BELIEF PROPAGATION DECODING OF POLAR CODES UNDER FACTOR GRAPH PERMUTATIONS

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

BY

AHMET GÖKHAN PEKER

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

JANUARY 2018

Approval of the thesis:

BELIEF PROPAGATION DECODING OF POLAR CODES UNDER FACTOR GRAPH PERMUTATIONS

submitted by **AHMET GÖKHAN PEKER** 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. Gülbin Dural Ünver
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. Erdal Arıkan _______________
Electrical and Electronics Engineering Dept., BİLKENT

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

Prof. Dr. Elif Uysal-Bıyıkoğlu
Electrical and Electronics Engineering Dept., METU _______________

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 : AHMET GÖKHAN PEKER

Signature :

ABSTRACT

BELIEF PROPAGATION DECODING OF POLAR CODES UNDER FACTOR GRAPH PERMUTATIONS

Peker, Ahmet Gökhan

MSc., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Melek Diker Yücel

January 2018, 88 pages

Polar codes, introduced by Arıkan, are linear block codes that can achieve the capacity of symmetric binary-input discrete memoryless channels with low encoding and decoding complexity. Polar codes of block length N are constructed by channel polarization method, which consists of channel combining and splitting operations to obtain N polarized subchannels from N copies of binary-input discrete memoryless channels. As N grows, symmetric channel capacities of the polarized subchannels converge to either 0 or 1. Polar codes are also close cousins of Reed-Muller codes and start to differ from each other for $N \geq 32$.

Encoding and decoding of polar or Reed-Muller codes can be performed by using a factor graph, obtained from the n-th Kronecker product $G_N = F^{\otimes n}$ of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ with $N = 2^n$. Such a factor graph contains $n = \log_2 N$ stages; hence, by changing the order of stages with respect to each other, $n!$ different factor graphs can be obtained. In the literature, some decoders using multiple factor graphs instead of a single factor graph are suggested. Therefore, it is of interest whether i) the $K \times N$ generator matrix of the code chosen by K active bits at the input of the encoder, and *ii*) the sum of the capacities of the K active channels that connect each input bit to the output vector of the encoder are invariant under stage permutations. In this study, we give an alternative proof of the fact that the answer to the first question is positive. It is also shown that the sum of the capacities of the K active channels is not invariant under stage permutations.

Belief Propagation decoding performances on single and multiple factor graph decoders of polar and Reed-Muller codes over binary erasure channels are evaluated and compared. For multiple factor graph decoders, practical choice of factor graph sets that gives the best performance with low complexity is examined.

Keywords: Polar Codes, Reed-Muller Codes, Belief Propagation, Factor Graph

ÖZ

KUTUPSAL KODLARIN FAKTÖR GRAFİĞİ PERMÜTASYONLARIYLA İNANÇ YAYILIMI KOD ÇÖZÜMÜ

Peker, Ahmet Gökhan

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Melek Diker Yücel

Ocak 2018, 88 sayfa

Arıkan tarafından tanıtılan kutupsal kodlar ikili-girişli ayrık hafızasız kanalların kapasitesine düşük kodlama ve kod çözme karmaşıklığı ile ulaşabilen doğrusal blok kodlardır. Kod sözcüğü uzunluğu N olan kutupsal kodlar, kanal birleştirme ve bölme operasyonlarından oluşan kanal kutuplaşması metoduyla, ikili-girişli ayrık hafızasız kanalların N kopyasından N polarize alt kanal elde edilerek oluşturulur. N büyüdükçe, polarize alt kanalların simetrik kanal kapasiteleri 0 ya da 1'e yakınsar. Kutupsal kodlar ayrıca Reed-Muller kodların yakın akrabalarıdır ve $N \geq 32$ için birbirinden farklılaşmaya başlarlar.

Kutupsal veya Reed-Muller kodların kodlaması ve çözülmesi, $N = 2^n$ için $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ matrisinin n'inci Kronecker çarpımından elde edilen $G_N = F^{\otimes n}$ matrisinden elde edilen bir faktör grafik gösterimi kullanılarak gerçekleştirilebilir. Böyle bir faktör grafiği $n = log_2 N$ kademe içerir; dolayısıyla kademelerin birbirine göre sırasını değiştirerek $n!$ farklı grafik gösterimi elde edilebilir. Literatürde, tek bir faktör grafiği yerine çoklu faktör grafikleri kullanan bazı kod çözücüler önerilmektedir. Bu sebeple, i) kodlayıcının girişindeki K aktif ikilin seçtiği $K \times N$

boyutlu kod üreteç matrisinin ve ii) kodlayıcının her giriş ikilini çıkış vektörüne bağlayan K aktif kanalın kapasite toplamının, kademe permütasyonları ile değişip değişmediği merak konusudur. Bu çalışmada, ilk sorunun cevabının olumlu olduğu farklı bir yoldan kanıtlanmıştır. Ayrıca, K aktif kanalın kapasite toplamının, kademe permütasyonlarına bağlı olarak değişebileceği gösterilmiştir.

Kutupsal ve Reed-Muller kodların tek ve çok faktör grafikli kod çözücülerinin ikili silinti kanalı üzerindeki İnanç Yayılımı çözücü performansları değerlendirilmiş ve karşılaştırılmıştır. Çok faktör grafikli kod çözücülerinde, düşük karmaşıklıkla en iyi performansı veren faktör grafiği kümelerinin pratik seçimi incelenmiştir.

Anahtar Sözcükler: Kutupsal Kodlar, Reed-Muller Kodlar, İnanç Yayılımı, Faktör Grafikler

To My Family

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Assoc. Prof. Dr. Melek Diker Yücel for the continuous support of my M.Sc. study and for her patient guidance. I would never have been able to finish my thesis without her guidance.

Besides my supervisor, I would like to thank my parents for their love and encouragement.

I would also like to thank to all my colleagues and friends for providing me with continuous encouragement through the process of researching and writing this thesis.

Finally, I would like to also acknowledge my company for providing me support during my graduate study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BCH	Bose, Chaudhuri, and Hocquenghem
B-DMC	Binary-input Discrete Memoryless Channel
BEC	Binary Erasure Channel
BP	Belief Propagation
CS	Capacity Sum
DMC	Discrete Memoryless Channel
FG	Factor Graph
FV	Number of Frozen Variables
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
RFG	Reference Factor Graph
RM	Reed-Muller
SC	Successive Cancellation

CHAPTER 1

INTRODUCTION

Communication systems aim to send data reliably over a noisy channel. The block diagram for a basic communication system is given in Figure 1.1, in which the source encoder eliminates the redundant information from the source. Channel encoder then adds some redundancy to the source-encoded data before transmission to send it reliably over a noisy channel. At the receiver, the channel decoder tries to remove the channel noise and obtain what is sent from the transmitter. Finally, the output of the channel decoder is processed by the source decoder, to recover the original source data.

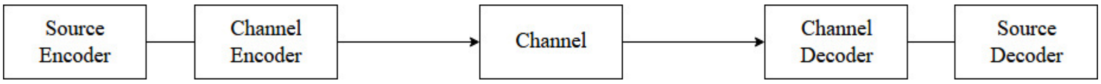


Figure 1.1: Block diagram of a basic communication system

The channel, that is illustrated Figure 1.1, can be described as a medium over which the information is transmitted. It is defined mathematically by a transition probability function $W(y|x)$ for $x \in X, y \in Y$; where X denotes the input alphabet, and Y denotes the output alphabet.

Throughout the thesis, we will consider binary erasure channels (BEC), which are in the class of binary-input discrete memoryless channels.

1.1. Channel Coding

Shannon, in his seminal work, discussed how to achieve reliable communication over a noisy channel [Shannon, 1948]. He obtained the capacity of a channel and related it to the maximum rate for sending data reliably. If the transmission information rate is less than the channel capacity, then it is possible to transmit data in a noisy environment reliably with arbitrarily small error probability.

Theorem 1.1 (Channel Coding Theorem) [Shannon, 1948]:

Let W be a discrete memoryless channel. The capacity, C, of a discrete memoryless channel is given by

$$C = I(W) = I(X;Y)$$

An (M,N) code that is transmitted over the channel $W: X \rightarrow Y$ consists of following:

1. An index set $\{1,2,\dots,M\}$ for representing messages.
2. An encoding function $f:\{1,2,\dots,M\} \rightarrow X^N$ for generating codeword.
3. A decoding function $g: Y^N \rightarrow \{1,2,\dots,M\}$

The rate of an (M,N) code is calculated as $R = \frac{log_2 M}{N}$ bits per transmission.

There is a nonnegative channel capacity C associated with each discrete memoryless channel, such that for all the rates R less than the capacity, information data can be transmitted over the channel and be reconstructed with an arbitrarily small probability of error. All the rates below the capacity C are achievable. ■

Since Shannon's work, main aim is to search capacity-achieving codes with low encoding and decoding complexity. A detailed review of channel coding theory and the coding schemes till 2007 are given in [Costello, 2007].

The initial research in channel coding is concentrated on obtaining linear codes with good algebraic properties, such as large minimum Hamming distance. Hamming codes [Hamming, 1950], Golay codes [Golay, 1949], Reed-Muller codes ([Reed, 1954], [Muller, 1954]), BCH codes [Bose and Ray-Chaudhuri, 1960] and Reed-Solomon codes [Reed and Solomon, 1960] can be shown as examples for algebraic codes.

Another research path in channel coding is focused on Shannon's random coding approach that is also called probabilistic coding. Probabilistic coding is concentrated on finding classes of codes that optimize average performance as a function of encoding and decoding complexity. Convolutional codes, product codes, concatenated codes, trellis-coded modulation, and trellis decoding of block codes are examples for this this type of coding schemes.

A big step towards the channel capacity has been achieved by the introduction of Turbo codes [Berrou, 1993]. They are a class of codes that can perform very close to the Shannon limit with linear decoding complexity. Shortly thereafter, Gallagher's LDPC codes are rediscovered along with low-complexity iterative decoding [MacKay and Neal, 1996] that can attain near-Shannon-limit performance.

The source and channel encoder blocks have to work on data with large block lengths to get small error probabilities. Having large block lengths affect complexity in encoding and decoding. For practical applications, communication system is required to have low computational complexity. We focus our attention on polar codes, which is one of the latest advances in coding theory.

1.2. Polar Codes

Polar codes introduced by Arıkan are linear block codes with low encoding and decoding complexity that can achieve the capacity of symmetric binary-input discrete memoryless channels (B-DMC's) [Arıkan, 2009].

Polar codes can be considered as a generalization of the Reed-Muller (RM) codes ([Arıkan, 2008], [Hussami, 2009]). The $K \times N$ generator matrix G_P of the (N, K) polar code where $N = 2^n$ is constructed choosing K rows of the matrix $G_N = F^{\otimes n}$, which is the n-th Kronecker product of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. The generator matrix of RM codes is also formed by selecting rows of G_N. The main difference between polar and RM codes is that, while the RM codes choose all rows of G_N with Hamming weights over a specific value, the choice for polar codes depends on channel capacities.

Polar codes are constructed by the channel polarization method that consists of two phases, channel combining and channel splitting. The idea in channel polarization is to obtain N polarized subchannels from N copies of binary-input discrete memoryless channels. As N grows, the symmetric channel capacities of distinct channel converges to either 0 or 1 after channel combining and splitting operations. The basic idea of polar coding is to transmit data only through channels, whose channel capacities approaches to 1. The inputs of the remaining channels are frozen; i.e., not used for sending information. Encoding and decoding of the polar codes with blocklength $N = 2^n$, can be performed using special factor graphs (FG's) with n stages that correspond to G_N transformation.

Two common decoding algorithms for polar codes are the successive cancellation (SC) and belief propagation (BP) algorithms. The SC decoder has serial architecture, and hence, it has disadvantages such as the high latency and limited throughput.

Belief Propagation (BP) is a message-passing algorithm that can be run on the mentioned factor graphs having n stages. The BP decoding algorithm for the RM codes proposed by Forney [Forney, 2001] is also applicable to the polar codes.

The message passing schedules and error performances under finite lengths are examined in [Eslami and Pishro-Nik, 2013]. BP decoding with min-sum (MS) approximation is reviewed in [Yuan and Parhi, 2013] and a performance-improved scaled min-sum (SMS) polar decoding algorithm is presented. The XJ-BP decoding algorithm in [Xu et al., 2015] simplifies min-sum BP decoding and it produces the same decoding performance of the scaled min-sum algorithm with reduced complexity. Convential BP decoding algorithm requires large number of iterations and it causes high computational complexity. In [Abbas et al., 2015] and [Lin et al., 2015], low-complexity BP decoding algorithms for polar codes are proposed to decrease the average number of computations. The soft-cancellation (SCAN) decoding method in [Fayyaz and Barry, 2014] is another decoding method to reduce the computational complexity of BP decoders. However, SCAN decoder is serial in nature as opposed to conventional BP decoder, which has parallel architecture, and it causes much longer decoding latency. In [Abbas et al., 2017], an early stopping scheme is proposed for BP decoders. This method decreases the required number of iterations, it also reduces the energy consumption and decoding latency. Moreover, in [Sha et al., 2016], a modified BP polar decoder that performs as well as the conventional BP decoder is proposed by combining adjacent processing stages in the factor graph into a new stage.

Arıkan compares the performance of polar codes with Reed-Muller codes under BP decoding at short block lengths [Arıkan, 2008]. Korada discusses why the performance of BP decoders is better than that of SC decoders; and considers BP decoding with multiple factor graphs (multi-FG's), where different stage permutations of an FG are employed. He observes that performance of the multi-factor graph BP decoder is improved significantly by using $\log_2 N$ factor graphs corresponding to all possible cyclic stage permutations [Korada, 2009].

In this thesis, Belief Propagation algorithm is used for decoding polar and RM codes on single and multiple FG's generated by $G_N = F^{\otimes n}$, where $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. The use of multiple FG's provokes the stimulating question of whether two parameters remain invariant under stage permutations:

i) the $K \times N$ generator matrix chosen by K active bits at the input of the encoder, and

ii) the sum of capacities of the K active channels that connect each input bit to the output vector of the encoder.

One may notice that a positive answer to the first question warrants that all FG encoders implement the same linear code; whereas an affirmative answer to the second question supports the polar code concept of activating the maximum-capacity channels, while freezing the low-capacity ones.

1.3 Aim and Organization of the Thesis

Remembering that factor graph (FG) representations of polar and RM codes with blocklength $N = 2^n$, consist of n stages; and multi-FG decoders use some of the $n!$ factor graphs that are formed by permuting the stage order; goals of this study can be summarized as:

- To discuss whether *i*) the $K \times N$ generator matrix of the (N, K) polar code and ii) the sum of the capacities of the K active channels are invariant under stage permutations.

- To examine the choice of suitable values for the number M of the FG's in multi-FG Belief Propagation (BP) decoding over binary erasure channels and to discuss the choice of the best set of FG's for multi-FG decoding.

- To evaluate and compare BP decoding performances on single-FG and multi-FG decoders of the RM and polar codes over binary erasure channels.

The thesis is organized as follows:

In Chapter 2, construction of the polar codes by channel polarization method is reviewed. Encoding of polar codes is summarized and the factor graph representation, which is used both in their encoding and decoding, is described. Belief Propagation (BP) decoding algorithm, that is performed based on the factor graph representation, is explained. The chapter ends with a discussion about the relation between the two important parameters for the construction of polar codes over binary erasure channels; namely, Bhattacharyya parameters and the symmetric channel capacities.

The $K \times N$ generator matrix G_P of the (N, K) polar is constructed by selecting K basis vectors from the $N \times N$ transformation matrix $G_N = F^{\otimes n}$ according to information bit indices. In Chapter 3, we discuss whether the generator matrix G_P of an (N, K) code alters, when the stage order of the factor graph is changed while keeping the positions of the information bits the same as those of the reference factor graph, RFG. We give an alternative proof the fact that G_N is invariant under stage permutations and so is G_P, provided that the information bit positions are kept the same as those of the RFG.

In Chapter 4, performances of polar and Reed-Muller codes over a BEC with a given erasure rate (ϵ) are evaluated, using the Belief Propagation (BP) algorithm on factor graphs. Two parameters are used to classify the factor graphs: capacity sum of the K transmission paths and the total number of frozen variables on the factor graph implementation. Effects of these two parameters on the BP decoder are investigated. Single-FG decoders that use a single factor graph and multiple-FG decoders that employ a pre-chosen set of $M > 1$ factor graphs are engaged to evaluate the BP decoding performances of polar and Reed-Muller codes. Practical choice of M and factor graph sets for multi-FG decoders that gives the best decoding performances with low complexity is discussed.

Finally, Chapter 5 concludes the study by summarizing the main contributions mentioned in each chapter.

CHAPTER 2

POLAR CODES

In this chapter, construction of polar codes is summarized with reference to Arıkan [Arıkan, 2009] and Korada [Korada, 2009].

In Section 2.1, construction of polar codes by using channel polarization method is explained briefly. Channel combining and splitting operations to obtain N polarized subchannels from N copies of binary-input discrete memoryless channels are described. In Section 2.2, we display how to calculate the capacities of N subchannels after channel combining and splitting operations. In Section 2.3, the encoding of polar codes is summarized, and the factor graph representation, which is employed both in their encoding and decoding, is described. Belief propagation (BP) decoding algorithm that is used in this study is explained in Section 2.4. Finally, Section 2.5 discusses the relation between two important parameters for the construction of polar codes over binary erasure channels (BEC); namely, the symmetric capacity and the Bhattacharya parameters.

We use $W: X \rightarrow Y$ to represent a binary-input discrete memoryless channel (B-DMC) with input alphabet X, output alphabet Y and transition probabilities $W(y|x)$ for $x \in X,\ y \in Y$. Input alphabet X will always be $\{0,1\}$. W_N will denote the channel corresponding to N uses of W, that can be shown as $W_N: X^N \rightarrow Y^N$ with transition probability $W_N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i)$. Upper case letters such as X, Y will be used for showing random variables; and lower case letters such as x, y, to represent their realizations (sample values).

In the following, u_1^N will denote a row vector $(u_1, u_2, \dots, u_N)$ and u_i^j will denote the subvector $(u_i, \dots, u_j)$ for $1 \leq i, j \leq N$. For a given u_1^N and a set of indices $U \subset$

$\{1, \ldots, N\}$, u_U will be used to denote the subvector $(u_i : i \in U)$. All vectors, matrices and operations will be over $GF(2)$.

The Kronecker product of an m-by-n matrix of $A = [A_{ij}]$ and an r-by-s matrix of $B = [B_{ij}]$ is described as

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

which is an mr-by-ns matrix.

