764321-3765421-1765432", *ii)* purple curve: 4-FG cyclic subset starting with odd-numbered stages, chosen from the cyclic set

Until this point, the 4-FG cyclic subset with factor graphs {7654321, 5432176, 3217654, 1765432} seems to be the most successful 4-FG decoder for the (128, 64) polar codes over BEC(0.35). Corresponding CS, FV, SON values as well as the distribution of stages to individual positions have a wide ranged variety. For another comparison, we propose a set choice algorithm to derive a new set, producing factor graphs with wide range of required properties.

3.2.7 Set Choice Algorithm for $M = \lceil n/2 \rceil$ Factor Graphs

To ensure the variance of the stage numbers at each stage, and the variety between stage passages as well, we offer a simple algorithm to generate a compatible FG set with $M = \lceil n/2 \rceil$. We then compare the performance of this compatible set with that of the 4-FG cyclic subset having the factor graphs {7654321, 5432176, 3217654, 1765432}.

"Set Choice" algorithm explained below uses the reference factor graph as the first element of the set. Other factor graphs are produced by shifting the odd indexed stages of the RFG left-cyclically and its even indexed stages right-cyclically. In this way, factor graphs have been sorted by descending order of their stage order numbers (SON's) as generalized in Section 3.2.2. Set Choice algorithm provides a variety of stages at each stage position. Moreover; it delivers diversity for stage passages between successive state positions, which is missing in the cyclic set of Korada [Korada, 2009] that conserves the successive stage numbers. Due to the nature of the BP decoding algorithm, each stage is an individual decoder, so variety at successive stage transitions can improve the decoding performance of the M-FG decoder.

Set Choice Algorithm (for $M = \lceil n/2 \rceil$ FGs) = leftRightCyclicSetFGs()

Input:
N : Blocklength
Output:
FGs : Factor graphs of the M-FG Decoder

```
Set n to log base 2 of N
Set M to round to upper integer of n/2
Set RFG to row vector with size n

for loopcounter = n to 1
        RFG[n+1-loopcounter] = loopcounter
endfor

Set i to 1
Set i'th row of FGs to RFG

for loopcounter = 1 to M-1
        Set odd index stages of tempFG = odd index stages of RFG i times left cyclic shifted
        Set even index stages of tempFG = even index stages of RFG i times right cyclic shifted
        Set i to i+1
        Set i'th row of FGs to tempFG
endfor
```

By using the Set Choice algorithm, we generate the factor graph set, {7654321, 5236147, 3412765, 1674523}. As can be seen in **Table 3.11** stage numbers at specific positions of the factor graphs within the generated set are dissimilar. In **Figure 3.23**, codeword error ratio (CER) performances of two sets of factor graphs are indicated, for $1 < M < 4$.

Table 3.11 Stage order numbers (SON's) of the 4-FG sets used in this section.

4-FG Set Produced by the Set Choice Algorithm	7654321	5236147	3412765	1674523
4-FG Cyclic Subset Starting with Odd-Numbered Stages	7654321	5432176	3217654	1765432

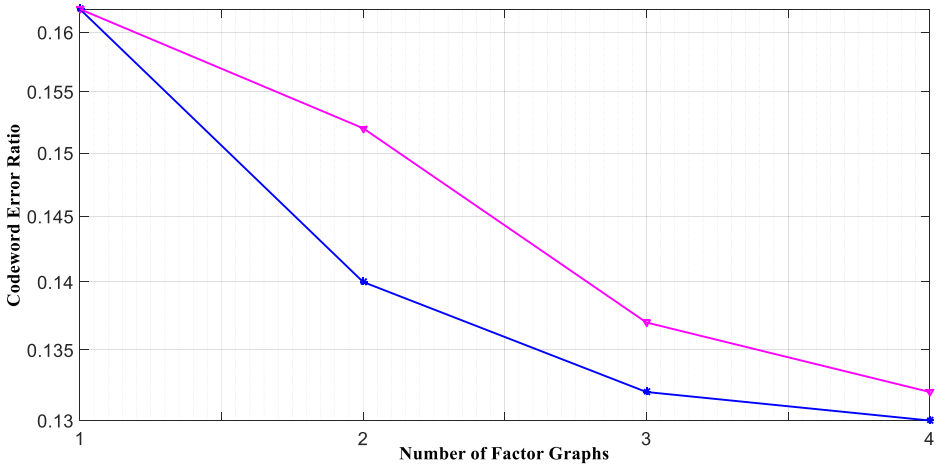


Figure 3.23 Codeword error ratio (CER) performance for the (128, 64) adaptive polar code over BEC(0.35) of two decoder sets, *i)* pink curve (upper one): 4-FG cyclic subset, *ii)* blue curve (lower one): 4-FG set produced by the Set Choice algorithm.

In **Figure 3.23**, since RFG is the first FG in both sets; CER's start with the same value (approximately 16.2%). At the output of 4-FG decoders, performance of the set produced by the Set Choice algorithm is approximately 0.2% superior to first one.

As a second trial, we compare the performance of a compatible set produced by the Set Choice algorithm for the (64, 32) polar code over BEC(0.35), with that of the genie-chosen set in [Doğan, 2015], which consists of factor graphs {643521, 645213,

465312, 624351}. Set Choice algorithm generates the 3-FG set {654321, 412563, 236145} and we add one more FG with SON=165432 (by cyclically right-shifting the RFG) for a fair 4-FG comparison. Doğan states that his 4 factor graphs are chosen according to empirical evidence, and his 4-FG set gives very similar CER performance to the cyclic 6-FG set. In **Figure 3.24**, CER performances of the two sets are shown for $1 < M < 4$. Unlike the previous ones using 1000 erased codewords, in this experiment, codeword error ratio is measured by decoding 10000 codewords with erased bits; in order to get more accurate results regardless of the erasure patterns.

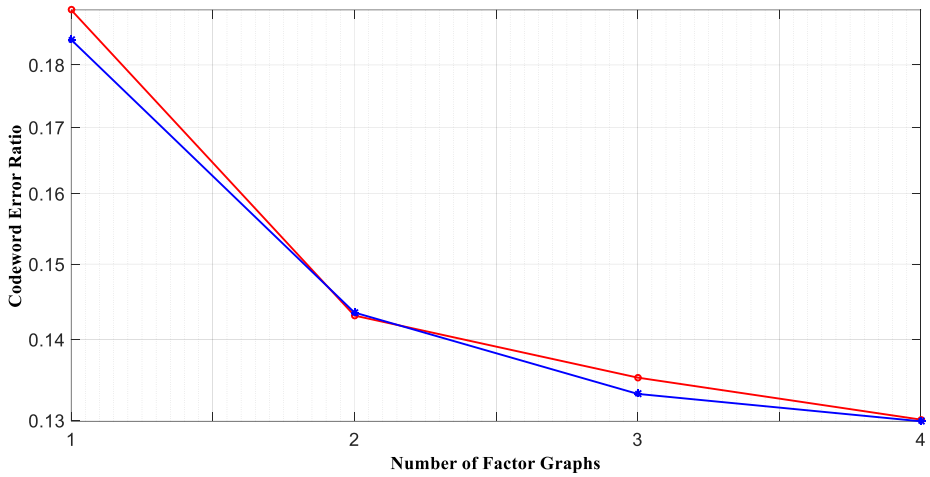


Figure 3.24 Codeword error ratio (CER) performance for the (64, 32) adaptive polar code over BEC(0.35) of two decoder sets, *i)* blue curve (lower one): 4-FG set produced by our Set Choice algorithm, *ii)* red curve (upper one): Genie-chosen set of [Doğan, 2015]