For a given B-DMC W, the symmetric capacity (the mutual information between the input and output of W) is given as

$$I(W) = I(X;Y) \triangleq \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \ . \qquad (2.1)$$

2.1 Channel Polarization

Polar codes are constructed by using channel polarization method. Channel polarization is an operation by which one transforms N independent copies of a given binary-input discrete memoryless channel (B-DMC) W into a second set of N channels $\{W_N^{(i)} : 1 \leq i \leq N\}$ that show a polarization effect in the sense that, as N becomes large, the symmetric capacity $\{I(W_N^{(i)})\}$ converges to 0 or 1 for all but a vanishing fraction of indices i. This operation is called as channel polarization and it consists of two phases which are channel combining and channel splitting phases.

Basic Channel Transformation

Polar codes are constructed by recursive steps. In the first step, two independent copies of W are combined and a new channel $W_2 : X^2 \to Y^2$ is obtained by applying the transform G_2. Let X_1^2 be the input to two independent copies of W and Y_1^2 be the output, then $X_1^2 = U_1^2 G_2$ and $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ is the n-th Kronecker product $G_N = F^{\otimes n}$ of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ with $N = 2^n$, and $n = 1$. This phase is defined as channel combining. Channel combining of two channels is plotted in Figure 2.1.

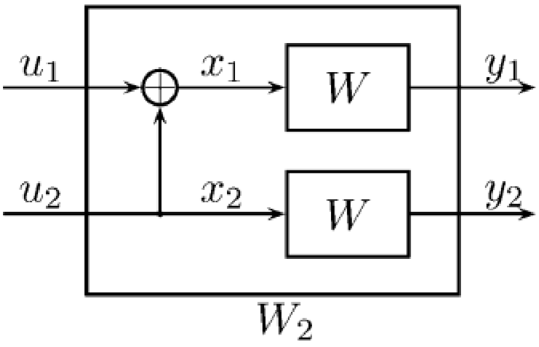


Figure 2.1: Basic channel transform with N = 2.

Obtained new channel W_2 is defined by transition probabilities:

$$W_2(y_1^2|x_1^2) = \prod_{i=1}^{2} W(y_i|x_i) = W(y_1|u_1 \oplus u_2)W(y_2|x_2)$$

By applying the chain rule, the mutual information between the input and the output of the W_2 channel can be written as

$$I(U_1^2; Y_1^2) = I(U_1; Y_1^2) + I(U_2; Y_1^2|U_1)$$

$$I(U_1^2; Y_1^2) = I(U_1; Y_1^2) + I(U_2; Y_1^2, U_1)$$

$I(U_1; Y_1^2)$ and $I(U_2; Y_1^2, U_1)$ terms can be interpreted as the mutual informations of the new subchannels $W_2^{(1)}: X \to Y^2$ and $W_2^{(2)}: X \to X \times Y^2$

The new subchannels $W_2^{(1)}$ and $W_2^{(2)}$are illustrated in Figure 2.2 and 2.3. $W_2^{(1)}$subchannel has U_1 and Y_1^2 as an input and output, with U_2 considered as noise. Also, $W_2^{(2)}$subchannel has U_2 as an input and Y_1^2 as an output, with U_1 being available at the decoder. This phase of splitting W_2 into the subchannels $W_2^{(1)}$and $W_2^{(2)}$ is defined as channel splitting.

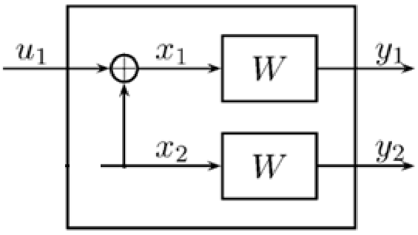


Figure 2.2: $W_2^{(1)}$subchannel after channel splitting with *N* =2.

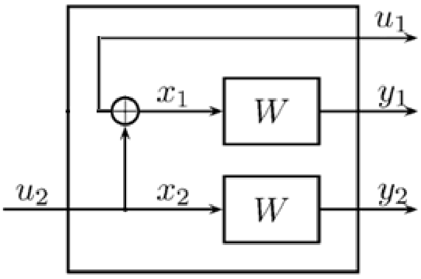


Figure 2.3: $W_2^{(2)}$subchannel after channel splitting with *N* =2.

Capacity of the new W_2 channel can be computed as

$$I(W_2) = I(U_1^2; Y_1^2) = \sum_{i=1}^{2} I(X_i, Y_i) = \sum_{i=1}^{2} I(W) = 2I(W)$$

The new subchannels $W_2^{(1)}$ and $W_2^{(2)}$ satisfy the following transmission of the rate property:

$$I\left(W_2^{(1)}\right) \leq I(W) \leq I\left(W_2^{(2)}\right)$$

Recursive Channel Transformation

In the previous part, channel combining, and splitting operations are applied for block length $N = 2$; $W_2^{(1)}$and $W_2^{(2)}$subchannels are constructed. In this part, for N equals to any power of two, $N = 2^n$, construction of a series of N subchannels will be shown by using channel combining and splitting operations recursively.

In the second step of recursive construction, for $N = 4$, with using two identical copies of W_2, $W_4 : X^4 \to Y^4$ is generated with transition probabilities

$$W_4(y_1^4 | x_1^4) = W_2(y_1^2 | u_1 \oplus u_3, u_2 \oplus u_4) W_2(y_3^4 | u_3, u_4)$$

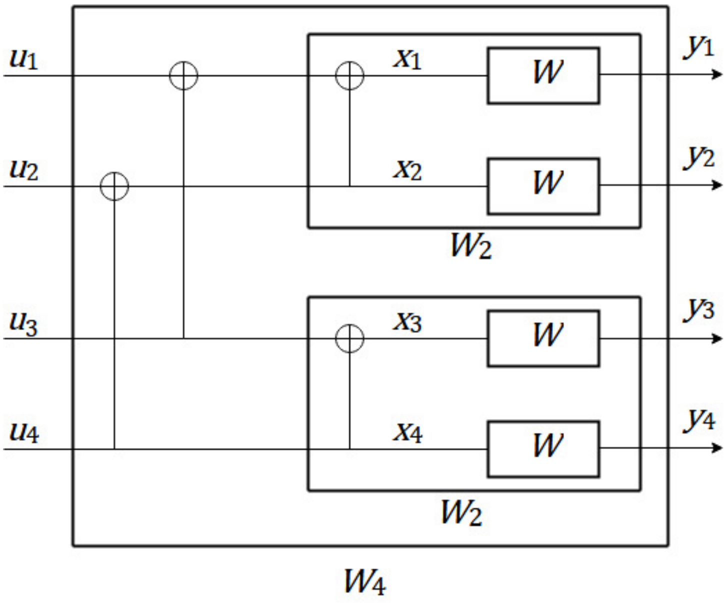


Figure 2.4: W_4 channel which is generated by combining two copies of W_2.

Obtaining W_4 channel by channel combining is illustrated in Figure 4. The mapping $u_1^4 \to x_1^4$ for W_4 channel can be written as $x_1^4 = u_1^4 G_4$ where the transition matrix $G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ is obtained as the n-th Kronecker product $G_N = F^{\otimes n}$ of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ with $N = 2^n$, and $n = 2$.

For $N = 2^n$, the channel combining is done in a recursive manner and $W_N: X^N \to Y^N$ is generated by combining two identical copies of $W_{N/2}$.

Transition probabilities for the channel $W_N: X^N \to Y^N$ are defined as

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

where $u_1^{N/2} = \{u_1, u_2, \dots, u_{N/2}\}$ and $u_{(N/2)+1}^N = \{u_{(N/2)+1}, u_{(N/2)+2}, \dots, u_N\}$.

The overall mapping $u_1^N \to x_1^N$, from the input of the synthesized channel W_N to the input of binary-input discrete memoryless channels W, is also linear and can be represented by $x_1^N = u_1^N G_N$ using the n-th Kronecker product $G_N = F^{\otimes n}$ of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ with $N = 2^n$. The synthesized channel W_N is plotted in Figure 2.5.

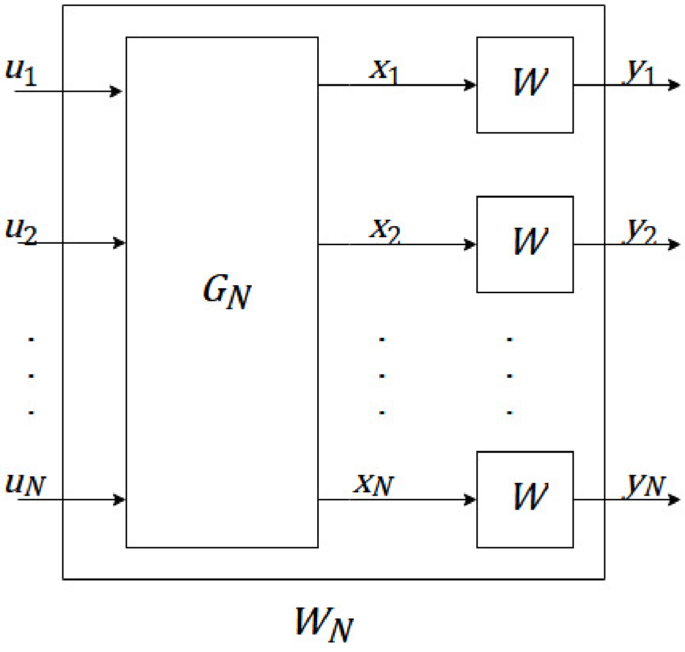


Figure 2.5: The synthesized channel W_N after channel combining operations.

After obtaining the synthesized channel W_N, the next step is splitting W_N into a set of N subchannels $W_N^{(i)}: X \to Y^N \times X^{i-1}, 1 \le i \le N$, defined by transition probabilities

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

where (y_1^N, u_1^{i-1}) represents the output of $W_N^{(i)}$and (u_i) denotes the input of $W_N^{(i)}$. Capacity of the synthesized channel W_N can be written as

$$C(W_N) = I(U^N; Y^N) = I(X^N; Y^N) = NC(W).$$

With applying the chain rule of mutual information, capacity of W_N is split as

$$C(W_N) = I(U^N; Y^N) = \sum_{i=1}^N I(U_i; Y^N, U^{i-1}),$$

where subchannels after channel splitting operation are defined as $W_i: U_i \rightarrow (Y^N, U^{i-1})$ for $1 \leq i \leq N$ as illustrated in Figure 2.6.

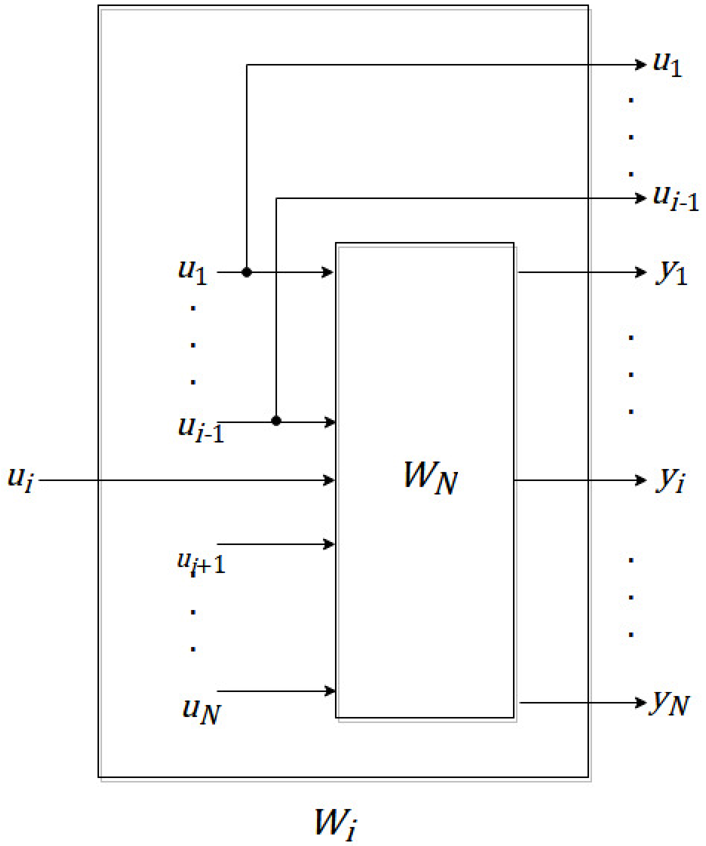


Figure 2.6: The subchannel W_i after splitting of the synthesized channel W_N.

So, N copies of binary-input discrete memoryless channels W transform into a set of N distinct channels $\{W_N^{(i)}: 1 \leq i \leq N\}$ with channel combining and splitting operations. As N goes to infinity, the channel polarization theorem is formalized by Arıkan as follows [Arıkan, 2009].

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

$$\lim_{N \to \infty} \frac{\text{Number of indices } i \in \{1, \dots, N\} \text{ with } I(W_N^{(i)}) \in (1-\delta, 1]}{N} = I(W),$$

$$\lim_{N \to \infty} \frac{\text{Number of indices } i \in \{1, \dots, N\} \text{ with } I(W_N^{(i)}) \in (0, \delta]}{N} = 1 - I(W)\,.$$

This theorem shows that as N goes to infinity, the symmetric capacity of each individual channel converges almost surely to 0 or 1.

2.2. Calculating the Capacities of *N* Subchannels $W_N^{(i)}$ over a BEC

In this part, capacities of the N subchannels $\{W_N^{(i)}: 1 \leq i \leq N\}$ obtained after channel combining and splitting operations will be computed for a binary erasure channel W with erasure probability ϵ, which is abbreviated by BEC(ϵ).

For $N = 2$, subchannels $W_2^{(1)}$ and $W_2^{(2)}$ plotted in Figure 2.2 and 2.3 will be denoted by $W_2^{(1)} = W^-$ and $W_2^{(2)} = W^+$. Then, for $N = 4$, descendants of W^- will be called W^{--}, W^{-+} and those of W^+ will be named as W^{+-}, W^{++}. For N=4, synthesized channel W_4 after channel combining is plotted in Figure 2.7, and subchannels after splitting, in Figure 2.8.

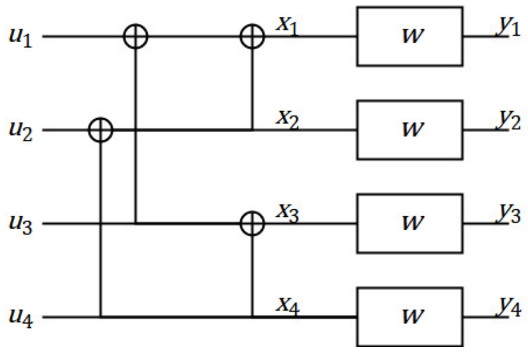


Figure 2.7: Synthesized channel W_4 by combining two copies of W_2.

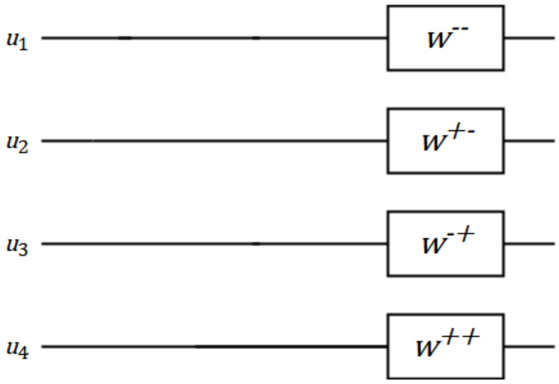


Figure 2.8: Subchannels after channel splitting for $N = 4$.

For a binary erasure channel BEC(ϵ), channel capacity equals to $I(W) = 1 - \epsilon$, where ϵ is the erasure probability. For $N = 2$, obtained channels after channel splitting, W^- and W^+ are also BEC's with erasure probabilities ϵ^- and ϵ^+. For W^- channel, from Figure 2.2, one can observe that $u_1 = x_1 + x_2$. So, u_1 is not an erasure, when both x_1 and x_2 are not erasures. For W^+ channel, that is shown in Figure 2.3, $u_2 = x_2$. If x_2 is an erasure, then u_2 will also be an erasure.

Erasure probabilities of W^- and W^+ channels are calculated as $\epsilon^- \triangleq 1 - (1-\epsilon)^2 = 2\epsilon - \epsilon^2$ and $\epsilon^+ \triangleq \epsilon^2$. Capacities of W^- and W^+ are equal to $1 - \epsilon^-$and $1 - \epsilon^+$ respectively.

For $N = 4; W^{--}, W^{-+}, W^{+-}, W^{++}$ subchannels are generated after channel combining and splitting operations. These channels are also BEC's and their erasure probabilities are calculated by using erasure probabilities ϵ^{-} and ϵ^{+} of W^{-} and W^{+}

$$\epsilon^{--} = 2(\epsilon^{-}) - (\epsilon^{-})^2$$

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

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

$$\epsilon^{++} = (\epsilon^{+})^2$$

In general, channel capacities $\{I({W_N}^{(i)})\}$ of the subchannels ${W_N}^{(i)}$, for $1 \leq i \leq N$, are calculated using the recursive relations,

$$I({W_N}^{(2i)}) = 2I({W_{N/2}}^{(i)}) - I({W_{N/2}}^{(i)})^2$$

$$I({W_N}^{(2i-1)}) = I({W_{N/2}}^{(i)})^2,$$

where $I({W_1}^{(1)}) = I(W) = 1 - \epsilon.$

2.3. Polar Code Encoding

In the previous parts, we have obtained series of channels that are polarized, where the symmetric capacities $I(W_N^{(i)})$ of each individual channel $\{W_N^{(i)}\}$ converges to 0 or 1. By using the polarization effect, polar codes are constructed for a block length $N = 2^n$, where $n \geq 0$. The basic idea of polar coding is to send data only through channels $\{W_N^{(i)}: 1 \leq i \leq N\}$, whose symmetric channel capacity $I(W_N^{(i)})$ approaches to 1. The inputs of the remaining channels are frozen; i.e., not used for sending information. Hence, (N, K) polar codes are produced by freezing $N - K$ lowest capacity paths of the structure defined by $y_1^N = u_1^N G_N$, where $G_N = F^{\otimes n}$ is the n-th Kronecker product of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Factor Graph (FG) Representation of Polar Codes

Encoding and decoding of a polar code with a blocklength $N = 2^n$, can be performed using a graph that corresponds to $y_1^N = u_1^N G_N$ transformation. As an example, for $N = 8$, the synthesized channel W_8, that is constructed recursively is illustrated in Figure 2.9.

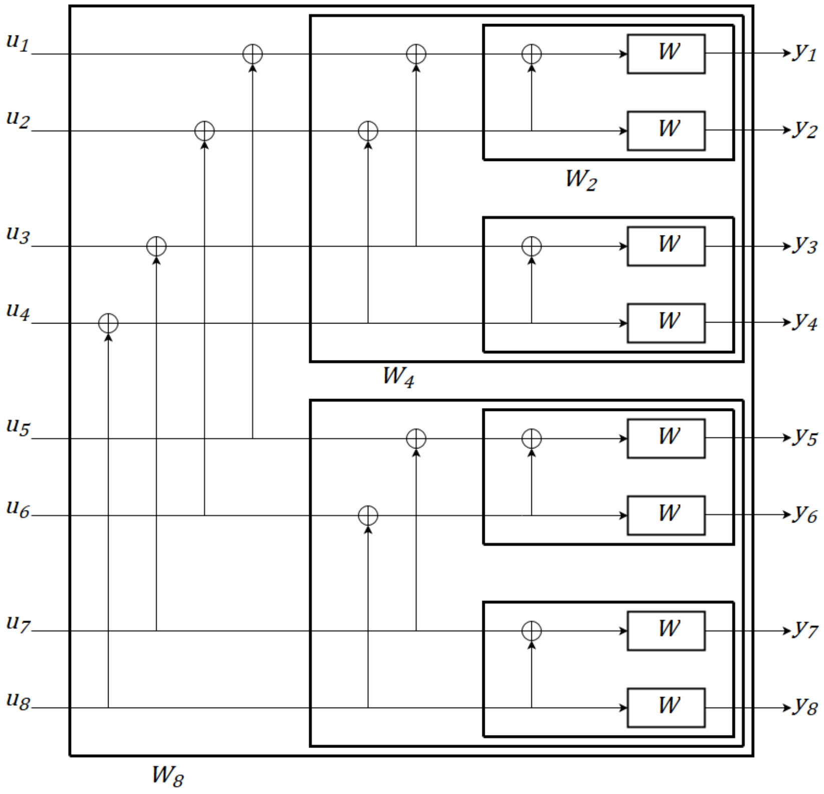


Figure 2.9: The synthesized channel W_8 that is constructed by $y_1^8 = u_1^8 G_8$ transformation.

The synthesized channel W_8, that is illustrated in Figure 2.9, is generated by combining two copies of W_4; and W_4 is produced by combining two copies of W_2. So, the construction of W_8 can be divided into three stages and can be implemented by the factor graph representation given Figure 2.9, which is redrawn in Figure 2.10 using 3 columns of Z connections, where each column corresponds to a stage. In Figure 2.10, each Z connection connects two inputs to two outputs. In general, a polar code with a blocklength $N = 2^n$ has $n = log_2^N$ stages. We name the stages as 1, 2, …, n [Doğan 2015], where Stage 1 contains the smallest Z's with adjacent inputs and Stage n contains the largest Z's that connects input nodes separated by $2^{n-1} = N/2$. Hence, we call the synthesized channel W_8 obtained by $y_1^8 = u_1^8 G_8$ transformation as implemented in Figure 2.10, the "3-2-1" factor graph.

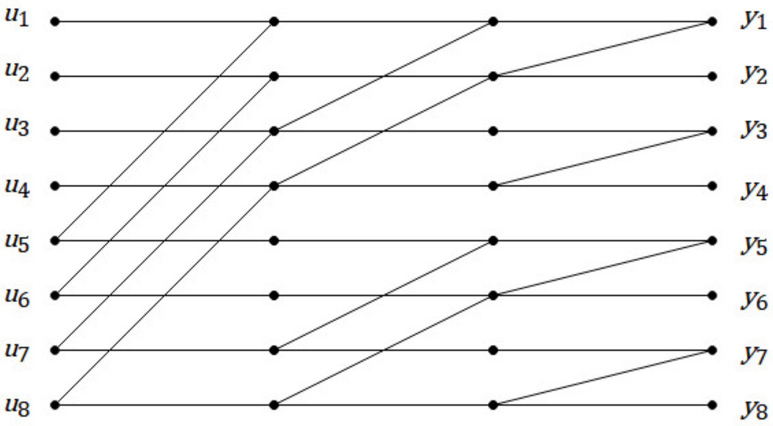


Figure 2.10. Graph representation of the transformation $y_1^8 = u_1^8 G_8$.

For a polar code with blocklength $N = 2^n$, one can form $n!$ different graph representations by changing the order of stages with respect to each other. For example, a polar code with block length $N = 8$ has 6 different factor graph (FG) implementations as illustrated in Figure 2.11; where the FG's in the first line are named as "1-2-3", "1-3-2", "2-1-3" and those in the second line correspond similarly to "2-3-1", "3-1-2", "3-2-1" factor graphs.

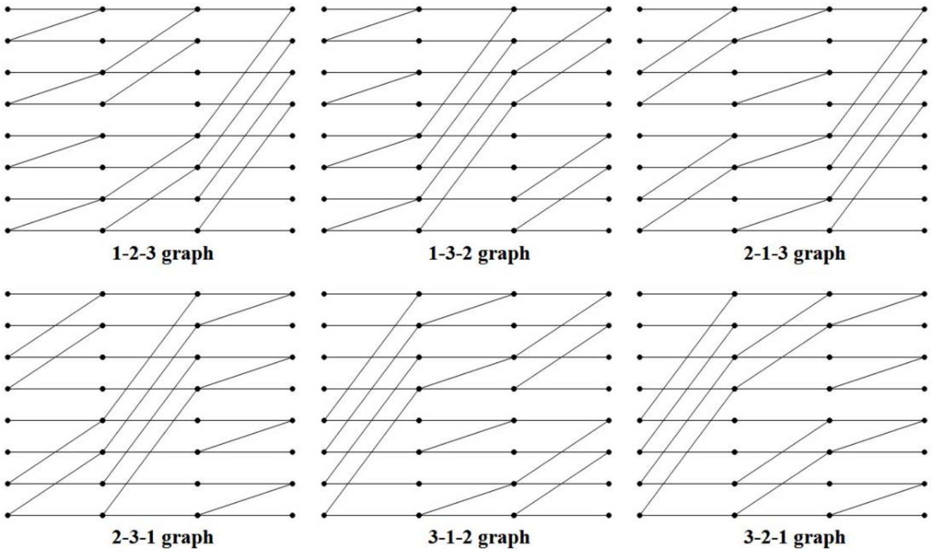


Figure 2.11: 6 different factor graph (FG) representations for a polar code with blocklength $N = 8$.

Selection of Channels for Sending Information and Constructing the Generator Matrix

After obtaining N subchannels $\{W_N^{(i)}: 1 \leq i \leq N\}$, (N, K) polar codes are generated by choosing K channels with the highest capacities. Let $\mathcal{A}$ refer to the information set, which is a subset of $\{1, \dots, N\}$. K is the code dimension and it specifies the size of $\mathcal{A}$. Information vector, denoted as i_1^K, is sent from inputs $u_{\mathcal{A}}$ and the remaining $N - K$ channels are frozen. $\mathcal{A}^c$ refers to the frozen set, which is an $(N - K)$-element subset of $\{1, \dots, N\}$. In the design of polar codes, the frozen vector $u_{\mathcal{A}^c}$ is selected as the 0 vector.

Arıkan uses the Bhattacharyya parameters in the construction of polar codes [Arıkan, 2008]. The Bhattacharyya parameter Z(W) defined as

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

takes values between 0 and 1. For a given BEC(ϵ), if $Z(W) \leq \epsilon$, then this channel is considered as almost noiseless. In the opposite case, for $Z(W) \geq 1-\epsilon$, the corresponding channel is considered as pure-noisy.

For a blocklength $N = 2^n$, after the computation of the transformation matrix $G_N = F^{\otimes n}$, where $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, the rows of G_N are labelled from top to bottom as $i = 1,2,\dots,N$. Then, the Bhattacharyya parameters of all bit channels are calculated in the form of $Z_N = (Z_{N,i}, 1 \leq i \leq N)$ by using the recursive formula

$$Z_{2k,i} = \begin{cases} 2Z_{k,i} - Z_{k,i}^2\,, & 1 \leq i \leq k \\ Z_{k,i-k}^2, & k+1 \leq i \leq 2k \end{cases} \text{ for} = 1,2,2^2,\dots,2^{n-1}\,. \tag{2.3}$$

The initial condition $Z_{1,1}$ is equal to ϵ for the BEC(ϵ). Bhattacharya parameters of all bit-channels are computed by (2.3). Then, the $K \times N$ generator matrix G_P is constructed by choosing K rows of the matrix G_N, whose indices correspond to the bit-channels with K smallest values of the Bhattacharya parameters.

For the BEC(ϵ), the highest-capacity channels of the "n-…2-1" FG also have the least possible Bhattacharya parameters [Arıkan 2009]; but the same claim cannot be made for other FG's with different stage orders. The relation between the Bhattacharyya parameters $Z(W)$ and the symmetric capacities $I(W)$ of different FG's will be demonstrated for $N = 8$ in Section 2.5.

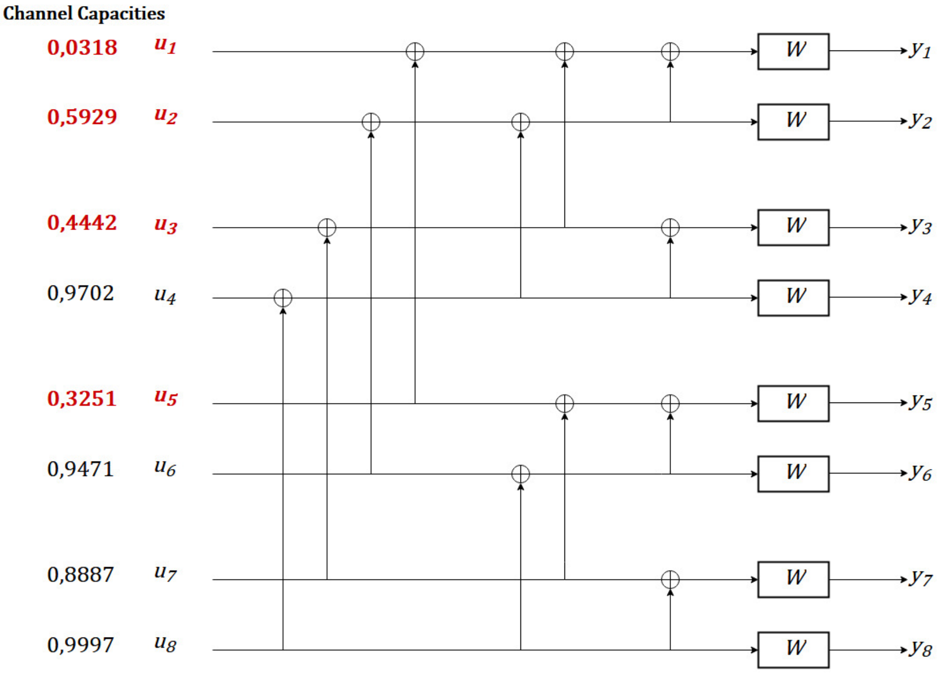


Figure 2.12. Channel capacities of the (8, 4) polar code.

To illustrate the selection of the information and frozen bits by an example, "3-2-1" FG of a (8, 4) polar code and corresponding capacities over BEC(0.35) is given in Figure 2.12. Channel capacities of 8 subchannels are calculated and $K = 4$ channels with the highest capacities are selected to send information. According to the computed channel capacities, the information bit indices are $\mathcal{A} = \{4,6,7,8\}$ and inputs $i_1^4 = (u_4, u_6, u_7, u_8)$ are used to send information. Channels that have the four lowest capacity values are colored red. The frozen bits indices are $\mathcal{A}^c = \{1,2,3,5\}$ and inputs (u_1, u_2, u_3, u_5) are frozen, by fixing their values to 0.

2.4. Belief Propagation Decoding Algorithm

Polar codes are commonly decoded by the successive cancellation (SC) or Belief Propagation (BP) decoding algorithms. The SC algorithm is more popular because of its low complexity compared to the BP algorithm, but it also has disadvantages such as high latency and limited throughput due to its serial decoding nature. In our simulations with polar and Reed-Muller (RM) codes, we have used the BP decoding algorithm (on FG's mentioned in Section 2.3) as summarized in the following.

Belief Propagation (BP) is a message-passing algorithm, which is used for decoding graph-based codes. For a code with block length $N = 2^n$, the FG contains n stages and each stage has N input and N output nodes. Output of the i^{th} stage equals to the input of the $(i+1)^{\text{th}}$ stage. There are nN check nodes and the number of total variable nodes is $(n+1)N$. As an example, all possible factor graphs for $N = 2^3$ have been drawn in Fig 2.11; consisting of $n = 3$ stages. We redraw the 3-2-1 factor graph by denoting the variable nodes and check nodes with circles and rectangles respectively in Fig 2.13.

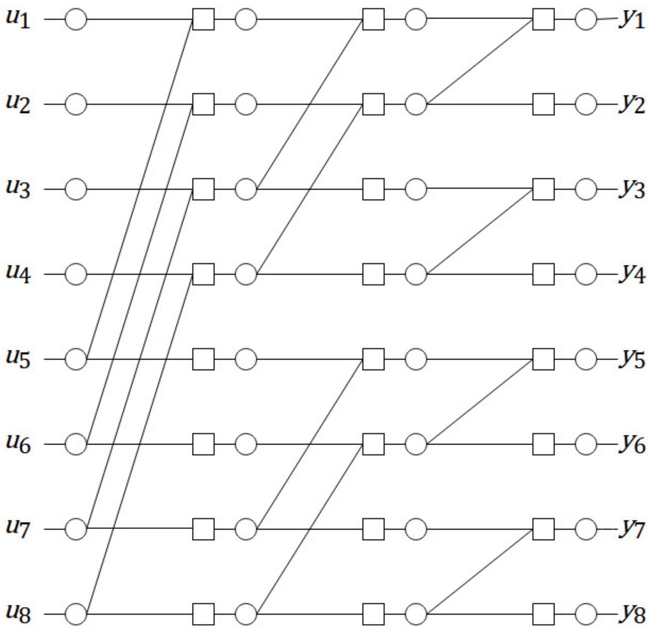


Figure 2.13: 3-2-1 factor graph with representing variable and check nodes.

In Fig. 2.13, the bits on the leftmost column represent the input of the FG. Each stage consists of $N/2 = 4$ processing elements, which correspond to the Z connections with two input and two output nodes in the FG representations. Details of the processing element of BP decoder is given in Fig. 2.14, where the input and output of the node (i, j), where $i = 0, \dots, n$ shows the stage number and $j = 1, \dots N$, are denoted by $v_I(i, j)$ and $v_O(i, j)$.

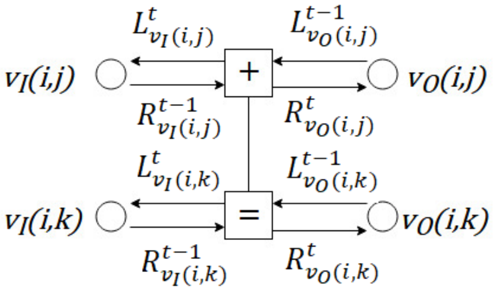


Figure 2.14: Details of the processing element for the BP decoder

The left-directed message passing the node (i,j) is represented by $L_{i,j}$ and the right-directed message crossing the node (i,j) is denoted as $R_{i,j}$. These two types of messages (or log-likelihood ratios (LLRs)) move and update iteratively between adjacent nodes. In Fig. 2.14, $L^t_{v_I(i,j)}$ and $L^t_{v_I(i,k)}$ show the right-to-left probability messages coming to the input nodes $v_I(i,j)$ and $v_I(i,k)$, where *t* corresponds to the iteration number. The other probability messages that move towards left, $L^{t-1}_{v_O(i,j)}$ and $L^{t-1}_{v_O(i,k)}$, are passed from the output nodes $v_O(i,j)$ and $v_O(i,k)$. Left-to-right messages are denoted as $R^t_{v_O(i,j)}$, $R^t_{v_O(i,k)}$, $R^{t-1}_{v_I(i,j)}$ and $R^{t-1}_{v_I(i,k)}$ in the processing element representation. During the belief propagation process, these message probabilities are updated and calculated as [Zhang, 2014]

$$
\begin{aligned}
L^t_{v_I(i,j)} &= L^{t-1}_{v_O(i,j)} \otimes \left[L^{t-1}_{v_O(i,k)} \odot R^{t-1}_{v_I(i,k)}\right],\\
L^t_{v_I(i,k)} &= \left[R^{t-1}_{v_I(i,j)} \otimes L^{t-1}_{v_O(i,j)}\right] \odot L^{t-1}_{v_O(i,k)},\\
R^t_{v_O(i,j)} &= R^{t-1}_{v_I(i,j)} \otimes \left[R^{t-1}_{v_I(i,k)} \odot L^{t-1}_{v_O(i,k)}\right],\\
R^t_{v_O(i,k)} &= \left[R^{t-1}_{v_I(i,j)} \otimes L^{t-1}_{v_O(i,j)}\right] \odot R^{t-1}_{v_I(i,k)}\,. \qquad (2.4)
\end{aligned}
$$

Two operations $\otimes$ and $\odot$ used in equations (2.4) are defined as

$$
\begin{aligned}
(p_x \otimes p_y)(0) &= p_x(0) \times p_y(0) + p_x(1) \times p_y(1),\\
(p_x \otimes p_y)(1) &= p_x(0) \times p_y(1) + p_x(1) \times p_y(0),\\
(p_x \odot p_y)(0) &= p_x(0) \times p_y(0),\\
(p_x \odot p_y)(1) &= p_x(1) \times p_y(1)\,. \qquad (2.5)
\end{aligned}
$$

In equations (2.5), $p_x(0)$ is the probability of "variable x equals 0", and $p_x(1)$ is the probability of "variable $x = 1$".

BP decoding algorithm that we have used can be summarized as follows.

Firstly, input frozen bits on the leftmost column of the FG are assigned 0. In the intermediate stages of the FG, if all bits added to one node from the previous stage are frozen, then this node also becomes frozen. In Figure 2.15, frozen variables are illustrated on the processing element of the BP decoder for 4 possible situations, where frozen nodes are colored red.