In **Figure 3.24**, since two sets start with different factor graphs, their beginning CER performances are different. Our set, whose first factor graph is RFG, has approximately 18.4% CER performance on the first FG decoder. As expected, CER performance of the first FG decoder of Doğan's genie-chosen set is a little bit worse than ours and approximately equals to 18.9%. However, on the second and third FG decoders, CER performances of both sets are similar. Also; 4-FG decoding performances of both sets are almost identical and approximately equal to 13%. Therefore; it can be said that our set, which is chosen with a systematically approach,

is as successful as Doğan's genie-chosen set, which is generated by assessing the empirical results.

3.2.8 Performance Comparison of M-FG Belief Propagation Decoders with $M = \lceil n/2 \rceil$ and $M = n$

In this section, we compare the performances of M-FG BP decoders, for $M = 6, \ldots, 11$; with $M = n$-FG's cyclically shifted as in [Korada, 2009] and $M = \lceil n/2 \rceil$-FG's generated by the Set Choice algorithm. As we have stated in Section 3.2.7, using the Set Choice algorithm to generate a compatible set of $M = \lceil n/2 \rceil$ FG's provides successful performance results for dependent M-FG BP decoding for the (128, 64) and (64, 32) polar codes. In this section, unlike the previous sections, we also work on different blocklengths, $N = 64, 128, 256, \ldots, 2048$. We select a binary erasure channel with 0.4 erasure rate for all blocklengths, in order to observe the CER performance at larger blocklengths more accurately. Our aim is comparing the sets of size $\lceil n/2 \rceil$ generated by the Set Choice algorithm with the n-FG cyclic decoders in [Korada, 2009]. In each comparison curve, we also include the reduced versions of the n-FG cyclic sets to approximately half size $\lceil n/2 \rceil$. That is; the first set of each figure consists of $n = \log_2 N$ different FG's as cyclically left-shifted forms of RFG and corresponding CER performance is shown by a red curve. The second set of each figure is the reduced version of the first set and shown by a pink curve. The third set is generated by using the Set Choice algorithm and shown by a blue curve.

Codeword error ratio is measured, after 1000 codewords with erased bits are decoded by dependent n-FG and $\lceil n/2 \rceil$-FG decoders; and depicted in Figure 3.25, Figure 3.26, Figure 3.27, Figure 3.28, Figure 3.29 and Figure 3.30 for $n = 6, 7, 8, 9, 10$ and 11 respectively.

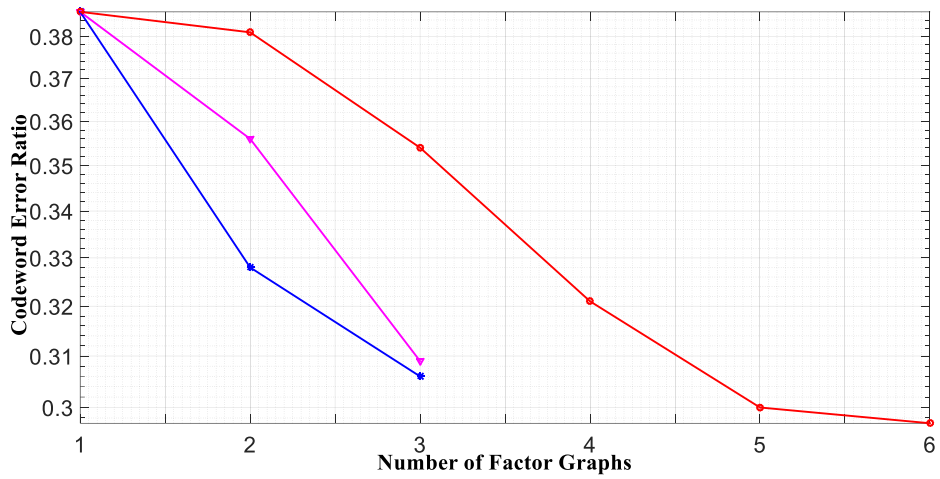


Figure 3.25 Codeword error ratio performance for the (64, 32) adaptive polar code over BEC(0.4) of three M-FG decoder sets, *i)* red curve: 6 FG's by cyclically left-shifting stages of the RFG, *ii)* pink curve: 3 FG's sampled from the first set, *iii)* blue curve: 3 FG's by the Set Choice algorithm.

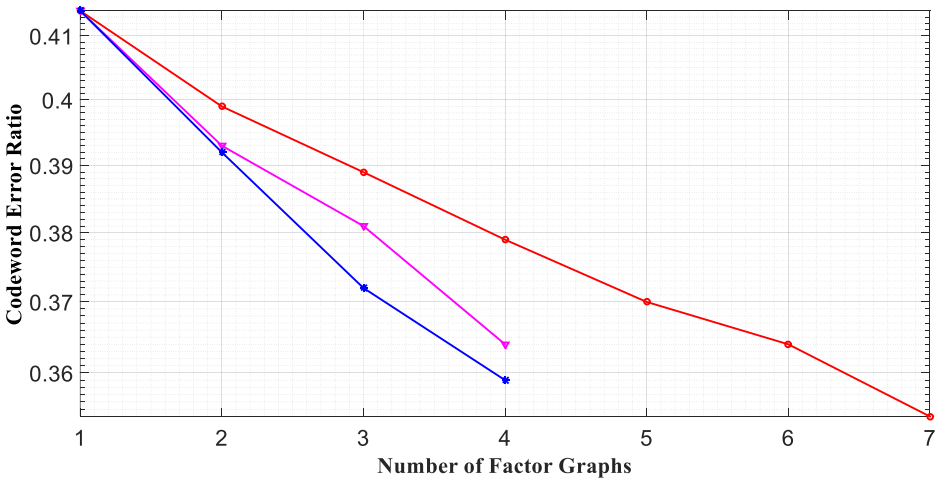


Figure 3.26 Codeword error ratio performance for the (128, 64) adaptive polar code over BEC(0.4) of three M-FG decoder sets, *i)* red curve: 7 FG's by cyclically left-shifting stages of the RFG, *ii)* pink curve: 4 FG's sampled from the first set, *iii)* blue curve: 4 FG's by the Set Choice algorithm.

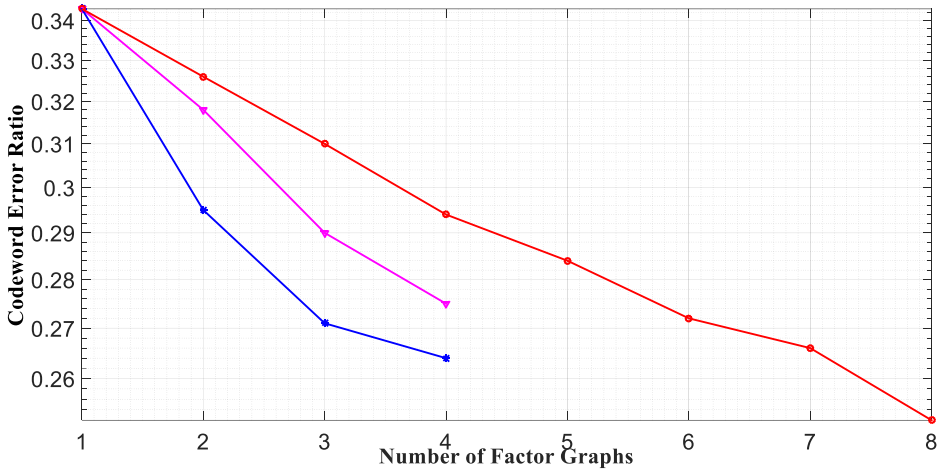


Figure 3.27 Codeword error ratio performance for the (256, 128) adaptive polar code over BEC(0.4) of three M-FG decoder sets, *i)* red curve: 8 FG's by cyclically left-shifting stages of the RFG, *ii)* pink curve: 4 FG's sampled from the first set, *iii)* blue curve: 4 FG's by the Set Choice algorithm.

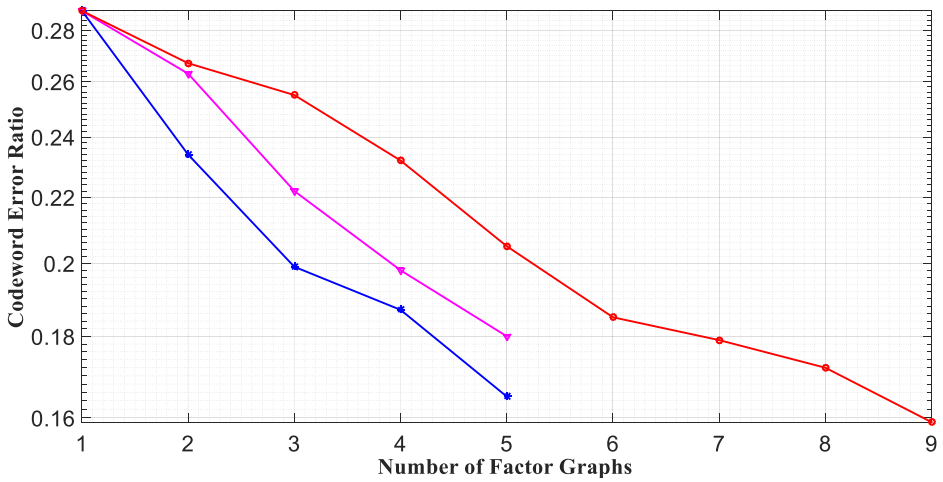


Figure 3.28 Codeword error ratio performance for the (512, 256) adaptive polar code over BEC(0.4) of three M-FG decoder sets, *i)* red curve: 9 FG's by cyclically left-shifting stages of the RFG, *ii)* pink curve: 5 FG's sampled from the first set, *iii)* blue curve: 5 FG's by the Set Choice algorithm.

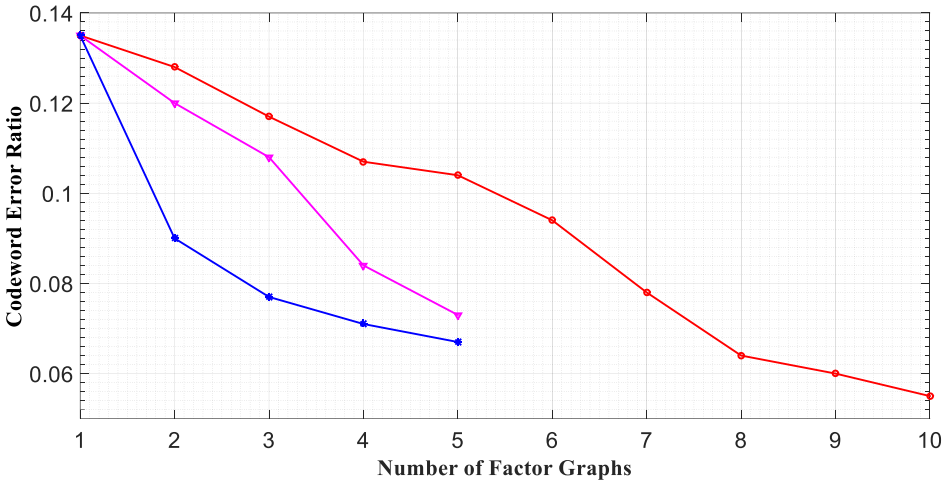


Figure 3.29 Codeword error ratio performance for the (1024, 512) adaptive polar code over BEC(0.4) of three M-FG decoder sets, *i)* red curve: 10 FG's by cyclically left-shifting stages of the RFG, *ii)* pink curve: 5 FG's sampled from the first set, *iii)* blue curve: 5 FG's by the Set Choice algorithm.

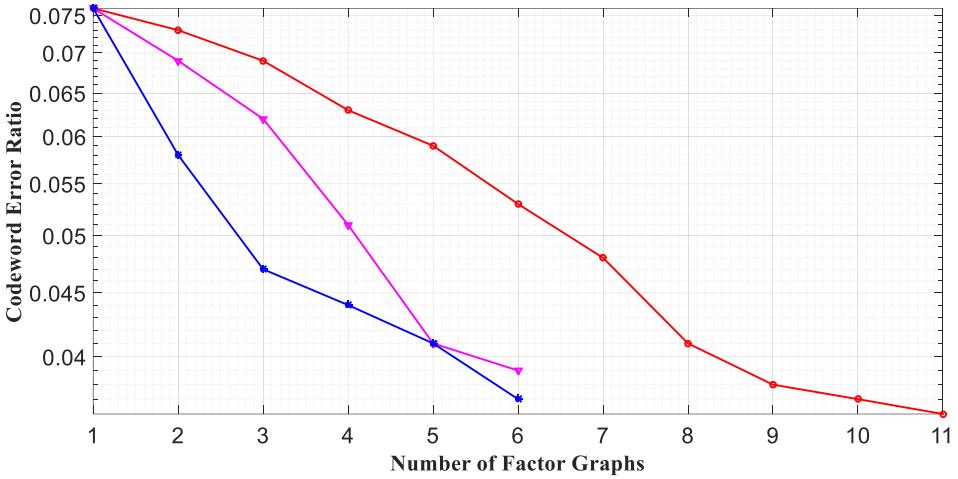


Figure 3.30 Codeword error ratio performance for the (2048, 1024) adaptive polar code over BEC(0.4) of three M-FG decoder sets, i) red curve: 11 FG's by cyclically left-shifting stages of the RFG, ii) pink curve: 6 FG's sampled from the first set, iii) blue curve: 6 FG's by the Set Choice algorithm.

First FG and n-FG decoding performances of the cyclic sets as well as the $\lceil n/2 \rceil$-FG performances of the other two sets are stated in Table 3.11, as extracted from the above six figures (3.25 to 3.30).