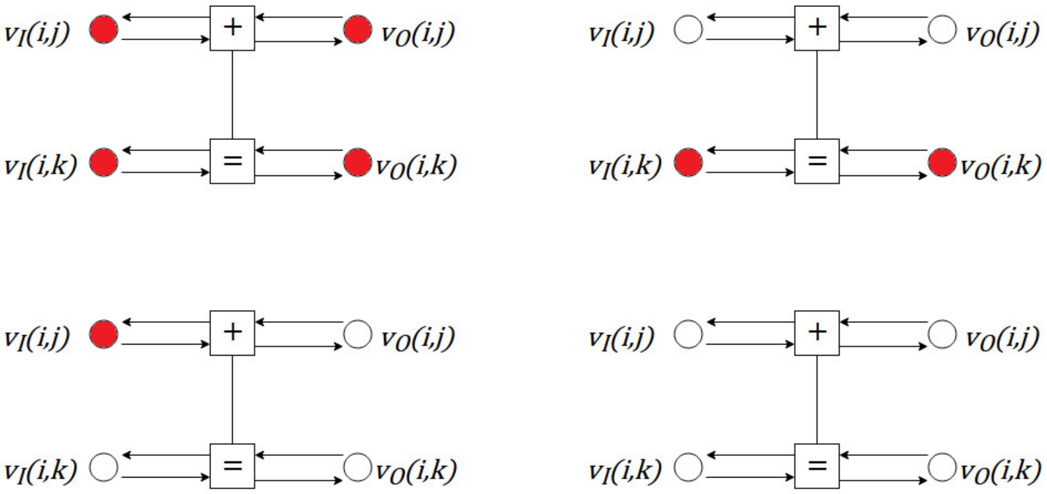


Figure 2.15: Frozen variables of the processing element for 4 different situations.

After finding all frozen variables in the FG representation, one assigns 0 to all frozen variables and they remain 0 throughout the BP decoding algorithm.

Decoding starts from the rightmost end of the FG. Received word, $v_O(n,j)$ for $j = 1,2,\dots,N$, is the output of the last (n'th) stage of the FG. We assign 0.5 for all erasures. In our decoding algorithm that uses the same procedure as in [Doğan, 2015] and [Xu et al., 2015], we start from the output nodes that correspond to the rightmost column of the FG and proceed to the leftmost by shifting only the left-directed messages, from the output of the last stage to input of the first stage. Initial probabilities $L^{0}_{v_o(n,j)}$ of left messages are determined according to the received word $r(i)$ as

$$L^{0}_{v_o(n,j)} = \begin{cases} 0 & if\ r(i) = 0 \\ 1 & if\ r(i) = 1 \\ 0.5 & if\ r(i) = e \end{cases} ,$$

at the output of the last stage. Then, the algorithm shifts messages from the input of the first stage to the rightmost of the FG with only right-oriented messages; and a single iteration is completed when the right-oriented messages reach the output nodes.

Left and right messages are updated by (2.4) during the iteration process. After completing one iteration, the decoding algorithm checks whether there is an erasure at output nodes $v_O(n,j)$ for $j = 1,2,\dots,N$ or not. If there is no erasure, the decoding algorithm stops. If there is still an erasure, decoding algorithm continues shifting messages from right to left and left to right until it reaches the pre-assigned “maximum number of iterations”. We observe that after a certain number of iterations, the number of the undecoded words does not decrease and their erasures cannot be corrected. The maximum number of iterations that the decoding algorithm needs to perform is assigned at the beginning, depending on the length N of the codeword, and the erasure rate ϵ of the binary erasure channel.

2.5. Relation Between the Bhattacharya Parameters and the Capacity Over Binary Erasure Channels

For the binary erasure channel BEC(ϵ) shown in Figure 2.16, the relationship between the channel capacity $I(W)$ and the Bhattacharya Parameter $Z(W)$ is explained in this section.

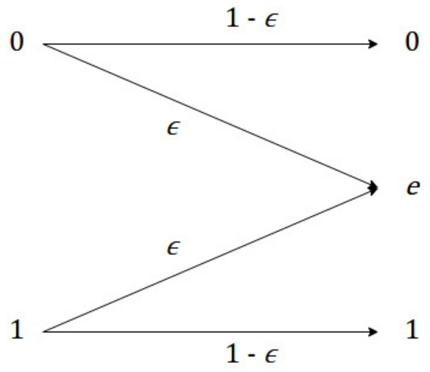


Figure 2.16: Binary erasure channel BEC(ϵ)

For a BEC(ϵ), input and output alphabets are $X = \{0,1\}$, $Y = \{0,1,e\}$ respectively. So, Bhattacharyya parameter for a BEC is calculated as

$$Z(W) = \sqrt{W(0|0)W(0|1)} + \sqrt{W(e|0)W(e|1)} + \sqrt{W(1|0)W(1|1)} \ .$$

$W(0|0)W(0|1)$ and $W(1|0)W(1|1) = 0$, so, $Z(W) = \sqrt{W(e|0)W(e|1)} = \epsilon$.

In other words, $Z(W)$ equals to the erasure probability of the binary erasure channel. Moreover, the capacity of the BEC(ϵ) is calculated as

$$C = max\ I(X;Y) = max\big(H(X) - H(X|Y)\big)$$

If $Y = 0$ or $Y = 1$, then X is known exactly, so $H(X|Y \neq e) = 0$. Moreover, $H(X|Y = e) = H(X)$, so

$$\begin{aligned} H(X|Y) &= H(X|Y \neq e)P(Y \neq e) + H(X)P(Y = e) \\ &= H(X)P(Y = e) \\ &= H(X) \times \epsilon \end{aligned}$$

and $I(X;Y)$ equals

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) = (1 - \epsilon)H(X) \\ &\leq 1 - \epsilon \end{aligned}$$

Hence $C = max\ I(X;Y) = 1 - \epsilon$.

So, the capacity $I(W)$ of the binary erasure channel (BEC) is equal to $1 - Z(W)$.

For $N = 2^n, n \geq 0, 1 \leq i \leq N$ and BEC(ϵ), Bhattacharya parameters can be computed through the recursion

$$Z({W_N}^{(2i-1)}) = 2Z({W_{N/2}}^{(i)}) - Z({W_{N/2}}^{(i)})^2$$

$$Z({W_N}^{(2i)}) = Z({W_{N/2}}^{(i)})^2 \tag{2.6}$$

where $Z({W_1}^{(1)}) = \epsilon$. Also, channel capacities $\{I({W_N}^{(i)})\}$ can be computed using the recursive relations

$$I({W_N}^{(2i)}) = 2I({W_{N/2}}^{(i)}) - I({W_{N/2}}^{(i)})^2$$

$$I({W_N}^{(2i-1)}) = I({W_{N/2}}^{(i)})^2,$$

where $I({W_1}^{(1)}) = 1 - Z({W_1}^{(1)}) = 1 - \epsilon$. These recursions can be used to show that $I({W_N}^{(i)}) = 1 - Z({W_N}^{(i)})$ for a BEC [Arıkan, 2009]. However, this relation between the Bhattacharya parameters and channel capacities is obtained for the synthesized channel W_N, which is represented by the "*n*-...-2-1" stage ordered FG.

As was mentioned in Section 2.3, for polar (or RM) codes of blocklength $N = 2^n$, there are n! different FG's. By changing the stage order of an FG, the order of channel capacities may also change. For the synthesized channel W_8, which is represented by the 3-2-1 factor graph, channel capacities of each subchannel over BEC(0.35) was given in Figure 2.12. We now present in Table 2.1, the channel capacities for $N = 8$ over BEC(0.35) for each of the 6 possible FG's. One observes that each FG has a different order of channel capacities.

On the other hand, Bhattacharya parameters, which are obtained by using (2.6), do not change with changing stage order and they remain the same for all FG's. In Table 2.1, capacities that match with $1 - Z({W_N}^{(i)})$ values are given in bold numbers to demonstrate that only the reference 3-2-1 FG satisfies the equality: $I({W_N}^{(i)}) = 1 - Z({W_N}^{(i)})$.

Table 2.1: Channel capacities over BEC(0.35) of all factor graphs as compared to (1— Bhattacharya parameters) for $N = 8$.

Input Bit Positions (i)	$1 - Z({W_N}^{(i)})$	$I({W_N}^{(i)})$ for 3-2-1 Factor Graph	$I({W_N}^{(i)})$ for 3-1-2 Factor Graph	$I({W_N}^{(i)})$ for 2-3-1 Factor Graph	$I({W_N}^{(i)})$ for 2-1-3 Factor Graph	$I({W_N}^{(i)})$ for 1-3-2 Factor Graph	$I({W_N}^{(i)})$ for 1-2-3 Factor Graph
1	**0.0319**	**0.0319**	**0.0319**	**0.0319**	**0.0319**	**0.0319**	**0.0319**
2	**0.5929**	**0.5929**	0.4442	**0.5929**	0.4442	0.3251	0.3251
3	**0.4442**	**0.4442**	0.5929	0.3251	0.3251	0.5929	**0.4442**
4	**0.9702**	**0.9702**	**0.9702**	0.9471	0.8888	0.9471	0.8888
5	**0.3251**	**0.3251**	**0.3251**	0.4442	0.5929	0.4442	0.5929
6	**0.9471**	**0.9471**	0.8888	0.9702	0.9702	0.8888	**0.9471**
7	**0.8888**	**0.8888**	0.9471	**0.8888**	0.9471	0.9702	0.9702
8	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**

For an (N, K) polar code designed over a BEC, the choice of the K information bits depends on the channel capacities and the Bhattacharya parameters. Relation $I({W_N}^{(i)}) = 1 - Z({W_N}^{(i)}), i = 1, \dots, N,$ that is valid for the "n-...-2-1" stage ordered factor graph [Arıkan, 2009] ensures that channels with highest capacities also have the least possible Bhattacharya parameters. Therefore, channels of the "n-...-2-1" FG that are selected to send information, have the largest symmetric capacities $I({W_N}^{(i)})$, and the smallest Bhattacharya parameters $Z({W_N}^{(i)})$.

Because of this $I({W_N}^{(i)}) = 1 - Z({W_N}^{(i)})$ characteristic of the "n-...-2-1" factor graph, we call it the reference factor graph (RFG), and adjust the input information bit positions of all other FG's of the polar code according to this reference FG.

As mentioned in Section 2.3, the information bit indices corresponding to the largest-capacity bit channels on the reference factor graph (RFG) of the (N, K) polar encoder select K basis vectors from the $N \times N$ transformation matrix $G_N = F^{\otimes n}$, which construct the $K \times N$ generator matrix G_P. A question of interest to be discussed in Chapter 3 is, whether the transformation matrix G_N differs or remains the same for FG’s with different stage permutations. Although stage transitions seem to be linear, it is not clear whether these linear transitions are permutable without affecting the overall transformation matrix G_N, because $N \times N$ matrix multiplication is not commutative in general.

CHAPTER 3

TRANSFORMATION MATRIX FOR DIFFERENT STAGE PERMUTATIONS

In this chapter, we discuss whether the generator matrix G_P of an (N, K) code changes, when the stage order of the FG is changed while keeping the positions of the information bits the same as those of the reference, “n-...-2-1” factor graph. Since the information bit positions are fixed, selected row indices from the transformation matrix G_N remain the same for all stage permutations. This implies that if G_N is invariant under stage permutations, so is the generator matrix G_P.

For polar codes, the effect of stage permutations on the encoder graph is discussed, and permutation-invariance of the transformation matrix $G_N = F^{\otimes n}$ is proved in [Vangala, 2014] by using the properties of the Kronecker product operation. In this chapter, we use a different approach to prove the invariance of the transformation matrix under stage permutations. In Section 3.1, we decompose the transformation matrix G_N into stage matrices $G_N^{(i)}$ and discuss their properties. In Section 3.2, by using the properties of the stage matrices, we show that $n!$ different factor graphs have the same transformation matrix G_N; in other words, both the $N \times N$ transformation matrix G_N, and the $K \times N$ generator matrix G_P are invariant under stage permutations, provided that the input information bit positions are kept fixed.

3.1. Decomposition of the Transformation Matrix G_N into Stage Matrices $G_N^{(i)}$

In this section, we factor the overall transformation G_N into n stage matrices for a block length $N = 2^n$, where each stage is represented by a particular $N \times N$ matrix.

We call the transformation matrix of the i'th stage, the stage matrix $G_N^{(i)}$, for $1 \le i \le n$. To simplify the description of a stage matrix $G_N^{(i)}$, we start with an example for $N = 8$.

Example 3.1: For $N = 8$, one can construct the stage matrices $G_8^{(1)}, G_8^{(2)}, G_8^{(3)}$ corresponding to three stages. Referring to the 1-2-3 factor graph (that is the first diagram in Figure 2.11), let us call the input vector u_1^8, and the outputs of the 1st, 2nd and 3rd stages a_1^8, b_1^8, and y_1^8 respectively. Notice that the basic Z structure represented by $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ is implemented by the smallest size Z's in Stage 1 of the graph. So, calling the input u_1^N, and the output of Stage 1, a_1^N, corresponding input-output relation can be expressed as $a_1^8 = u_1^8 G_8^{(1)}$, where the 8×8 stage matrix $G_8^{(1)}$is obtained by repeating four 2x2 submatrices (F), shown below in bold and normal letters successively as

$$G_8^{(1)} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{0} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Stage 2 contains 4 larger size Z's, represented by 3×3 submatrices $F^{(2)} = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} \\ 0 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} \end{bmatrix}$, obtained by lengthening F with an all-zero second row and all-zero second column. Corresponding stage matrix $G_8^{(2)}$ describes the input-output relation $b_1^8 = a_1^8 G_8^{(2)}$ of Stage 2, where we use bold and normal letters for four $(N/2)$ successive Z connections

$$G_8^{(2)} = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Stage 3 has the input-output relation $y_1^8 = b_1^8 G_8^{(3)}$ where the stage matrix, $G_8^{(3)}$, is formed using four 5×5 submatrices $F^{(3)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \end{bmatrix}$ obtained by lengthening F with all-zero 2nd, 3rd, 4th rows and all-zero 2nd, 3rd, 4th columns. These four $(N/2)$ submatrices are shown below in $G_8^{(3)}$, again by successive bold and normal letters, as

$$G_8^{(3)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Notice that the overall transformation matrix of the 1-2-3 factor graph can now be obtained by substituting the input-output relation of the 2nd and 1st stages into that of the 3rd stage as $y_1^8 = b_1^8 G_8^{(3)} = a_1^8 G_8^{(2)} G_8^{(3)} = u_1^8 G_8^{(1)} G_8^{(2)} G_8^{(3)} = u_1^8 G_8$. Hence, $G_8^{(1)} G_8^{(2)} G_8^{(3)} = G_8$.

After this example for $N = 8$, we generalize the descriptions of the submatrices $F^{(i)}$ and the stage matrices $G_N^{(i)}$ in Definition 3.1 and Definition 3.2.

Definition 3.1 - Submatrices $\boldsymbol{F^{(i)}},\ \boldsymbol{i = 1, \ldots, n}$:

$M_i \times M_i$ submatrices $F^{(i)}$ with $M_i = 2^{i-1} + 1$ of the form $F^{(i)} = \begin{bmatrix} \mathbf{1} & 0 & \ldots & 0 & \mathbf{0} \\ 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 \\ \mathbf{1} & 0 & \ldots & 0 & \mathbf{1} \end{bmatrix}$ are generated by lengthening F with $(M_i - 2)$ all-zero rows and $(M_i - 2)$ all-zero columns. The matrix $F^{(i)} = \{f_{jk}\},\ j, k = 1, \ldots, M_i,$ has all-zero entries except the three unity entries; i.e., $f_{11} = f_{M_i,1} = f_{M_iM_i} = 1$. Notice that $F^{(1)} = F$, and all submatrices $F^{(\boldsymbol{i})}$ are lower triangular just like F.

Definition 3.2 - Stage Matrices $\boldsymbol{G_N^{(i)}},\ \boldsymbol{i = 1, \ldots, n}$:

$N \times N$ stage matrices $G_N^{(i)}$ are formed by successively inserting $M_i \times M_i$ submatrices $F^{(i)}$ to proper diagonal positions of an all-zero $N \times N$ matrix so that none of the unity elements of the submatrices overlap. The submatrix size $M_i = 2^{i-1} + 1$, where $i = 1, \ldots, n$, is less than $N = 2^n$ for $n > 1$; and $G_N^{(i)}$ contains $N/2$ non-overlapping $M_i \times M_i$ submatrices of the form

$$F^{(i)} = \begin{bmatrix} \mathbf{1} & 0 & \ldots & 0 & \mathbf{0} \\ 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 \\ \mathbf{1} & 0 & \ldots & 0 & \mathbf{1} \end{bmatrix}.$$

Insertion of the $N/2$ submatrices $F^{(i)}$ into an all-zero $N \times N$ matrix is such that the first element $f_{11}^{(i)}$ of $F^{(i)}$ is equated to the first 2^{i-1} successive diagonal elements, $g_{jj}^{(i)}$, for $j = 1, \ldots, 2^{i-1}$, and the following 2^{i-1} successive diagonal positions are skipped. This procedure is repeated until all diagonal positions of $G_N^{(i)}$ are filled with one's.

Examples of $\boldsymbol{G_N^{(i)}}$ for $\boldsymbol{i = 1, 2, 3}$:

- The first element $f_{11}^{(1)}$ of the original $F = F^{(1)}$ matrix is inserted into the 1st, 3rd, 5th … and $(N-1)$st diagonal positions of an all-zero $N \times N$ matrix, by skipping its 2nd, 4th, 6th, … Nth diagonal elements, which are already occupied by $f_{22}^{(1)}$ of the previous insertions. $G_N^{(1)}$ is formed when all diagonal entries are filled with 1's.
- Similarly, the first element $f_{11}^{(2)}$ of the 3×3 $F^{(2)}$ matrix is inserted into the 1st, 2nd and 5th, 6th … and $(N-3)$rd, $(N-2)$nd diagonal elements of an all-zero $N \times N$ matrix by skipping its 3rd, 4th, … and $(N-1)$th , Nth … diagonal positions. $G_N^{(2)}$ is formed when all diagonal entries are filled with 1's.
- The 5×5 matrix $F^{(3)}$ is inserted into the 1st, 2nd , 3rd, 4th, …., $(N-7)$th, $(N-6)$th, $(N-5)$th, $(N-4)$th rows/columns of an all-zero $N \times N$ matrix by skipping its 5th , 6th, 7th , 8th … rows/columns until $G_N^{(3)}$is formed.