Table 3.12 Stage order numbers (SON's) of the $\lceil n/2 \rceil$-FG sets used in Section 3.2.8.

n	**Set Name**	**CER of the First FG (%)**	**CER of *M*-FG (%)**	**1st FG**	**2nd FG**	**3rd FG**	**4th FG**	**5th FG**	**6th FG**
6	3-FG Set Produced by the Set Choice Algorithm	41.4	35.9	654321	412563	236145			
	3-FG Cyclic Subset Starting with Even Stages		36.4	654321	432165	216543			
	6-FG Cyclic Set		35.4	6 FG's by cyclically left-shifting stages of the RFG					
7	4-FG Set Produced by the Set Choice Algorithm	38.6	30.6	7654321	5236147	3412765	1674523		
	4-FG Cyclic Subset Starting with Odd Stages		30.9	7654321	5432176	3217654	1765432		
	7-FG Cyclic Set		29.7	7 FG's by cyclically left-shifting stages of the RFG					
8	4-FG Set Produced by the Set Choice Algorithm	34.3	26.4	87654321	61472583	43218765	25836147		
	4-FG Cyclic Subset Starting with Even Stages		27.5	87654321	65432187	43218765	21876543		
	8-FG Cyclic Set		25.2	8 FG's by cyclically left-shifting stages of the RFG					

n	Set Name	CER of First FG (%)	CER of M-FG (%)	1st FG	2nd FG	3rd FG	4th FG	5th FG	6th FG
9	5-FG Set Produced by the Set Choice Algorithm	28.8	16.5	987654321	725836149	543218967	361492785	189674523	
	5-FG Cyclic Subset Starting with Odd Stages		18.0	987654321	765432198	543219876	321987654	198765432	
	9-FG Cyclic Set		15.9	9 FG's by cyclically left-shifting stages of the RFG					
10	5-FG Set Produced by the Set Choice Algorithm	13.5	6.7	10.98765432.	81694725.10.3	634129.10.785	4523.10.18967	27.10.5836149	
	5-FG Cyclic Subset Starting with Even Stages		7.3	10.987654321	7654321.10.98	54321.10.9876	321.10.987654	1.10.98765432	
	10-FG Cyclic Set		5.5	10 FG's by cyclically left-shifting stages of the RFG					
11	6-FG Set Produced by the Set Choice Algorithm	7.6	3.7	11.10.987654321	927.10.583614.11	74523.10.18.11.69	563412.11.10.987	3816.11.4927.10.5	1.10.11.89674523
	6-FG Cyclic Subset Starting with Odd Stages		3.9	11.10.987654321	987654321.11.10	7654321.11.10.98	54321.11.10.9876	321.11.10.987654	1.11.10.98765432
	11-FG Cyclic Set		3.6	11 FG's by cyclically left-shifting stages of the RFG					

As mentioned in Section 2.3, while N is getting larger, channel capacities $I(W_N^{(i)})$ are polarizing to pure-noisy or noiseless channels and the fraction of noiseless ones is becoming closer to the capacity $I(W)$. Thus, CER's of the decoders in **Table 3.12** decrease as the codeword lengths increase. M-FG decoding performances (for $M = n$ or $\lceil n/2 \rceil$) of polar codes become more successful for larger blocklengths. Although the n-FG decoding performance of the cyclic set is better than $\lceil n/2 \rceil$-FG performances of the other two sets for all considered blocklengths (difference varying between 0.2% and 1.2%), corresponding performance advantage is not worthwhile, when we consider the doubled number of FG's in the decoder.

Furthermore; the $\lceil n/2 \rceil$-FG set, which is generated by using the Set Choice algorithm, consistently has better CER's than CER's at the $\lceil n/2 \rceil$'th FG outputs of both sets, for all considered blocklengths. Therefore; using the Set Choice algorithm to generate a compatible set for lower complexity $\lceil n/2 \rceil$-FG belief propagation decoding appears to be reasonable.

3.2.9 Polar Code Construction and Related Performance for Different Erasure Rates

Bhattacharyya parameters used in adaptive polar code (APC) construction [Arıkan, 2008] depend upon the erasure probability ε of the given binary erasure channel BEC(ε) as explained in Section 2.5, which makes the adaptive polar code design "channel specific". In this section, we demonstrate the dependence of polar code construction on erasure probabilities ε, by designing the (128, 64) adaptive polar code for 11 different values of ε in the interval [0.01 to 0.5] and comparing the corresponding 11 generator matrices to that of the (128, 64) Reed-Muller code.

In Table 3.13, the leftmost column shows the row indices (1, 2 , … , 128) of the 128×128 transformation matrix $F^{\otimes 7}$; the 7th Kronecker power of the base matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Table 3.13 Basis code vectors (and corresponding Hamming weights) of the (128, 64) RM and adaptive polar codes for various ε, the first column indicating the row number in $F^{\otimes 7}$.

	APC											RM
Row Index	**ε=0.5**	**ε=0.45**	**ε=0.4**	**ε=0.35**	**ε=0.3**	**ε=0.25**	**ε=0.2**	**ε=0.15**	**ε=0.1**	**ε=0.05**	**ε=0.01**	
8	8	8	8	8	8	8	8	8	8	8		
12	8	8	8	8	8	8	8	8	8			
14	8	8	8	8	8	8	8	8				
16	16	16	16	16	16	16	16	16	16	16	16	16
20	8	8	8	8	8	8	8	8				
22	8	8	8	8	8	8						
24	16	16	16	16	16	16	16	16	16	16	16	16
28	16	16	16	16	16	16	16	16	16	16	16	16
30	16	16	16	16	16	16	16	16	16	16	16	16
31	16	16	16	16	16	16	16	16	16	16	16	16
32	32	32	32	32	32	32	32	32	32	32	32	32
36	8	8	8	8	8	8	8					
40	16	16	16	16	16	16	16	16	16	16	16	16
44	16	16	16	16	16	16	16	16	16	16	16	16
46	16	16	16	16	16	16	16	16	16	16	16	16
47	16	16	16	16	16	16	16	16	16	16	16	16
48	32	32	32	32	32	32	32	32	32	32	32	32
52	16	16	16	16	16	16	16	16	16	16	16	16
54	16	16	16	16	16	16	16	16	16	16	16	16
55	16	16	16	16	16	16	16	16	16	16	16	16
56	32	32	32	32	32	32	32	32	32	32	32	32
58	16	16	16	16	16	16	16	16	16	16	16	16
59	16	16	16	16	16	16	16	16	16	16	16	16
60	32	32	32	32	32	32	32	32	32	32	32	32
61							16	16	16	16	16	16
62	32	32	32	32	32	32	32	32	32	32	32	32
63	32	32	32	32	32	32	32	32	32	32	32	32
64	64	64	64	64	64	64	64	64	64	64	64	64
68	8	8	8	8	8							
72	16	16	16	16	16	16	16	16	16	16	16	16
76	16	16	16	16	16	16	16	16	16	16	16	16
78	16	16	16	16	16	16	16	16	16	16	16	16
79	16	16	16	16	16	16	16	16	16	16	16	16
80	32	32	32	32	32	32	32	32	32	32	32	32
84	16	16	16	16	16	16	16	16	16	16	16	16
86	16	16	16	16	16	16	16	16	16	16	16	16
87	16	16	16	16	16	16	16	16	16	16	16	16
88	32	32	32	32	32	32	32	32	32	32	32	32
90	16	16	16	16	16	16	16	16	16	16	16	16
91	16	16	16	16	16	16	16	16	16	16	16	16
92	32	32	32	32	32	32	32	32	32	32	32	32
93								16	16	16	16	16
94	32	32	32	32	32	32	32	32	32	32	32	32
95	32	32	32	32	32	32	32	32	32	32	32	32
96	64	64	64	64	64	64	64	64	64	64	64	64
100	16	16	16	16	16	16	16	16	16	16	16	16
102	16	16	16	16	16	16	16	16	16	16	16	16
103	16	16	16	16	16	16	16	16	16	16	16	16
104	32	32	32	32	32	32	32	32	32	32	32	32
106	16	16	16	16	16	16	16	16	16	16	16	16
107						16	16	16	16	16	16	16
108	32	32	32	32	32	32	32	32	32	32	32	32
109									16	16	16	16
110	32	32	32	32	32	32	32	32	32	32	32	32
111	32	32	32	32	32	32	32	32	32	32	32	32
112	64	64	64	64	64	64	64	64	64	64	64	64
114	16	16	16	16	16	16	16	16	16	16	16	16
115									16	16	16	16
116	32	32	32	32	32	32	32	32	32	32	32	32
117										16	16	16
118	32	32	32	32	32	32	32	32	32	32	32	32
119	32	32	32	32	32	32	32	32	32	32	32	32
120	64	64	64	64	64	64	64	64	64	64	64	64
121											16	16
122	32	32	32	32	32	32	32	32	32	32	32	32
123	32	32	32	32	32	32	32	32	32	32	32	32
124	64	64	64	64	64	64	64	64	64	64	64	64
125	32	32	32	32	32	32	32	32	32	32	32	32
126	64	64	64	64	64	64	64	64	64	64	64	64
127	64	64	64	64	64	64	64	64	64	64	64	64
128	128	128	128	128	128	128	128	128	128	128	128	128

The remaining columns of **Table 3.13** show the Hamming weights of the basis vectors of the (128, 64) adaptive polar code for each value of ε. Pink-colored elements of each column are the rows of $F^{\otimes 7}$ that do not belong to the code designed for this specific ε, but they exist in at least one other column corresponding to a different ε. Reed-Muller code construction is not dependent on the erasure rate, and its codewords are demonstrated at the rightmost column.

As seen in Table 3.13, the Reed-Muller code selects all weight-16 rows of $F^{\otimes 7}$, in addition to the weight-64 and weight-32 rows of $F^{\otimes 7}$; but the adaptive polar codes may prefer some weight-8 rows to weight-16 rows, since the corresponding channel capacities are higher. Notice that the adaptive polar codes designed for erasure rates $0.3 \leq \varepsilon \leq 0.5$ are all the same, and they contain 7 weight-8 code vectors (with indices 8, 12, 14, 20, 22, 36, 68) instead of the 7 weight-16 vectors (with indices 61, 93, 107, 109, …) of the RM code. For $\varepsilon = 0.25$, the number of weight-8 rows decreases to 6, by excluding the 68^{th} weight-8 row and including the 107^{th} weight-16 row. While ε reduces, it is observed that the similarity to RM codes increases, and the adaptive polar code designed for $\varepsilon = 0.01$ becomes exactly the same as the RM code. So, one obtains totally seven different polar codes as ε varies between 0.5 and 0.01.

We evaluate the BP decoding performances of the (128, 64) adaptive polar codes over BEC's with different erasure rates (varying from 0.5 to 0.15 in steps of 0.05) by simulations. Codeword error ratio (CER) is measured over 1000 codewords for $\varepsilon =$ 0.5, 0.45, 0.4, 0.35, over 10000 codewords for $\varepsilon = 0.3$, 0.25, and over 100000 codewords for $\varepsilon = 0.2$, 0.15; using single-FG and 4-FG decoders. Also, the average number of iterations for each decoding process to stop is counted, where the preset value of the maximum number of iterations for each single factor graph is 20, but the decoder stops much earlier especially for small erasure rates.

We compare the performances of the following four decoders; *i*) the least SON valued single-FG decoder 1234567, *ii*) the reference single-FG decoder 7654321, *iii*) the 4-FG cyclic subset {7654321, 5432176, 3217654, 1765432}, *iv*) the 4-FG set {7654321, 5236147, 3412765, 1674523} generated by Set Choice Algorithm.

In Figure 3.31, codeword error ratio (CER) performances of the mentioned four decoders are sketched versus channel erasure rate for the (128, 64) adaptive polar codes (having the basis vectors shown in **Table 3.13**). Also, in Figure 3.32, corresponding bit error ratio (BER) values are demonstrated; which are much smaller than CER values as expected (since even a single bit error may result in an undecoded codeword of length 128 bits).

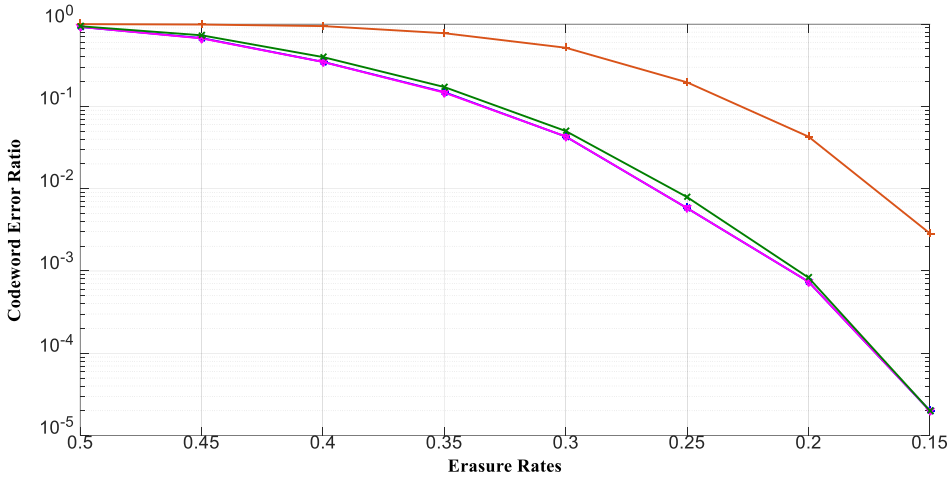


Figure 3.31 Codeword error ratio performance for the (128, 64) adaptive polar code over different BEC(ε) of four decoders, *i*) orange curve: single-FG with SON=1234567, *ii*) green curve: the RFG (7654321), *iii*) pink curve: 4-FG cyclic subset, *iv*) blue curve: 4-FG set produced by the Set Choice algorithm.