Properties of the Stage Matrix $\boldsymbol{G_N^{(i)} = \left\{g_{jk}^{(i)}\right\}, j, k = 1, \dots, N}$:

1. It is lower triangular, since each lower triangular submatrix $F^{(i)}$ is inserted into a diagonal element of $G_N^{(i)}$ having the same row/column index.
2. All diagonal elements $g_{jj}^{(i)}$ are equal to 1.
3. There are $N + \frac{N}{2}$ nonzero elements of $G_N^{(i)}$ that are equal to one. N of them are on the diagonal, remaining $\frac{N}{2}$ one's are on the m'th subdiagonal, having the elements $g_{j,(j-m)}^{(i)}$, where $m = 2^{i-1}$ and $j = (2^{i-1}+1), \dots, N$.
4. $\frac{N}{2}$ one's on the (2^{i-1})'th subdiagonal have the specific positions defined by

$$g_{j,(j-2^{i-1})}^{(i)} = \begin{Bmatrix} 1, & if\ (j-1) mod (2^i) \geq\ 2^{i-1} \\ 0, & else \end{Bmatrix}.$$

3.2. Invariance of the Transformation Matrix $\mathrm{G_N}$ under Stage Permutations

Now we will remember the mathematical definitions of “persymmetry” and “flip transpose” and prove a sequence of lemmas to arrive at the main conclusion given by Corollary 3.1 of Proposition 3.1; that is, the input-output relation describing any of the n-stage-factor graphs is invariable under the permutation of stages. In other words, all of the n! different factor graphs have the same transformation matrix G_N in their input-output relation $y_1^N = u_1^N G_N$.

Definition 3.3 (Flip Transpose and Persymmetric Matrices):

Flip transpose A^F of a matrix $A = \{a_{ij}\}$ is defined as $A^F = B = \{b_{ij}\} = \{a_{N-j+1,\ N-i+1}\}$.

If A is a persymmetric matrix, its elements satisfy $a_{ij} = a_{N-j+1,\ N-i+1}$ for all $i, j = 1, \ldots, N$. Hence, $A = A^F$ and A is symmetrical with respect to its skew diagonal combining the upper right element a_{1N} to the lower left element a_{N1}.

Notice that the identity matrix I is both symmetric and persymmetric; that is $I^F = I^T = I$.

One can also show that the “flip transpose of a product”, $(AB)^F = B^F A^F$ can be found by the same rule applied to the “transpose of a product”; i.e., $(AB)^T = B^T A^T$.

Lemma 3.1: For a blocklength $N = 2^n$, the transformation matrix G_N of the “1-2-…-n” factor graph can be calculated as $G_N = G_N^{(1)} G_N^{(2)} \ldots G_N^{(n)}$.

Proof:

For a blocklength $N = 2^n$, the input-output relation is defined by the transformation

$$y_1^N = u_1^N G_N.$$

To simplify notation, we denote the N bit input and output vectors of the first stage by X_1 & Y_1, those of the second stage by X_2 & Y_2 and those of the n'th stage by X_n & Y_n. If the stages are in the order "1-2-…-n", corresponding input-output equations are

$Y_1 = X_1 G_N^{(1)}$, $Y_2 = X_2 G_N^{(2)}$, … , $Y_n = X_n G_N^{(n)}$, where $Y_n = y_1^N$, $X_1 = u_1^N$, and $X_i = Y_{i-1}$ for $i = 2, \dots, n$. Hence the input-output equation from the first to the second stage can be written as $Y_2 = X_2 G_N^{(2)} = Y_1 G_N^{(2)} = X_1 G_N^{(1)} G_N^{(2)}$.

Now assuming that the input-output equation from Stage 1 to Stage $(n-1)$ is

$$Y_{n-1} = X_1 G_N^{(1)} G_N^{(2)} \dots G_N^{(n-1)},$$

one can compute the overall input-output relation as

$$Y_n = X_n G_N^{(n)} = Y_{n-1} G_N^{(n)} = X_1 G_N^{(1)} G_N^{(2)} \dots G_N^{(n-1)} G_N^{(n)}$$

which shows that the overall transition matrix is $G_N = G_N^{(1)} G_N^{(2)} \dots G_N^{(n-1)} G_N^{(n)}$.

In the following part, our aim is to show that the multiplication order of these stage matrices can be changed without affecting the overall transformation. Hence the factor graphs implemented by $n!$ different stage permutations have the same input-output relation. Following definitions and lemmas are needed to approach this aim step by step.

Lemma 3.2 Submatrix $F^{(i)}$ of an arbitrary i^{th} stage, $1 \leq i \leq n$, is a persymmetric matrix.

Proof:

The $M_i \times M_i$ submatrices $F^{(i)} = \begin{bmatrix} \mathbf{1} & 0 & \dots & 0 & \mathbf{0} \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ \mathbf{1} & 0 & \dots & 0 & \mathbf{1} \end{bmatrix}$, with $M_i = 2^{i-1} + 1$, are generated by lengthening F with $(M_i - 2)$ all-zero rows and $(M_i - 2)$ all-zero columns, So, $F^{(i)}$ has all-zero entries f_{jk} except the three unity entries; i.e., $f_{11} = f_{M_i1} = f_{M_iM_i} = 1$. The element f_{M_i1} is on the skew diagonal and the equation $f_{11} = f_{M_iM_i}$ satisfies the condition of persymmetry. Hence, $F^{(i)}$'s are symmetric about their cross diagonal (combining the upper right-end entry f_{1M_i} to the lower left-end entry f_{M_i1}); i.e., they are persymmetric, hence $\left(F^{(i)}_{M_i \times M_i}\right)^F = F^{(i)}_{M_i \times M_i}$.

Lemma 3.3 The stage matrix $G_N^{(i)}$ of an arbitrary i^{th} stage, $1 \le i \le n$, is also a persymmetric matrix.

Proof:

For $n = 1$, there is a single stage and the stage matrix equals to $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. It is a persymmetric matrix, because it is symmetrical with respect to its skew-diagonal.

For $n = 2$, block length is $N = 2^n = 4$ and there are two stage matrices; $G_4^{(1)}$ and $G_4^{(2)}$:

$$G_4^{(1)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} \end{bmatrix}, \quad G_4^{(2)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & \mathbf{1} \end{bmatrix}.$$

Stage matrices; $G_4^{(1)}$ and $G_4^{(2)}$, satisfy $a_{ij} = a_{N-j+1,\ N-i+1}$ for all $i, j = 1, \dots, N = 4$, so, they are persymmetric (for $n = 3$, persymmetry can be observed in Example 3.1).

We also observe that $G_8^{(1)}$and $G_8^{(2)}$ stage matrices can respectively be written in terms of $G_4^{(1)}$and $G_4^{(2)}$ stage matrices as

$$G_8^{(1)} = \begin{bmatrix} (G_4^{(1)})_{4\times 4} & (0)_{4\times 4} \\ (0)_{4\times 4} & (G_4^{(1)})_{4\times 4} \end{bmatrix},\ G_8^{(2)} = \begin{bmatrix} (G_4^{(2)})_{4\times 4} & (0)_{4\times 4} \\ (0)_{4\times 4} & (G_4^{(2)})_{4\times 4} \end{bmatrix}.$$

In general, when increasing the stage number from n to $n+1$, block length increases from $2^n = N$ to $2^{n+1} = 2N$. For an arbitrary i^{th} stage, $1 \leq i \leq n$, dimensions of the submatrix $F^{(i)}$ does not change with increasing stage number. $G_{2N}^{(i)}$ is generated by inserting $F^{(i)}$ to the first 2^{i-1} successive positions (in diagonal entries with starting from g_{11}) and skipping the following 2^{i-1} successive positions. From g_{11} to g_{NN}, by inserting $F^{(i)}$ to the first 2^{i-1} successive positions and skipping the following 2^{i-1} successive positions, we obtain the stage matrix $G_N^{(i)}$, in which $F^{(i)}$'s is repeated 2^{n-1} times. From $a_{N+1,N+1}$ to $a_{2N,2N}$, $F^{(i)}$'s are inserted 2^{n-1} times in the same manner and again we obtain $G_N^{(i)}$ stage matrix. In total, $F^{(i)}$'s are repeated 2^n times in $G_{2N}^{(i)}$ and $G_{2N}^{(i)}$ can be written in terms of the stage matrix $G_N^{(i)}$ as

$$G_{2N}^{(i)} = \begin{bmatrix} (G_N^{(i)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(i)})_{N\times N} \end{bmatrix}.$$

In order to prove Lemma 3.3 by induction, we first observe the persymmetry for $n = 1, 2$. Then we let $n = k$, corresponding to the block length $N = 2^k$, and assume that, the stage matrix of an arbitrary i^{th} stage, $1 \leq i \leq n$, $G_N^{(i)}$ is persymmetric; that is $(\left(G_N^{(i)}\right)_{N\times N})^F = G_N^{(i)}$.

For $n = k+1$, block length equals to $2^{k+1} = 2 \times 2^k = 2N$ and we need to show that the stage matrix of an arbitrary i^{th} stage, $1 \leq i \leq k+1$, $G_{2N}^{(i)}$ is persymmetric.

One can write the stage matrix of the i^{th} stage, $G_{2N}^{(i)}$ for $1 \leq i \leq k$, as

$$G_{2N}^{(i)} = \begin{bmatrix} (G_N^{(i)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(i)})_{N\times N} \end{bmatrix}$$

By taking the flip-transpose of $G_{2N}^{(i)}$, we obtain

$$(G_{2N}^{(i)})^F = \begin{bmatrix} (\left(G_N^{(i)}\right)_{N\times N})^F & ((0)_{N\times N})^F \\ ((0)_{N\times N})^F & (\left(G_N^{(i)}\right)_{N\times N})^F \end{bmatrix}$$

$$= \begin{bmatrix} (G_N^{(i)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(i)})_{N\times N} \end{bmatrix}$$

Using the persymmetry assumption in the last line, we observe that $(G_{2N}^{(i)})^F = G_{2N}^{(i)}$, hence, the stage matrix $G_{2N}^{(i)}$ is persymmetric for an arbitrary i^{th} stage, $1 \leq i \leq k.$

For $i = k + 1$, where $2^{k+1} = 2N,$ using the properties of the stage matrices, $G_{2N}^{(k+1\,)}$ can be written as

$$G_{2N}^{(k+1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

$G_{2N}^{(k+1)}$ has $2N$ unity entries on the diagonal (from a_{11} to $a_{2N,2N}$) and N one's on the subdiagonal (from $a_{1,N+1}$ to $a_{N,2N}$). Therefore, $G_{2N}^{(k+1)}$ can be partitioned into four $N \times N$ blocks as:

$$G_{2N}^{(k+1)} = \begin{bmatrix} I_{N\times N} & (0)_{N\times N} \\ I_{N\times N} & I_{N\times N} \end{bmatrix}$$

By taking the flip-transpose of $G_{2N}^{(k+1)}$, one obtains

$$(G_{2N}^{(k+1)})^F = \begin{bmatrix} (I_{N\times N})^F & ((0)_{N\times N})^F \\ (I_{N\times N})^F & (I_{N\times N})^F \end{bmatrix} = \begin{bmatrix} I_{N\times N} & (0)_{N\times N} \\ I_{N\times N} & I_{N\times N} \end{bmatrix} = G_{2N}^{(k+1)}$$

So, the flip-transpose of the stage matrix, $G_{2N}^{(k+1\,)}$ equals to itself and it is persymmetric. Hence, the assumption which is made for $n = k$, also holds for $n = k + 1$, and $G_{2N}^{(i)}$ is a persymmetric matrix for an arbitrary i^{th} stage, $1 \leq i \leq k + 1$.

We have thus shown by induction that the stage matrix of an arbitrary i^{th} stage, $1 \leq i \leq n$, $G_N^{(i)}$ is a persymmetric matrix.

Not only the stage matrices, but their products are also persymmetric as demonstrated for a blocklength $N = 8$ in Example 3.2 and proved in Lemma 3.4.

Example 3.2:

For a blocklength $N = 8$, where $n = 3$, there are three stage matrices; $G_8^{(1)}, G_8^{(2)}, G_8^{(3)}$. The pairwise products of stage matrices can be found as:

$$A = G_8^{(1)} G_8^{(2)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

$$B = G_8^{(1)} G_8^{(3)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

$$C = G_8^{(2)} G_8^{(3)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \end{bmatrix}$$

One observes that the pairwise product matrices A, B and C are all persymmetric because elements of these matrices satisfy $a_{ij} = a_{N-j+1,\ N-i+1}$ for all $i, j = 1, \dots, 8$.

Lemma 3.4: Product $G_N^{(r)} G_N^{(s)}$ of the stage matrices $G_N^{(r)}$ and $G_N^{(s)}$ of two arbitrary stages ($1 \leq r, s \leq n$ and $r \neq s$) is persymmetric.

Proof: In order to make the proof by induction, we first show that for $n = 2$, $G_4^{(r)} G_4^{(s)}$ is persymmetric for all $1 \leq r, s \leq n$.

For $n = 2$ and block length $N = 4$, there are two stage matrices; $G_4^{(1)}$ and $G_4^{(2)}$. Product of $G_4^{(1)}$ and $G_4^{(2)}$ equals to a persymmetric matrix, which is symmetrical with respect to its cross diagonal.

$$G_4^{(1)} G_4^{(2)} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix} = G_4^{(2)} G_4^{(1)}$$

Note that this multiplication is also commutative as will be generalized in Proposition 3.1. Let us now assume that the product of the stage matrices $G_N^{(r)} G_N^{(s)}$ of two arbitrary stages ($1 \leq r, s \leq n$ and $r \neq s$) is persymmetric for $n = k,\ k \geq 1$ and $N = 2^k$.

For $n = k + 1$, blocklength equals to $2^{(k+1)} = 2N$. Then, $2N \times 2N$ stage matrices of the r^{th} and s^{th} stages for $1 \leq r, s \leq k$, $G_{2N}^{(r)}$ and $G_{2N}^{(s)}$, are equal to

$$G_{2N}^{(r)} = \begin{bmatrix} (G_N^{(r)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(r)})_{N\times N} \end{bmatrix},\ G_{2N}^{(s)} = \begin{bmatrix} (G_N^{(s)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(s)})_{N\times N} \end{bmatrix}$$

Then, the product of $G_{2N}^{(r)}$ and $G_{2N}^{(s)}$equals

$$G_{2N}^{(r)} G_{2N}^{(s)} = \begin{bmatrix} (G_N^{(r)} G_N^{(s)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(r)} G_N^{(s)})_{N\times N} \end{bmatrix}$$

The flip-transpose of $G_{2N}^{(r)} G_{2N}^{(s)}$ equals to

$$\left(G_{2N}^{(r)} G_{2N}^{(s)}\right)^F = \begin{bmatrix} \left(\left(G_N^{(r)} G_N^{(s)}\right)_{N\times N}\right)^F & ((0)_{N\times N})^F \\ ((0)_{N\times N})^F & \left(\left(G_N^{(r)} G_N^{(s)}\right)_{N\times N}\right)^F \end{bmatrix}$$

$$= \begin{bmatrix} (G_N^{(r)} G_N^{(s)})_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & (G_N^{(r)} G_N^{(s)})_{N\times N} \end{bmatrix}$$

We obtain that $\left(G_{2N}^{(r)} G_{2N}^{(s)}\right)^F = G_{2N}^{(r)} G_{2N}^{(s)}$. So, the product of two stage matrices $G_{2N}^{(r)}$ and $G_{2N}^{(s)}$ is persymmetric for two arbitrary stages ($1 \le r, s \le k$ and $r \ne s$).

For $n = k + 1$, there is an additional stage for $i = k + 1$, and $G_{2N}^{(k+1)}$ can be written as

$$G_{2N}^{(k+1)} = \begin{bmatrix} I_{N\times N} & (0)_{N\times N} \\ I_{N\times N} & I_{N\times N} \end{bmatrix}.$$

Product of $G_{2N}^{(k+1)}$ and the stage matrix of an arbitrary stage $G_{2N}^{(j)}$, $1 \le j \le k$ equals to:

$$G_{2N}^{(k+1)} G_{2N}^{(j)} = \begin{bmatrix} I_{N\times N} & (0)_{N\times N} \\ I_{N\times N} & I_{N\times N} \end{bmatrix} \begin{bmatrix} \left(G_N^{(j)}\right)_{N\times N} & (0)_{N\times N} \\ (0)_{N\times N} & \left(G_N^{(j)}\right)_{N\times N} \end{bmatrix}$$

$$= \begin{bmatrix} I_{N\times N}\left(G_N^{(j)}\right)_{N\times N} & (0)_{N\times N} \\ I_{N\times N}\left(G_N^{(j)}\right)_{N\times N} & I_{N\times N}\left(G_N^{(j)}\right)_{N\times N} \end{bmatrix}$$

$$= \begin{bmatrix} \left(G_N^{(j)}\right)_{N\times N} & (0)_{N\times N} \\ \left(G_N^{(j)}\right)_{N\times N} & \left(G_N^{(j)}\right)_{N\times N} \end{bmatrix}$$

By taking the flip-transpose of $G_{2N}^{(k+1)} G_{2N}^{(j)}$, we obtain

$$\left(G_{2N}^{(k+1)} G_{2N}^{(j)}\right)^F = \begin{bmatrix} \left(\left(G_N^{(j)}\right)_{N\times N}\right)^F & \left((0)_{N\times N}\right)^F \\ \left(\left(G_N^{(j)}\right)_{N\times N}\right)^F & \left(\left(G_N^{(j)}\right)_{N\times N}\right)^F \end{bmatrix}$$

$$= \begin{bmatrix} \left(G_N^{(j)}\right)_{N\times N} & (0)_{N\times N} \\ \left(G_N^{(j)}\right)_{N\times N} & \left(G_N^{(j)}\right)_{N\times N} \end{bmatrix} = G_{2N}^{(k+1)} G_{2N}^{(j)}$$

Flip-transpose of $G_{2N}^{(k+1)} G_{2N}^{(j)}$ equals to itself, so $G_{2N}^{(k+1)} G_{2N}^{(j)}$ is also persymmetric. We obtain that for $n = k + 1$, the product of two stage matrices $G_{2N}^{(r)}$ and $G_{2N}^{(s)}$ is persymmetric for two arbitrary stages ($1 \leq r, s \leq k + 1$ and $r \neq s$).

It shows that for $n = k + 1$, assumption that is made for $n = k,\ k \geq 1$ holds. By the principle of induction, it is true for all $n \geq 1$.

Proposition 3.1:

Product of the stage matrices $G_N^{(i)}$, for $1 \leq i \leq n$, is commutative; i.e.; $G_N^{(i)} G_N^{(j)} = G_N^{(j)} G_N^{(i)}$.

Proof:

The stage matrices $G_N^{(i)}$ are persymmetric matrices, $g_{ik} = g_{N-k+1,N-i+1}$ for all i, k; as shown in Lemma 3.3.

Let $A = G_N^{(r)} = \{g_{ij}\} \in \mathrm{GF}(2)$ and $B = G_N^{(s)} = \{g'_{ij}\} \in \mathrm{GF}(2)$ be the stage matrices of two arbitrary stages for $1 \leq r, s \leq n$ and $r \neq s$.

Calling $AB = C$ and $BA = D$, the entries c_{ij} and d_{ij} of C and D matrices are computed as

$$c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj} \quad \text{and } d_{ij} = \sum_{k=1}^{N} b_{ik} a_{kj} \, .$$

Flip-transpose of a persymmetric matrix equals to itself; i.e., $A^F = A$, where A^F shows the flip-transpose of A, which flips A across its skew diagonal. So, if

$$A = \{a_{ij}\}, \text{ then } A^F = \{a_{N-j+1, N-i+1}\}.$$

From Lemma 3.4, we know that the product of stage matrices is also persymmetric, so C and D matrices are persymmetric matrices. Hence,

$$C = C^F = (AB)^F = B^F A^F = BA = D \, , \tag{3.1}$$

where the last line in (3.1) follows because the flip-transposes of the persymmetric A and B matrices are equal to themselves.

So, $A = G_N^{(r)}$, and $B = G_N^{(s)}$ are commutative $N \times N$ matrices; i.e., $G_N^{(r)} G_N^{(s)} = G_N^{(s)} G_N^{(r)}$ for $1 \leq r, s \leq \boldsymbol{n}$**.**

Corollary 3.1:

For a block length $N = 2^n$, the transformation matrix G_N of the "1-2-…-*n*" factor graph calculated as $G_N = G_N^{(1)} G_N^{(2)} \ldots G_N^{(n)}$ remains fixed for all other factor graphs with different stage orders.

Proof:

In Proposition 3.1, we show that the stage matrices $G_N^{(i)}$ commute under multiplication. By using the commutative property, in $G_N = G_N^{(1)} G_N^{(2)} \ldots G_N^{(n)}$, one can pairwise change the order of $G_N^{(i)}$'s to obtain $n!$ different permutations. Hence, the overall transformation matrix G_N remains fixed under stage permutations.

It is now clear that $(\log_2 N)!$ different FG representations of an (N, K) polar code with fixed frozen bit and information bit positions of the RFG yield the same generator matrix G_P; however, since the capacity ranking of transmission paths may change as mentioned in Chapter 2 and shown in Table 2.1 (see p.30), the capacity sum (CS) of the K largest-capacities may change from one FG to the other as will be revealed in the next chapter.

CHAPTER 4

SET CHOICE FOR MULTIPLE FACTOR GRAPH BP DECODING

In this chapter, we investigate the performance dependence of the BP decoder upon two parameters: the capacity sum (CS) [Doğan 2015] and the number of frozen variables (FV) of the implemented factor graph. For the factor graph (FG) implementation of an (N, K) code, the "capacity sum-CS" is defined as the sum of the capacities of the K transmission paths [Doğan, 2015] and FV is the "total number of frozen variables". Since $N - K$ input frozen bits form a subset of frozen variables in the overall graph, FV $\geq N - K$.

We simulate single-FG (1-FG) and multiple-FG (M-FG) decoders to evaluate the BP decoding performance of polar and Reed-Muller codes. Single-FG decoder uses a single factor graph; but multiple-FG decoder employs a pre-chosen set of $M > 1$ factor graphs, where all FG's within the set are used one after the other to decode a received word.

It is of interest to examine which choice of CS and FV values of the factor graphs gives the best single-FG and multiple-FG BP decoding performances. Dependence of the BP decoder performance on FV and CS values was previously discussed in [Doğan, 2015]. As expected, M-FG decoder performance improves with increasing M at the cost of increased complexity. So practical values of M should be kept small although there exists a large variety of $(\log_2 N)!$ factor graphs. The value of M can be optimized according to the needs of the user; so we examine performance dependence of the BP decoder on M. The criterion for the choice of M factor graphs among $(\log_2 N)!$ alternatives is another question that we seek for an answer. As differently from Doğan's choice of FG's, we restrict our search space to FG's with the highest number of frozen variables.

In Section 4.1 and 4.2, the capacity sum (CS) and the number of frozen variables (FV) parameters are briefly explained with simple examples for small values of $N =$ 8, 16, and 32. We then use the CS and FV parameters to classify the factor graphs of the (8, K), (16, 8) and (32, 16) codes. For the (32, 16) polar codes over BEC(0.35), we group factor graphs with the same FV value into 5 “equi-FV sets” and those with the same CS value into 10 "equi-CS sets". Similar classification is made in Section 4.3 for the (32, 16) RM codes, but it results in a single equi-CS set and a single equi-FV set; i.e., CS and FV values of the RM codes remain the same for all 120 FG’s.

In Section 4.4, single-FG and M-FG BP decoding performances of the (64, 32) and (128, 64) adaptive polar codes over BEC(0.35) are assessed. For evaluating 1-FG decoding performances, one FG from each equi-FV set is selected, and the effect of FV on the BP decoder is examined. For M-FG decoding, FG's in the maximum equi-FV set are used to evaluate the decoding performance. Factor graphs in the maximum equi-FV set are ranked in decreasing order of CS values and M-FG decoder uses the first M factor graphs from this set. Number of FG’s; i.e., M, is increased in steps of 4 and the choice of the optimum number of FG's is discussed for the (64, 32) and (128, 64) adaptive polar codes. In Section 4.5, single-FG and M-FG BP decoding performances of the (128, 64) Reed-Muller codes are examined and obtained results for M-FG BP decoding are compared with the performance of the (128, 64) polar codes. Lastly, in Section 4.6, the set of FG’s in M-FG decoders are reduced to 4-element equi-CS sets and for the (64, 32) adaptive polar code over BEC(0.35), 1-FG and 4-FG BP decoding performances of equi-CS sets are evaluated.

4.1 Capacity Sum of the K Active Paths of an (N, K) Code

It was mentioned in Section 2.3 that for the factor graph (FG) representation of an (N, K) polar code, $n = \log_2 N$ stages are used; so $n!$ different FG’s can be constituted by changing the order of stages with respect to each other. Each stage can be considered as a column of Z connections, where a Z connection connects two inputs to two outputs. In [Doğan, 2015], the stages are named as 1, 2, …, n, where Stage 1 contains the smallest Z's with adjacent inputs and Stage n contains the largest Z's that connects input nodes separated by 2^{n-1}. Depending on the positions of these stages with respect to each other, the corresponding FG is called "1-2-...-n" factor graph, "n-…-2-1" factor graph, etc.

As explained in Section 2.2, N channel capacities $I\left(U_i; Y^N, U^{i-1}\right) = I({W_N}^{(i)})$ are calculated by channel combining and splitting operations. To design a polar code, K information bits with the least possible Bhattacharya parameters are chosen and sent over K channels. In Section 2.5, we have discussed that for a BEC, n-...-2-1 factor graph satisfies the relation: $I({W_N}^{(i)}) = 1 - Z({W_N}^{(i)})$ between channel capacities and Bhattacharya parameters [Arıkan, 2009]. Hence, the choice of K active bits with minimum Bhattacharya parameters is equivalent to the choice of K maximum-capacity paths over a BEC, for the n-...-2-1 factor graph. Consequently, the parameter defined in [Doğan 2015] as the "Capacity Sum-CS" denoting the sum of the capacities of the K transmission paths is maximum for the n-...-2-1 factor graph. Therefore, we have named the n-...-2-1 factor graph as our "reference factor graph - RFG".

Ranking of the computed N channel capacities $I({W_N}^{(i)}) = I\left(U_i; Y^N, U^{i-1}\right)$ may change from one FG to another, although the set of N capacities remains the same. Since the positions of the selected K channels are fixed for all $n!$ different FG's, capacity sum (CS) of these K transmission paths may change with changing stage order. We reason that the CS value of a factor graph can be successfully used in differentiating how effective the related factor graph (or implementation) of the polar code is.

In Table 4.1, we have chosen $N = 8$ over a BEC(0.35) and computed 8 channel capacities of $3! = 6$ different factor graphs. One observes that 8 elements of the channel capacity set remain the same, but their positions change.

Table 4.1: Channel capacities over BEC(0.35) of all factor graphs for $N = 8$

Input Bit Positions	**Channel Capacities For**					
	3-2-1 FG	3-1-2 FG	2-3-1 FG	2-1-3 FG	1-3-2 FG	1-2-3 FG
1	0.0319	0.0319	0.0319	0.0319	0.0319	0.0319
2	0.5929	0.4442	0.5929	0.4442	0.3251	0.3251
3	0.4442	0.5929	0.3251	0.3251	0.5929	0.4442
4	0.9702	0.9702	0.9471	0.8888	0.9471	0.8888
5	0.3251	0.3251	0.4442	0.5929	0.4442	0.5929
6	0.9471	0.8888	0.9702	0.9702	0.8888	0.9471
7	0.8888	0.9471	0.8888	0.9471	0.9702	0.9702
8	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998

Table 4.2: For the $(8, K)$ polar codes, capacity sums (CS) over BEC(0.35) of all factor graphs

$(\boldsymbol{N}, \boldsymbol{K})$	**Capacity Sums For**					
	3-2-1 FG	3-1-2 FG	2-3-1 FG	2-1-3 FG	1-3-2 FG	1-2-3 FG
(8,1)	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**
(8,2)	**1.9700**	**1.9700**	1.9469	1.8886	1.9469	1.8886
(8,3)	**2.9171**	2.8588	**2.9171**	2.8588	2.8357	2.8357
(8,4)	**3.8059**	**3.8059**	**3.8059**	**3.8059**	**3.8059**	**3.8059**
(8,5)	**4.3988**	4.2501	**4.3988**	4.2501	4.1310	4.1310

In Table 4.2, we tabulate the CS values of the (8, K) adaptive polar codes over BEC(0.35). Capacity sums of the K transmission channels may change from one FG to the other, as observed in Table 4.2, because the channel capacities are permuted with changing stage order (as shown in Table 4.1), whereas the active bit positions are fixed and chosen with respect to the reference, 3-2-1 FG.

From Table 4.2, one can observe that the 3-2-1 reference graph has the maximum CS value as expected, for all values of K. On the other hand, some other FG’s can also have the maximum CS value as shown by bold numbers in Table 4.2. For instance, all factor graphs of the (8, 1) polar (or RM) code select the same maximum CS path, which is the 8th input bit with capacity 0.9998. Similarly, all factor graphs of the

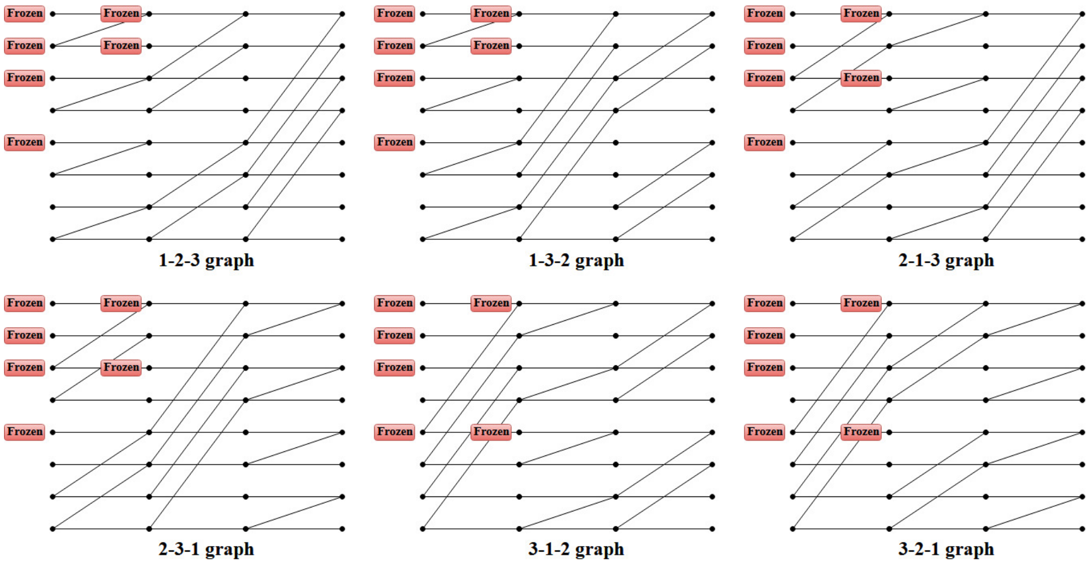
(8,4) polar (or RM) code select the same 4 paths, which are the maximum capacity paths (corresponding to the input bits 8, 7, 6 and 4 in Table 4.1, in varying order) and their CS values are therefore the same. Moreover, for $K = 2$, the 3-1-2 FG has the same CS value as the reference 3-2-1 FG; and for $K = 3$ and 5, the 2-3-1 FG has the maximum CS value equal to that of the reference FG.

Factor graphs of an (N, K) polar code can be grouped into sets of equal capacity sums. Doğan calls each set consisting of FG's with the same CS value, an "equi-CS set" [Doğan 2015]. For instance, there is only one equi-CS set containing 6 FG's for the (8, 1) and (8, 4) codes shown in Table 4.2; whereas the (8, 5) code has 3 equi-CS sets (with the CS values of 4.3988, 4.2501 and 4.1310) each consisting of 2 FG's.

4.2 Number of Frozen Variables in the Factor Graph of an (*N*, *K*) Code

$N - K$ frozen bits at the input of a factor graph may produce other frozen nodes at the following stages. If all bits merging at a given node are frozen, then this node also becomes frozen. Total number of frozen bits in the factor graph gives "the number of frozen variables-FV" of the factor graph. The number of frozen variables FV is greater than or equal to $N - K$. Unlike the number of frozen bits at the input, FV depends on the stage order of the factor graph and it can vary from one FG to another.

For the (8, 4) polar (and RM) code, there exist 6 different FG's. Frozen variables of all FG's are shown in Figure 4.1 (where Stage 3 has adders for the most distant bits, and Stage 1 has adders for adjacent bits). One can observe that the number of frozen variables is FV $= 6$ and it doesn't change with changing FG for the (8, 4) polar (and RM) code.

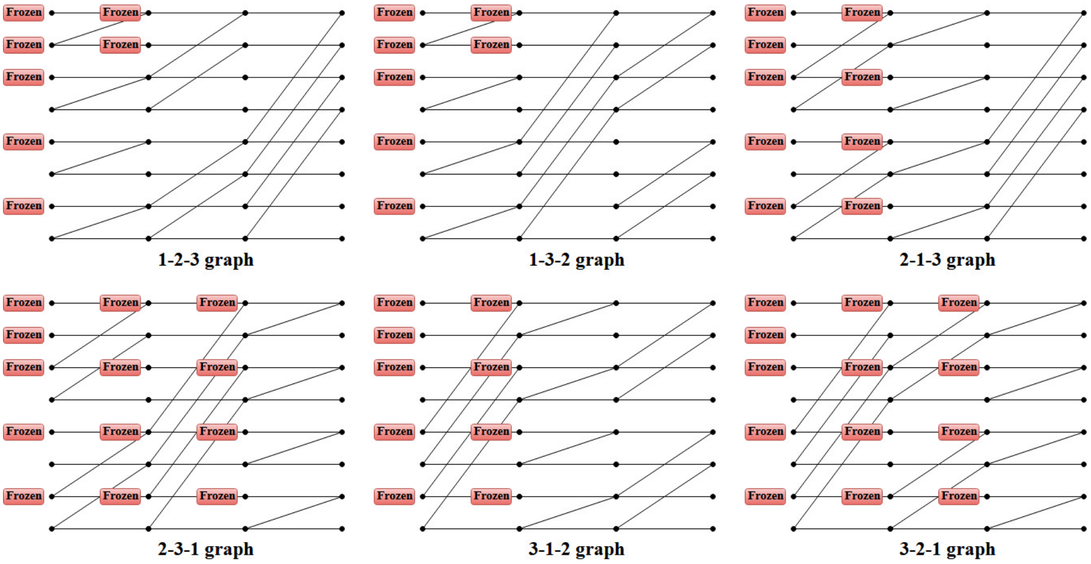


1-2-3 graph 1-3-2 graph 2-1-3 graph

2-3-1 graph 3-1-2 graph 3-2-1 graph

Figure 4.1: Frozen variables of all factor graphs for the (8,4) polar (and RM) code.

On the other hand, if we examine the FV's of a (8, 3) polar code (notice that there is no RM code with parameters (8, 3)), we can see a change from one FG to the other. For instance, FV= 7 for 1-2-3 and 1-3-2 FG's in Figure 4.2, whereas FV= 13 for 2-3-1 and 3-2-1 FG's.



1-2-3 graph 1-3-2 graph 2-1-3 graph

2-3-1 graph 3-1-2 graph 3-2-1 graph

Figure 4.2: Frozen variables of all factor graphs for the (8,3) polar code.

To further illustrate the change of FV with changing factor graph, we have increased the blocklength N to 16. In Figure 4.3, all frozen variables are depicted for the 4-3-2-

1 and 1-2-3-4 factor graphs of a (16, 8) polar code (again, notice that there is no RM code with parameters (16, 8)). Although 8 input frozen bits are the same for 1-2-3-4 and 4-3-2-1 FG's, starting from the second stage, the number and positions of the frozen variables differ such that 4-3-2-1 factor graph has FV=18 and 1-2-3-4 factor graph has FV=10.

Factor graphs of an (N, K) polar code can be divided into sets according to their FV's, as well as their CS values. We will call each set consisting of FG's with the same FV value, an "equi-FV set".