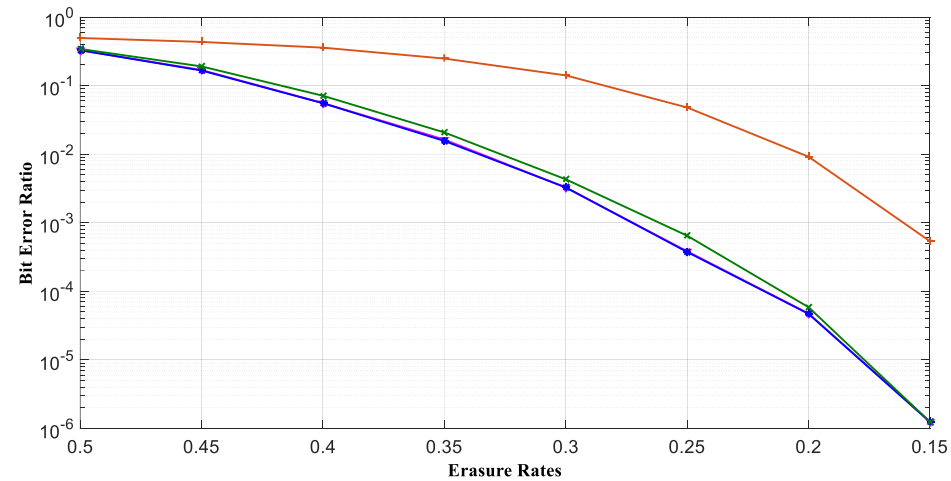


Figure 3.32 Bit error ratio performance for the (128, 64) adaptive polar code over different BEC(ε) of four decoders, *i*) orange curve: single-FG with SON=1234567, *ii*) green curve: the RFG (7654321), *iii*) pink curve: 4-FG cyclic subset, *iv*) blue curve: 4-FG set produced by the Set Choice algorithm.

As can be seen in Figure 3.31 and Figure 3.32, performance of the least SON valued single-FG decoder is the worst; for $\varepsilon = 0.15$ its CER is 143 times worse than the others, and its BER is approximately 430 times poorer. Other three decoders perform quite similarly; although the performances of the 4-FG decoders seem slightly better than the single RFG at high erasure rates, they become very close at high erasure rates. So, at low erasure probabilities, it seems to be redundant to use the remaining three factor graphs of the 4-FG decoder following its first RFG, with the perfect knowledge-based stopping criterion used throughout this work.

On the other hand, Elkelesh, Ebada, Cammerer and Brink have recently shown that as M gets as large as 1000, the performance of M-FG decoders over a BSC (with perfect knowledge-based stopping criterion) can slightly outperform that of the state-of-the-art SCL decoder (without an outer CRC code) [Elkelesh, Ebada, Cammerer, & Brink, 2018-1]. However, the addition of CRC improves the SCL performance by more than 100 times at E_b/N_0 values larger than 2.4 dB, whereas it doesn't appreciably contribute to the M-FG BP decoder performance, even for $M = 1000$ [Elkelesh, Ebada, Cammerer, & Brink, 2018-1]. The BP list decoder that uses $L = 32$ independent FG's mentioned in another paper of the same authors [Elkelesh, Ebada, Cammerer, & Brink, 2018-2] achieves similar performance with a G-matrix based stopping criterion. Again, it cannot reach the performance of the CRC aided SCL.

Finally, in Figure 3.33, the average number of iterations with "perfect knowledge-based stopping criterion" for single and 4-FG decoders, are indicated for different channel erasure rates. Since performances are measured over trials with different number of codewords, we normalize the number of average iterations per 1000 codewords for each decoder. At $\varepsilon = 0.5$, single-FG decoders complete all of the preset 20 iterations with no success, resulting in 20,000 iterations per 1000 codewords, and 4-FG decoders complete almost 4 times more iterations. As expected, average iteration numbers decrease while the channel erasure rate decreases; and those of the RFG and two 4-FG decoders converge approximately to the same value at $\varepsilon = 0.15$, since successful decoding is mostly performed by the first FG of the 4-FG decoders without any need for the remaining 3 FG's.

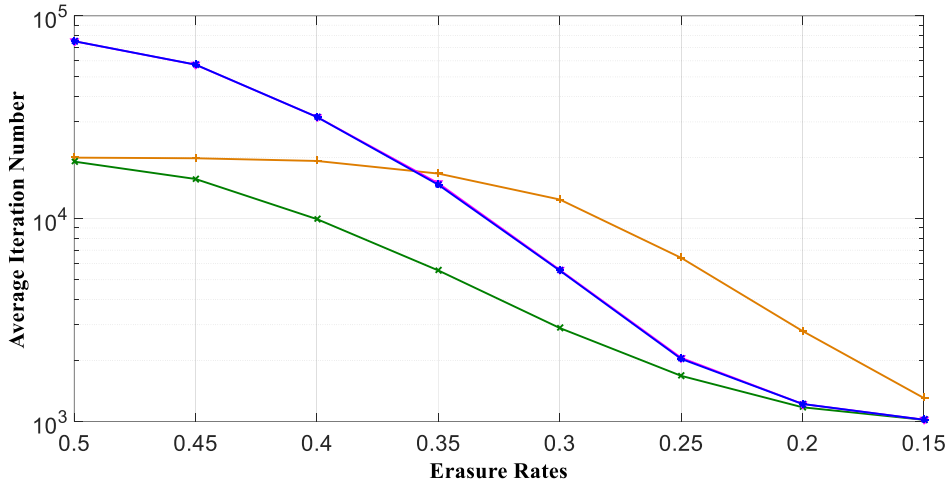


Figure 3.33 Average number of BP decoding iterations over 1000 codewords with perfect knowledge-based stopping condition for the (128, 64) adaptive polar code over different BEC(ε) of four decoders, *i*) orange curve: single-FG with SON=1234567, *ii*) green curve: the RFG (7654321), *iii*) pink curve: 4-FG cyclic subset, *iv*) blue curve: 4-FG set produced by the Set Choice algorithm.

CHAPTER 4

CONCLUSION

In this thesis, some practical results about the single-FG and multiple-FG (M-FG) belief propagation (BP) decoding performance of polar codes are discussed. There are possible $n!$ different factor graphs (FG's), which can be generated by permuting the stages with respect to each other for polar codes with blocklength $N = 2^n$. For the classification of these $n!$ FG's, two parameters are used in [Doğan, 2015] and [Peker, 2018]. One of them is the capacity sum (CS), which is defined as the sum of capacities of the K information channels [Doğan, 2015] and the second one is the number of frozen variables (FV), which depends on the locations of frozen input bits and their connections on the diagram. As a third classification parameter, we suggest the "stage order number-SON", defined as the "n-digit integer obtained by writing the stage names of the FG from left to right", like 123…n, 321…n, or n…231. Since the SON is unique for each FG, there are $n!$ different SON values for $n!$ different FG's. Relations between "FV versus SON" and "CS versus SON" are not linear but have parallel increment-decrement trends.

The best single-FG decoding performance generally belongs to the factor graph with maximum FV, CS [Doğan, 2015], [Peker, 2018], and the SON value (n…321), corresponding to the reference factor graph (RFG). We observe that single-FG decoding performances of the factor graphs with high FV, CS and SON values are usually more successful; and the performance remains approximately the same, as long as the order of the last stages is preserved. For example, single-FG decoding performance of the FG's with SON's 645321, 564321 and 465321 are very similar for the (64, 12) polar code.