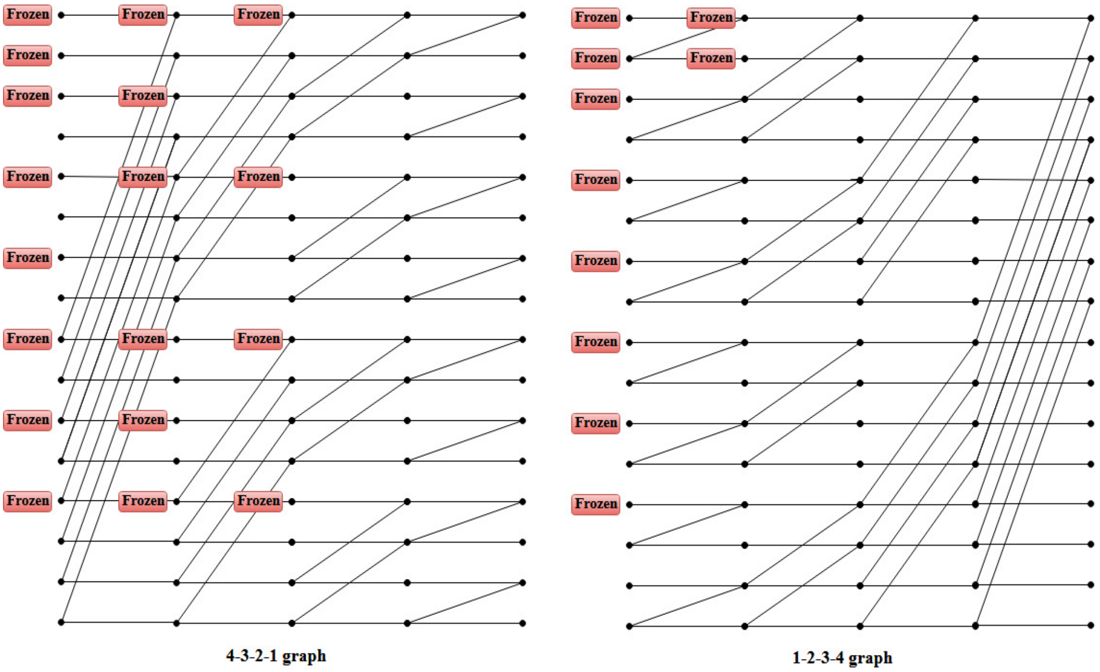


Figure 4.3: Frozen variables of 4-3-2-1 (FV=18) and 1-2-3-4 (FV=10) FG's for the (16,8) polar code.

For the (16, 8) adaptive polar code over BEC(0.35), distribution of equi-CS and equi-FV sets are given in Table 4.3. Equi-CS sets contain 6 FG's and they are the subsets of equi-FV sets. So, one can observe more than one equi-CS sets having the same value of FV. For instance, in Table 4.3, the first and second equi-CS sets have the same FV = 18.

Table 4.3: Distribution of equi-CS and equi-FV sets for the (16, 8) polar code over BEC(0.35).

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	18	7.66	6	2
		7.41	6	
2	14	7.21	6	1
3	10	7.05	6	1
3 Equi-FV sets	← **Total** →		24 FG’s	4 Equi-CS sets

Similarly, one can form 120 different factor graphs for the (32, 16) polar code over BEC(0.35), which can be partitioned into 10 "equi-CS sets", each having 12 FG’s with equal CS values. Distribution of equi-FV and equi-CS sets among 120 factor graphs for the (32, 16) adaptive polar code over BEC(0.35) is given in Table 4.4.

One observes 5 different FV values for the (32, 16) adaptive polar code over BEC(0.35). Again, notice that an equi-FV set may contain more than one equi-CS sets; i.e., equi-CS sets are subsets of equi-FV sets. CS values of the (32, 16) adaptive polar code given in Table 4.4 are sketched in Figure 4.4 versus FV.

Table 4.4: Distribution of equi-CS and equi-FV sets for the (32, 16) polar code over BEC(0.35).

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	14.70	12	1
5 Equi-FV sets	← **Total** →		120 FG’s	10 Equi-CS sets

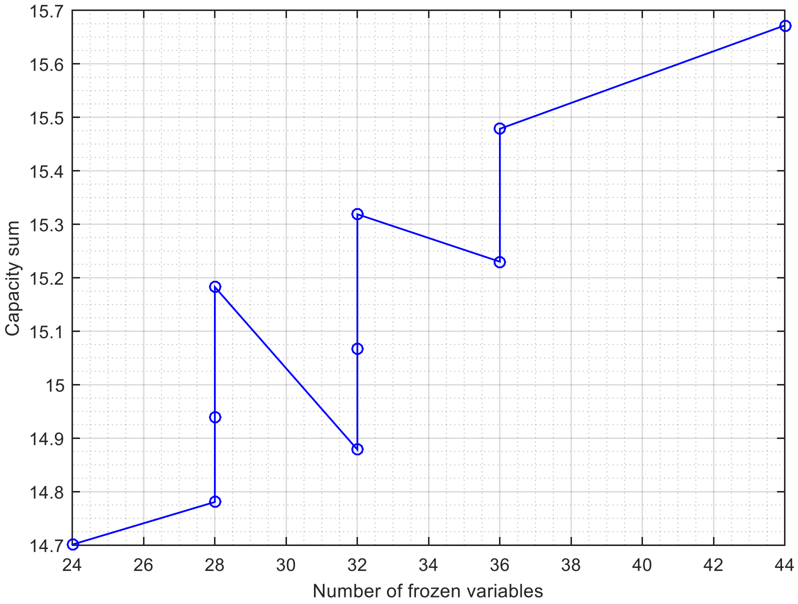


Figure 4.4: CS versus FV values of the (32, 16) adaptive polar code over BEC(0.35).

4.3 Capacity Sum (CS) and Number of the Frozen Variables (FV) for Reed-Muller Codes

For $N = 2^n$, $N \times N$ matrix G_N was defined in Chapter 2 as $G_N = F^{\otimes n}$ where $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. The generator matrices of both polar and Reed-Muller (RM) codes are obtained by suitably selecting rows from G_N. The r'th order RM code RM (r, n) is the linear code with generator matrix G_{RM}, which is obtained by selecting rows of G_N with Hamming weights $\geq 2^{n-r}$. The blocklength of the code is $N = 2^n$, with the information word length $K = \sum_{i=0}^{r} \binom{n}{i}$.

As explained in Section 2.3, the generator matrix G_P of an (N, K) polar code is constructed by choosing K rows of the matrix G_N, whose indices correspond to bit-channels with the least possible Bhattacharya parameters and the maximum channel capacities over a BEC for the reference FG [Arıkan, 2009].

Similarly to the polar codes, one can use one of the $n!$ factor graphs in the encoder and BP decoder of (N, K) RM codes. The capacity sum (CS) and the number of the frozen variables (FV) are computed likewise; CS equals to the sum of the capacities of the K transmission paths and total number of frozen variables in the factor graph corresponds to FV.

For a block length $N = 8$, selecting the rows of transformation matrix G_8 according to maximum Hamming weights coincide with choosing the codewords from G_N by minimum Bhattacharya parameters. Therefore, generator matrices of the (8, 4) polar and RM codes are the same. Difference between the polar and RM codes start to be observed for $N = 32$. Generator matrices of the (32, 16) RM and polar codes are given in Table 4.5; where the two different codewords on top of the generator matrices are colored in red. For the (32, 16) polar code, a specific 4-weight codeword is selected from $G_N = F^{\otimes 5}$ instead of a specific 8-weight codeword of the RM code.

Table 4.5: Generator matrices of the (32, 16) RM and polar codes

Generator matrix of the (32,16) RM code	**Row Weights**
10001000100010001000100010001000	**8**
11111111000000000000000000000000	8
11110000111100000000000000000000	8
11001100110011000000000000000000	8
10101010101010100000000000000000	8
11110000000000001111000000000000	8
11001100000000001100110000000000	8
10101010000000001010101000000000	8
11000000110000001100000011000000	8
10100000101000001010000010100000	8
11111111111111110000000000000000	16
11111111000000001111111100000000	16
11110000111100001111000011110000	16
11001100110011001100110011001100	16
10101010101010101010101010101010	16
11111111111111111111111111111111	32

Generator matrix of the (32,16) polar code	**Row Weights**
11110000000000000000000000000000	**4**
11111111000000000000000000000000	8
11110000111100000000000000000000	8
11001100110011000000000000000000	8
10101010101010100000000000000000	8
11110000000000001111000000000000	8
11001100000000001100110000000000	8
10101010000000001010101000000000	8
11000000110000001100000011000000	8
10100000101000001010000010100000	8
11111111111111110000000000000000	16
11111111000000001111111100000000	16
11110000111100001111000011110000	16
11001100110011001100110011001100	16
10101010101010101010101010101010	16
11111111111111111111111111111111	32

For the (32, 16) adaptive polar code, distribution of the equi-CS and equi-FV sets were given in Table 4.4 of the previous section, where 120 factor graphs are grouped into 10 different equi-CS and 5 distinct equi-FV sets. On the other hand, for the

(32,16) RM code, CS = 15.58 and FV = 30 values remain the same for all of the 120 FG's as shown in Table 4.6. Order of channel capacities may vary with changing stage order, but total capacity of the selected 16 channels remains the same. Input frozen bit positions are identical for all FG's of the polar (or RM) codes. However, although positions of the inner frozen variables can change from one FG to another, their total number remains the same for the (32, 16) RM code (Table 4.6); and does not change as in the case of the (32, 16) adaptive polar code (see Table 4.4).

Table 4.6: Distribution of equi-CS and equi-FV sets for the (32, 16) RM code.

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	30	15.58	120	1
1 Equi-FV set	← **Total** →		120 FG's	1 Equi-CS set

We perform some simulations to understand the effects of FV and CS of the chosen factor graph on the performance of BP decoders. Factor graphs, which have the highest CS values, are in accordance with the design philosophy of polar codes that chooses the highest-capacity paths. We expect that BP decoders using these FG's to perform better and faster, since they also have the highest number of frozen variables. In the following sections, BP decoding performances will be evaluated and the effects of FV and CS values of the factor graphs to be used in BP decoders will be observed.

4.4 Performance of Single-FG versus Multiple-FG Decoders in the Maximum Equi-FV Set of Polar Codes

We call the BP decoder that uses a specific factor graph, a "single-FG (or 1-FG) decoder". On the other hand, a "multiple-FG (or M-FG) decoder" employs a pre-chosen set of $M > 1$ factor graphs, where all FG's within the set are used successively to decode an erased word, until all erasures are filled or the M^{th} (i.e., the last) FG is used.

In our simulations, each bit of a generated codeword is erased randomly, with a given erasure probability ϵ. Erased codewords are decoded by using single-FG or M-FG BP decoders and the performance is measured by the number of remaining undecoded words.

In this section, we present some single-FG and M-FG simulation results for the BP decoding performance of the (64, 32) and (128, 64) polar codes, where the M-FG decoder attempts to rationally choose M factor graphs among $n!$ different permutations (720 choices for $N = 64$ and 5040 choices for $N = 128$).

In the BP decoding algorithm, we observe that after certain number of iterations, the number of remaining undecoded words does not change. Required number of BP iterations for the polar codes over a BEC(ϵ) with three different block lengths ($N =$ 32, 128, 512) has been examined in [Doğan, 2015]. He evaluated BP decoding performances for the erasure probabilities up to $\epsilon = 0.5$, with using "1-2-...-n" and "n-…-2-1" factor graphs and he determined the minimum sufficient number of iterations. Table 4.7 presents the required number of iterations for the (32, 16), (128, 64) and (512, 256) polar codes over BEC(0.35), according to [Doğan, 2015].

Table 4.7: Minimum sufficient number of round iterations for the BP decoding of rate 1/2 polar codes over BEC(0.35) with using "1-2-...-n" and "n-…-2-1" factor graphs.

Factor Graph	**(32, 16) polar code**	**(128, 64) polar code**	**(512, 256) polar code**
1-2-...-n	6	13	20
n-…-2-1	5	8	9

In our simulations, we utilized high-FV and high-CS valued FG's; hence, 1-2-...-n factor graph that requires relatively higher number of iterations was not in our FG set. So, for the BP decoding of the (64, 32) and (128, 64) codes over BEC(0.35), we have employed 15 round iterations, which are more than sufficient.

4.4.1 (64, 32) Adaptive Polar Code

Performance dependence of the 1-FG decoder on the number of frozen variables has been evaluated in [Doğan, 2015], where he presented results showing that the performance of the BP decoder increases while the number of frozen variables increases.

For the (64, 32) polar code, $6! = 720$ different factor graphs exist. There are 180 different CS values for the erasure rate $\epsilon = 0.35$, and 720 FG's are grouped into 10 equi-FV sets, containing 48, 84 or 96 FG's as shown in Table 4.8. The number of equi-CS sets is much larger, there are 180 different CS values for $\epsilon = 0.35$, where each equi-CS set is a subset of an equi-FV set, contains 4 FG's and distributed in 720 FG's as displayed in Table 4.8.

Table 4.8: Distribution of the number of frozen variables (FV) for the erasure rate $\epsilon = 0.35$ among 720 factor graphs of the (64, 32) adaptive polar code.

Equi-FV set no.	**FV**	**Number of FG's in the equi-CS set**	**Number of equi-CS sets in the equi-FV set**	**Maximum and Minimum CS Values**
1	88	48	12	31.6594, 30.4173
2	84	84	21	31.5661, 30.0330
3	80	48	12	31.4439, 29.6940
4	76	96	24	31.1861, 29.5418
5	72	84	21	31.0339, 29.4336
6	68	84	21	30.6115, 29.0113
7	64	96	24	30.5033, 28.8591
8	60	48	12	30.3511, 28.6012
9	56	84	21	30.0121, 28.4791
10	52	48	12	29.6279, 28.3858
10 Equi-FV sets	← **Total** →	720 FG's	180 Equi-CS sets	

To observe the effect of FV on the BP decoder, 10 FG's with different FV's are selected from each equi-FV set and their 1-FG decoding performances are evaluated. 1000 codewords are generated over BEC(0.35) and the number of undecoded words for each FV are plotted in Figure 4.5.

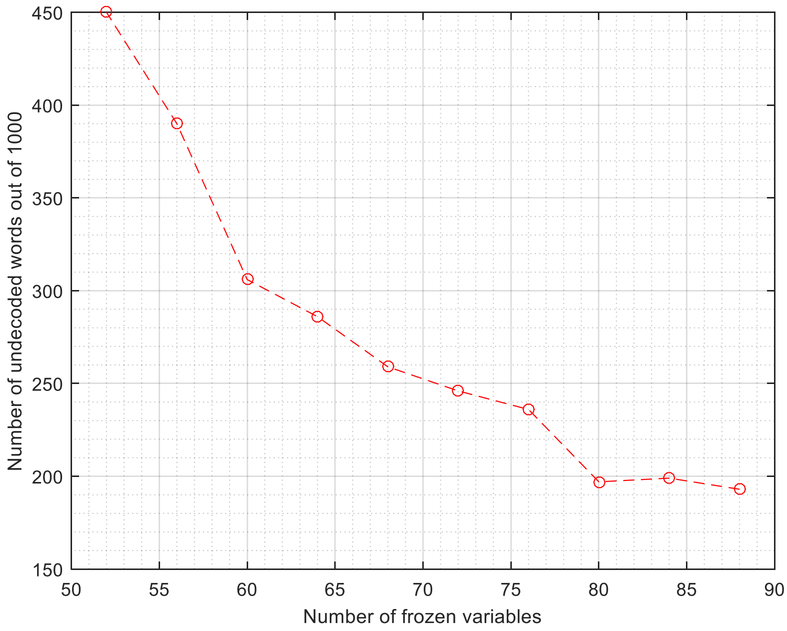


Figure 4.5: Single-FG BP performance versus FV for the (64, 32) adaptive polar code over BEC(0.35).

Figure 4.5 shows that with increasing number of frozen variables, remaining number of undecoded words decrease substantially. One observes that the factor graphs with the highest FV value give the best single-FG BP decoding performance. Performance advantage of the highest FV decoder (FV $=$ 88) over the lowest FV decoder (FV $=$ 52) is almost 26% $((450-190)/1000$ words).

For the (64, 32) adaptive polar code over BEC(0.35), the maximum FV value is 88 and there are 48 factor graphs that have this highest FV, as shown in Table 4.8. In the next experiment, we have used these 48 factor graphs of the best equi-FV set to evaluate the M-FG decoding performances for $M=4k$, and $k=1,2,\ldots,12$.

Initially, we rank the 48 FG's that have the maximum FV's in decreasing order of CS values. M-FG decoder employs the first M factor graphs from this set. The experiment is repeated for 20 trials (20 different erasure patterns) over BEC(0.35), where each pattern contains 1000 codewords. In each trial, the number of undecoded words out of 1000 erased words is found (firstly by the 1-FG decoder and then) by the 4-FG, 8-FG, … decoders, each time increasing the number of FG's, M, in steps of 4 until all FG's in the equi-FV set are exhausted. Resulting performance graphs for 20 trials are given in Figure 4.6.

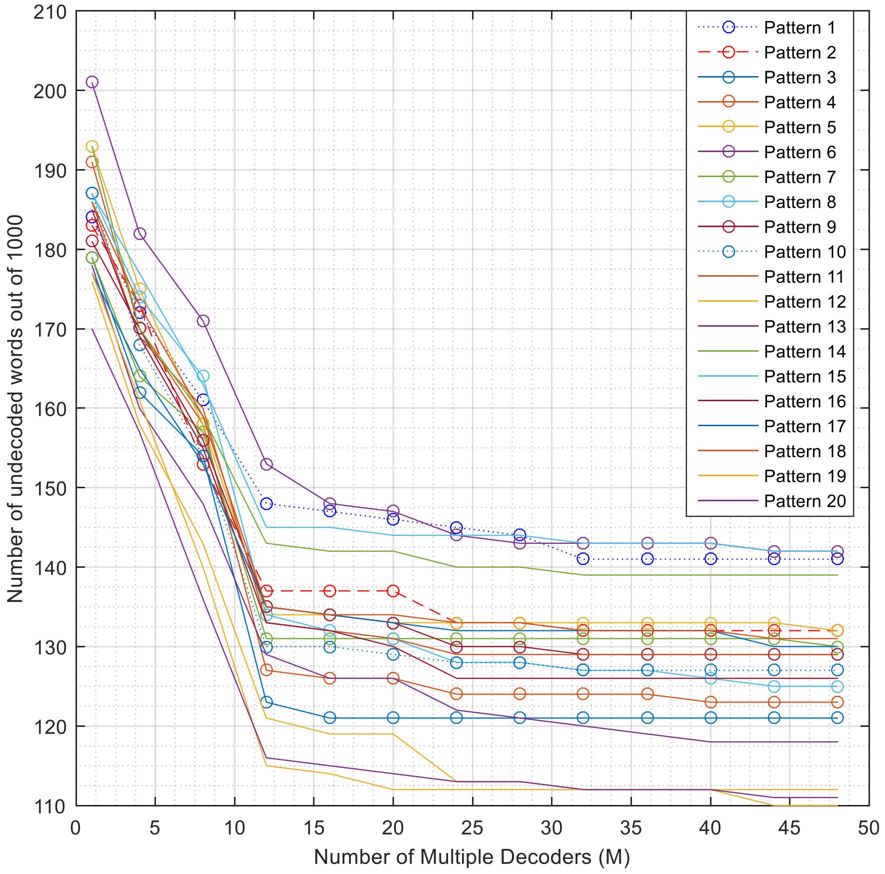


Figure 4.6: Performance of M-FG decoders in 20 trials for the (64,32) adaptive polar code over BEC(0.35) for $M = 1, 4, 8, 12, \dots, 48$, where the first M factor graphs ranked in decreasing CS order are used among the 48 maximum FV graphs (FV $= 88$).