We then study M-FG decoding, where $M > 1$ and the transfer of an undecoded word from one FG to the subsequent FG can be done in two ways; either keeping all the

erasure positions of the undecoded word, or letting some of its erasures be filled by the previous FG decoder. We name the first case as “independent” and the second case as “dependent” M-FG decoding. We observe that dependent M-FG BP decoder is more successful than the independent one for different erasure patterns and sets of FG’s. In addition, the time spent for dependent M-FG BP decoding is less than that of the independent one.

Since each FG sequentially tries to decode the remaining undecoded words from the previous one, and it has a different amount of contribution to the total decoding process; the choice of a compatible set of FG’s is significant in M-FG BP decoding. According to the results in Section 3.2.3 and 3.2.4, the factor graphs with distinct FV and CS values are more compatible with each other in team work. Thus, it can be said that using different FG’s with different FV and CS values may lead to better performance for M-FG decoders.

During the search for compatible sets of factor graphs to improve the multiple factor graph BP decoding of polar codes, it seems reasonable to choose the first element of the set as the FG with the best single-FG decoding performance; hence, all our compatible sets start with the reference FG.

We think that cyclic n-FG sets by [Korada, 2009] are compatible, because they achieve better performance than that of the single-RFG; however, at the cost of increased complexity. Our simulations show that similar performance can be obtained by eliminating some FG’s of the cyclic set while keeping the range of CS and FV values wide enough. The size of the M-FG decoder can be reduced from n to $\lceil n/2 \rceil$ without any significant loss in performance.

In addition to ensuring variety in the FV and CS values of FG’s within the compatible FG set, using distinct stage numbers at each stage of factor graphs seems to provide better M-FG decoding performance. To ensure the variance of stage numbers at each stage, and the variety between stage passages as well, we offer the Set Choice algorithm to generate a compatible FG set with $M = \lceil n/2 \rceil$. Since RFG has the best single-FG performance, algorithm starts by picking the RFG as the first element of the set. Other factor graphs are produced by shifting the odd-indexed

stages of the RFG left-cyclically and its even-indexed stages right-cyclically. Set Choice algorithm provides a variety of stages at each stage position. Moreover, it delivers diversity for stage passages between successive state positions, which is missing in the cyclic set [Korada, 2009] that conserves the successive stage numbers. Due to the nature of the BP decoding algorithm, each stage is an individual decoder, so variety at successive stage transitions improves the performance of the M-FG decoder. For example, the compatible set generated by the Set-Choice algorithm for $n = 6$, performs as well as Doğan's genie-chosen set generated by assessing the empirical results [Doğan, 2009].

We also compare the performances of n-FG cyclic BP decoders [Korada, 2009], for $n = 6, \ldots, 11$, with the performances of their $\lceil n/2 \rceil$-FG subsets and $\lceil n/2 \rceil$-FG's generated by the Set Choice algorithm. We observe that although n-FG decoder performance is slightly better than $\lceil n/2 \rceil$-FG decoders, the sets produced by the Set Choice algorithm have consistently the smallest CER at the output of the $\lceil n/2 \rceil$'th FG, for all considered block lengths for rate ½ adaptive polar codes designed for a binary erasure channel with erasure rate 0.35. Therefore, using the Set Choice algorithm to generate a compatible set for a low-complexity dependent M-FG BP decoding appears to be reasonable.

Adaptive polar code construction is dependent on the erasure probability of the given binary erasure channel. Repeating our simulations for different (128, 64) adaptive polar codes over binary erasure channels with different erasure rates, we observe that the CER/BER advantage of the 4-FG decoders over the single reference FG decoder disappears at low erasure rates. The average number of iterations required for 4-FG decoder approaches to that of the reference FG decoder as the channel erasure rate decreases; where there usually remains no need for the FG's that follow the first FG.

Recent work on the subject of permuted factor graphs (FG's) under BP decoding is quite promising. Rather than concentrating on small sets of FG's, a recently announced BP decoder uses as many as 1000 different FG's by randomly permuting the FG stages of the (2048, 1024) polar codes in [Elkelesh, Ebada, Cammerer, & Brink, 2018-1]. Utilizing the perfect knowledge-based stopping criterion, the

proposed multiple trellis decoder slightly outperforms the SCL decoder with list size 32, but it cannot catch the performance of the SCL-CRC decoder with cyclic redundancy check (CRC) aided stopping criterion. The same researchers also investigate the BP list (BPL) decoding performance of L (=32 or 64) permuted FG's, which decode independently in a parallel decoding system [Elkelesh, Ebada, Cammerer, & Brink, 2018-2]. Rather than the random permutations used previously [Elkelesh, Ebada, Cammerer, & Brink, 2018-1], all cyclically shifted versions (i.e., 11 cyclic FG's for $N = 2048$) of the original reference FG (plus 21 or 53 more FG's) are included among these L factor graphs. With a stopping criterion based on the generator matrix G, parallel BP list decoder (i.e., the BPL decoder) reaches almost the same performance as the SCL decoder with a lower latency; but again, it cannot catch the performance of the SCL+CRC-16 that uses the cyclic redundancy check.

It is also interesting that at the CER value of 10^{-4}, the (2048, 1024) hybrid RM-polar code (with minimum distance 32) first suggested in [Li, Shen, & Tse, 2014] outperforms the (2048, 1024) polar code by 1.2 dB (Fig. 2 of [Li, Shen, & Tse, 2014]) or by 0.9 dB (Fig.4 of [Elkelesh, Ebada, Cammerer, & Brink, 2018-2]) with SCL decoding. On the other hand, the BPL performance of the mentioned RM-polar code is 0.45 dB below its SCL performance, whereas that of the polar code is only 0.05 dB below its SCL performance at the same CER (Fig.4 of [Elkelesh, Ebada, Cammerer, & Brink, 2018-2]).

Considering these recent works, which show that the performance of M-FG belief propagation list (BPL) decoding may approach to that of the state-of-the-art SCL decoding (without an outer CRC code); we believe that M-FG belief propagation decoding will gain a significant place in today's world, where the implementation speed matters a lot.

It is the subject of future studies to investigate the effect of stage orders on decoding performance and compose the sets with compatible factor graphs theoretically. Additionally, construction of hybrid RM-polar codes, concatenation of polar codes with other linear codes for better performance, and the design of parallelizable decoding algorithms with the smallest latency are extremely interesting and fruitful research areas.

REFERENCES

Arıkan, E. (2008). A performance comparison of polar codes and Reed-Muller codes. *IEEE Communications Letters 12(6).*

Arıkan, E. (2009). Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Transactions on Information Theory 55(7),* 3051-3073.

Arıkan, E. (2009). On the rate of channel polarization. *Information Theory, 2009, ISIT 2009. IEEE International Symposium on*.

Arıkan, E. (2011). Systematic Polar Coding. *IEEE Communications Letters 15(8),* 860-862.

Berrou, C., & Glavieux, A. (1996). Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications 44(10),* 1261-1271.

Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*.

Bose, R. C., & Ray-Chaudhuri, D. K. (1960). On A Class of Error Correcting Binary Group Codes. *Information and Control,* 68-79.

Cammerer, S., Ebada, M., Elkelesh, A., & Brink, S. t. (2018). Sparse Graphs for Belief Propagation Decoding of Polar Codes. *IEEE International Symposium on Information Theory (ISIT).*

Cammerer, S., Leible, B., Stahl, M., Hoydis, J., & Brink, S. t. (2017). Combining belief propagation and successive cancellation list decoding of polar codes on a GPU platform. *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*.

Chen, K., Niu, K., & Lin, J. (2012). List successive cancellation decoding of polar codes. *Electronics Letters 48(9)*.

Doan, N., Hashemi, S. A., Mondelli, M., & Gross, W. J. (2018). On the Decoding of Polar Codes on Permuted Factor Graphs.

Doğan, O. (2015). An Investigation On Belief Propagation Decoding Of Polar Codes. *MSc. Thesis METU*.

Dumer, I., & Shabunov, K. (2006). Soft-decision decoding of Reed-Muller codes: recursive lists. *IEEE Transactions on Information Theory 52(3)*, 1260-1266.

Elias, P. (1955). Coding for noisy channels. *IRE Conv. Rec. 4*, 37-46.

Elkelesh, A., Ebada, M., Cammerer, S., & Brink, S. t. (2016). Improving Belief Propagation decoding of polar codes using scattered EXIT charts. *2016 IEEE Information Theory Workshop (ITW)*.

Elkelesh, A., Ebada, M., Cammerer, S., & Brink, S. t. (2018). Belief Propagation Decoding of Polar Codes on Permuted Factor Graphs. *IEEE Wireless Communication and Networking Conference (WCNC)*.

Elkelesh, A., Ebada, M., Cammerer, S., & Brink, S. t. (2018). Belief Propagation List Decoding of Polar Codes. *IEEE Communications Letters*.

Eslami, A., & Pishro-Nik, H. (2010). On bit error rate performance of polar codes in finite regime. *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*.

Eslami, A., & Pishro-Nik, H. (2013). On Finite-Length Performance of Polar Codes: Stopping Sets, Error Floor, and Concatenated Design. *IEEE Transactions on Communications 61(3)*, 919-929.

Eslami, A., & Pishro-Nik, H. (2013). On Finite-Length Performance of Polar Codes: Stopping Sets, Error Floor, and Concatenated Design. *IEEE Transactions on Communications 61(3)*, 919-929.

Fayyaz, U. U., & Barry, J. R. (2014). A low-complexity soft-output decoder for polar codes. *Global Communications Conference (GLOBECOM), 2013 IEEE*.

Forney, G. D. (2001). Codes on graphs: Normal realizations. *IEEE Transactions on Information Theory 47(2)*, 520-548.

Gallager, R. G. (1962). Low-Density Parity-Check Codes. *IRE TRANSACTIONS ON INFORMATION THEORY 8(1)*, 21-28.

Gallager, R. G. (1963). Low-Density Parity-Check Codes. *PhD Thesis*.

Golay, M. J. (1949). Notes on digital coding. *Proceedings of the IRE, 37*, 657.

Guo, J., Qin, M., Fabregas, A. G., & Siegel, P. H. (2014). Enhanced belief propagation decoding of polar codes through concatenation. *Information Theory (ISIT), 2014 IEEE International Symposium on*.

Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal 29(2)*, 147-160.

Hashemi, S. A., Doan, N., Mondelli, M., & Gross, W. J. (2018). Decoding Reed-Muller and Polar Codes by Successive Factor Graph Permutations.

Hassani, S. H., Korada, S. B., & Urbanke, R. (2009). The compound capacity of polar codes. *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*.

Hussami, N., Korada, S. B., & Urbanke, R. (2009). Performance of polar codes for channel and source coding. *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*.

Korada, S. B. (2009). Polar Codes for Channel and Source Coding. *PhD. Thesis EPFL*.

Li, B., Shen, H., & Tse, D. (2014). A RM-Polar Codes. [Online]. Available at: https://arxiv.org/abs/1407.5483.

Lin, J., Sha, J., Li, L., Xiong, C., Yan, Z., & Wang, Z. (2016). A high throughput belief propagation decoder architecture for polar codes. *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*.

Lin, J., Xiong, C., & Yan, Z. (2015). Reduced complexity belief propagation decoders for polar codes. *Signal Processing Systems (SiPS), 2015 IEEE Workshop on*.

MacKay, D., & Neal, R. (1996). Near Shannon limit performance of low density parity check codes. *Electronics Letters 32(18)*, 1645.

Mahdavifar, H., & Vardy, A. (2011). Achieving the Secrecy Capacity of Wiretap Channels Using Polar Codes. *IEEE Transactions on Information Theory 57(10)*, 6428-6443.

Muller, D. E. (1954). Application of Boolean algebra to switching circuit design and to error detection. *Transactions of the I.R.E. Professional Group on Electronic Computers, 3(3)*, 6-12.

Pamuk, A. (2011). An FPGA implementation architecture for decoding of polar codes. *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*.

Peker, G. (2018). Belief Propagation Decoding of Polar Codes Under Factor Graph Permutations. *MSc. Thesis METU*.

Reed, I. (1954). A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory, 4(4)*, 38-49.

Reed, I., & Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society of Industrial and Applied Mathematics*, 300-304.

Sha, J., Liu, X., Wang, Z., & Zeng, X. (2015). A memory efficient belief propagation decoder for polar codes. *China Communications 12(5)*, 34-41.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system Technical Journal, 27(3)*, 379-423.

Şimşek, C., & Türk, K. (2016). Simplified Early Stopping Criterion for Belief-Propagation Polar Code Decoders. *IEEE Communications Letters 20(8)*, 1515-1518.

Tal, I., & Vardy, A. (2011). List decoding of polar codes. *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*.

Trifonov, P. (2012). Efficient Design and Decoding of Polar Codes. *IEEE Transactions on Communications 60(11)*, 3221-3227.

Vangala, H., Viterbo, E., & Hong, Y. (2014). Permuted successive cancellation decoder for polar codes. *Information Theory and its Applications (ISITA), 2014 International Symposium on*.

Xu, J., Che, T., & Choi, G. (2015). XJ-BP: Express Journey Belief Propagation Decoding for Polar Codes. *Global Communications Conference (GLOBECOM), 2015 IEEE*.

Yuan, B., & Parhi, K. K. (2013). Architecture optimizations for BP polar decoders. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*.

Yuan, B., & Parhi, K. K. (2014). Algorithm and architecture for hybrid decoding of polar codes. *Signals, Systems and Computers, 2014 48th Asilomar Conference on*.

Zhang, Y., Liu, A., Pan, X., Ye, Z., & Gong, C. (2014). A Modified Belief Propagation Polar Decoder. *IEEE Communications Letters 18(7)*, 1091-1094.

Zhang, Y., Zhang, Q., Pan, X., Ye, Z., & Gong, C. (2014). A simplified belief propagation decoder for polar codes. *Wireless Symposium (IWS), 2014 IEEE International*.

Zhao, S., Shi, P., & Wang, B. (2011). Designs of Bhattacharyya Parameter in the Construction of Polar Codes. *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*.