From Figure 4.6, one observes that using all 48 factor graphs, in the best case (the lowest curve) there remains 110 undecoded words out of 1000 erased words and in the worst case, the number of undecoded words is 142. So, we observe a 3.2% change in performance depending on the specific erasure pattern of these 20 trials.

On the other hand, performance of the 48-FG decoder seems ≈6% superior to that of the 1-FG decoder in all trials, because the average number of undecoded words out of 1000 reduce from ≈185 to ≈127. Similarly, performance advantage of the 12-FG decoder over the 1-FG decoder is around ≈5% in all 20 trials. Hence, in spite of the higher complexity with respect to 1-FG case, if one ventures on employing the M-FG decoder instead of 1-FG, it seems sufficient to choose $M = 12$ rather than $M = 48$, since the 1% performance advantage of the 48-FG decoder over 12-FG is not significant to account for the 4-times-increased decoder complexity, $O(MN \log N)$.

4.4.2 (128, 64) Adaptive Polar Code

In this part, we evaluate the BP decoding performance of 1-FG and M-FG decoders for the (128, 64) adaptive polar code over BEC(0.35), and compare it with the performance found for the (64, 32) adaptive polar code.

Number of stages for $N = 128$ is $n = 7$, so the encoder and decoder of the (128, 64) polar code can be implemented by $n! = 7!$ different FG's. Distribution of the number of frozen variables among $7! = 5040$ factor graphs and FG's that have the highest FV are given in Appendix A. For the (128, 64) polar code, there are 28 different FV values. We randomly pick one FG from each equi-FV set to evaluate the single-FG decoding performance. As in the previous part, 1000 codewords are generated over BEC(0.35) and the number of undecoded words out of 1000 erased words are sketched for the selected FG's in Figure 4.7. Single-FG decoding performances for the (128, 64) polar code show that remaining number of undecoded words significantly decreases while the FV value of the corresponding FG increases, as in the case of the (64, 32) polar code. Performance advantage of the highest FV decoder

(FV = 210) over the lowest FV decoder (FV = 98) is approximately 58% as opposed to the 26% of the (64, 32) polar code; hence, employment of FG's with the highest FV in BP decoding is supported strongly.

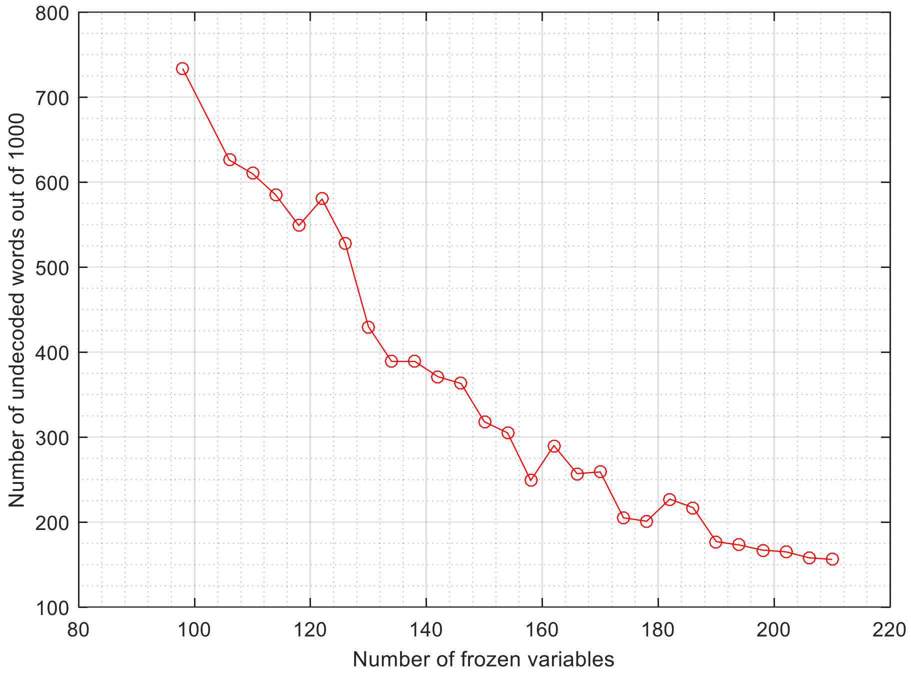


Figure 4.7: Single-FG BP performance versus FV for the (128, 64) adaptive polar code over BEC(0.35).

For the erasure rate $\epsilon = 0.35$, the highest FV value is 210 for the (128, 64) adaptive polar code and number of FG's with FV= 210 is 72. For evaluating M-FG decoding performances, 72 factor graphs within this highest equi-FV set are employed. 72 FG's are grouped into 18 equi-CS sets over BEC(0.35), where the same CS value is shared by 4 different FG's.

72 factor graphs are ordered in decreasing order of CS values and M-FG decoder performance of the first M-FG's over BEC(0.35) for $M = 4k$, $k = 1, \ldots, 18$ is found. Simulations are repeated for 10 different sets of 1000 erased codewords and performance in terms of the number of undecoded words is depicted in Figure 4.8.

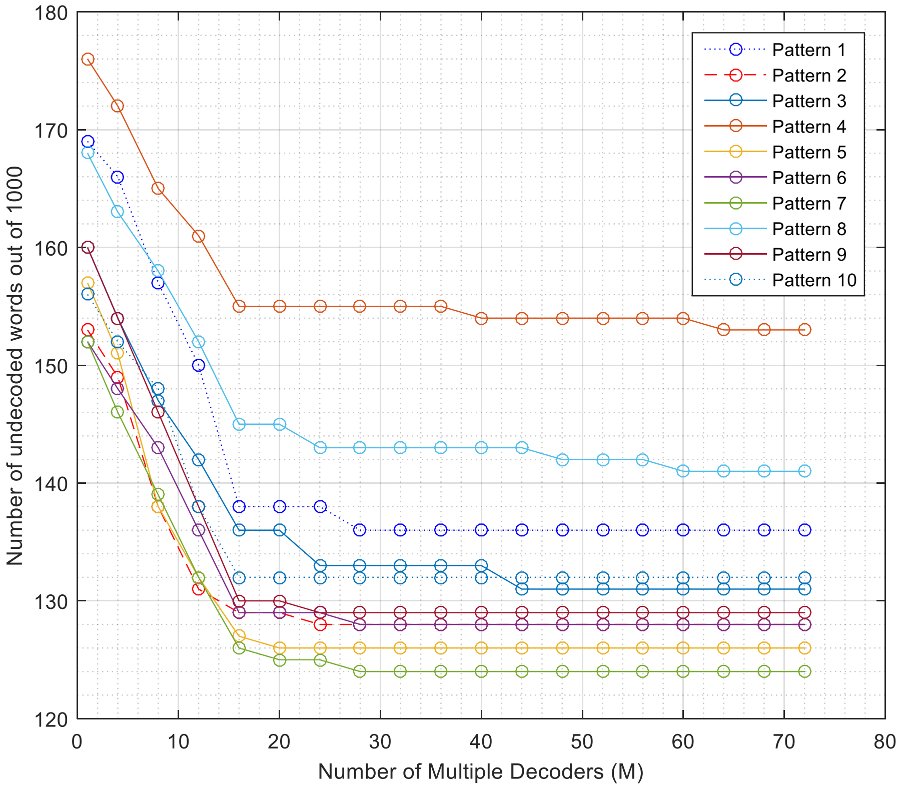


Figure 4.8: Performance of M-FG decoders in 10 trials for the (128,64) polar code over BEC(0.35) and for $M = 1, 4, 8, \ldots, 72$, where the first M factor graphs ranked in decreasing CS order are used among the 72 maximum-FV graphs (FV = 210).

From Figure 4.8, one observes that using all 72 factor graphs, there remains 124 undecoded words out of 1000 erased words in the best case (lowest curve) and 153 in the worst case. So, there is already a 2.9% change in performance depending upon the specific erasure pattern of these 10 trials.

By multiple-FG decoding using all 72 factor graphs, average of 10 trials show a reduction in the remaining number of undecoded words from 160 to only 133 among 1000 erased codewords. Figure 4.8 also shows that the performance does not improve significantly for $M > 16$. There is less than 3% performance improvement of the 16-FG decoder over the single-FG BP decoder; which doesn't seem sufficient to justify the use of 16-FG decoder instead of 1-FG.

For the (64, 32) and (128, 64) adaptive polar codes over BEC(0.35), average BP decoding performances are summarized in Table 4.9.

Table 4.9: BP decoding performances of (64, 32) and (128, 64) adaptive polar codes over BEC(0.35) by using maximum equi-FV sets (average results out of 1000 erased codewords over 20 trials for the (64, 32) code and 10 trials for the (128, 64) code)

Set Choice	(64, 32) adaptive polar code		(128, 64) adaptive polar code	
	Single-RFG decoder	12-FG decoder using 12 highest CS valued FG's in the max. equi-FV set	Single-RFG Decoder	16-FG decoder using 16 highest CS valued FG's in the max. equi-FV set
Number of undecoded words out of 1000	185	133	160	135

It seems that increasing the number M of FG’s in M-FG decoders does not result in significant performance improvement over 1-FG’s in case of the polar codes. In Section 4.6, we restrict the set of FG’s in M-FG decoders to 4-element equi-CS sets and investigate the performance for $N = 64$. Before that, we repeat the above experiments for the M-FG decoding of RM codes in Section 4.5.

4.5 Performance of Single-FG versus Multiple-FG Decoders for the Reed-Muller Codes

RM codes can also be decoded by BP decoders as polar codes. In this section, single-FG and multiple-FG BP decoding performances of RM codes are evaluated, and compared with the performance of polar codes.

It was mentioned in Section 4.3 that if active input bits (hence the remaining frozen bits) are chosen so as to generate the RM codes, all $n!$ different factor graphs have the same FV and CS values. For the (128, 64) RM code, corresponding capacity sum, CS is 62.15; and the number of frozen variables, FV equals 140. Because of the constant values of CS and FV, we expect that single-FG BP decoding performance of RM codes to be similar for all FG's. As an example, 20 factor graphs, 10 from the maximum equi-FV set (FV=210) and 10 from the minimum equi-FV set (FV=98) of the (128, 64) adaptive polar code, are used for decoding. Table 4.10 gives the performance of 20 single-FG decoders for the (128, 64) RM code over BEC(0.35) in terms of the number of undecoded words out of 1000. As expected, 1-FG decoding performances of the selected 20 FG's are very similar to each other.

Table 4.10: Single-FG BP decoding performances of 20 FG's for the (128,64) RM code over BEC(0.35).

Factor Graph	Number of Undecoded words
7654321	573
7654231	576
7653421	558
7645321	565
7645312	565
7645231	566
7643521	551
6754321	573
6754312	575
6754231	576
1254376	551
1254367	546
1253476	550
1253467	542
1245376	563
1243576	574
1243567	571
1235476	557
1235467	550
1234576	578

For evaluating the M-FG BP decoding performance of the (128, 64) RM code in similar conditions to that of the (128, 64) adaptive polar code over BEC(0.35), 10 different sets of 1000 erased words over BEC(0.35) are generated. For each set of

1000 erased words, M-FG BP decoding is performed by using the 72 FG's within the maximum equi-FV set of the (128, 64) adaptive polar code over BEC(0.35). The number of undecoded words at the M-FG decoder output is found for $M = 1$ and $M = 4k$, where $k = 1, \dots, 18$ and sketched in Figure 4.9 for 10 trials corresponding to 10 different erasure patterns.

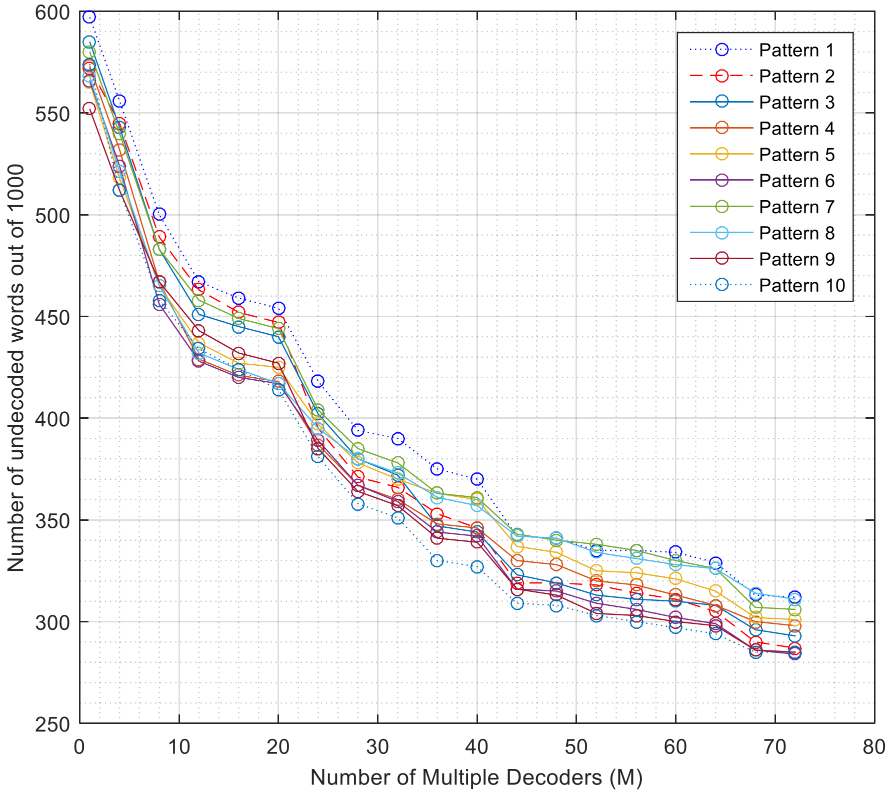


Figure 4.9: Performance of M-FG decoders in 10 trials for the (128,64) Reed-Muller codes over BEC(0.35), for $M = 1, 4, 8, \dots, 72$, using the 72 FG's in the maximum equi-FV set of the (128, 64) adaptive polar codes.

Considering the average of 10 patterns in Figure 4.9, the number of undecoded words at the M-FG decoder output of the (128, 64) RM code reduces almost by 28% as M increases from 1 to 72. However, although the M-FG BP decoding performance of the RM code improves considerably with increasing M, it is still far from the 1-FG performance of the adaptive polar code with similar parameters.

To have a fair comparison, remembering that the selected 72 FG's of this experiment have FV = 210 for the frozen-bit-choice of the (128, 64) adaptive polar code, but FV = 140 for the frozen-bit-choice of the (128, 64) RM code; we repeat the polar code experiment within smaller-valued equi-FV sets.

We select two equi-FV sets (with FV = 142 and FV = 138) of the (128, 64) polar code over BEC(0.35), and find the M-FG decoder performances for $M = 1$ and $M = 4k$, where $k = 1, \dots, 18$. In each equi-FV set, 72 highest-CS valued FG's are ranked in the order of decreasing CS, and the M-FG decoding performances of (128, 64) polar codes over BEC(0.35) are evaluated in terms of the undecoded words out of 1000 erased words. Average performance over 10 trials is given in Figure 4.10 for the RM decoder (with FV = 140) and polar decoders in 3 equi-FV sets (FV = 138, 142, 210).

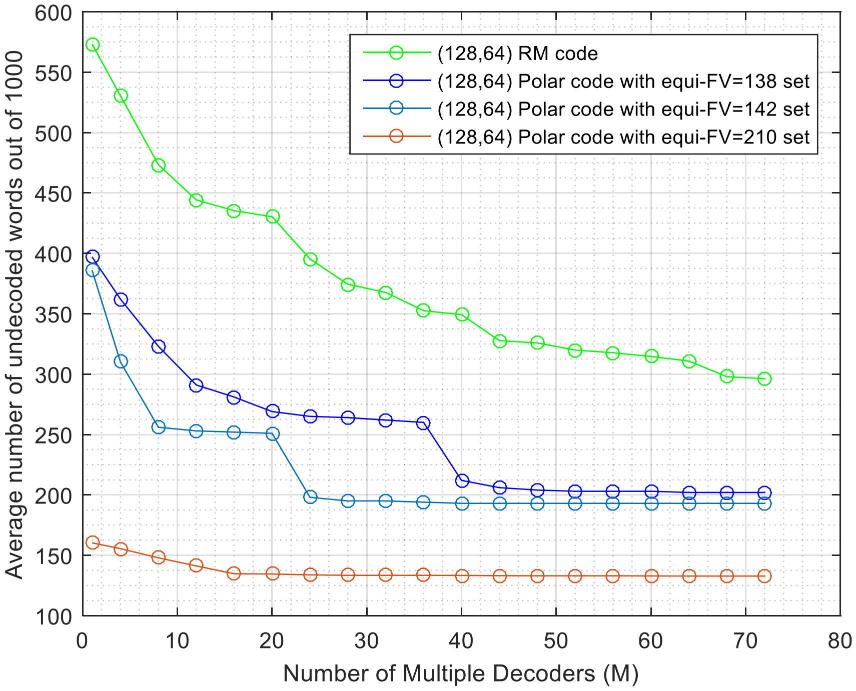


Figure 4.10: Average performance of M-FG BP decoders over a BEC(0.35) in 10 trials for the (128,64) Reed-Muller codes and (128,64) adaptive polar codes within equi-FV sets (with FV=210, 142, 138).

One observes from Figure 4.10 that BP decoding performance of the (128, 64) polar code is significantly better than that of the RM code, even when one restricts the decoder FG's to the small-valued equi-FV sets (FV = 138 and FV = 142) that have similar FV to that of the (128, 64) RM code (FV = 140).

Single-FG BP decoding performance of the (128, 64) RM code seems approximately 18% worse than the performances of polar codes with similar FV, and 42% worse than that of the polar code with the maximum FV; whereas (say) 12-FG decoding performance of the (128, 64) RM code is almost 18% worse than the performances of polar codes with similar FV; and 30% worse than that of the polar code within the maximum equi-FV set.

M-FG BP decoding performance of (128, 64) RM codes improves significantly as M increases from 1 to 72. It is of interest whether M-FG BP decoding performances of the RM codes can be as good as BP decoding performances of the polar code. To answer this, we increase M up to 152 and performances of M-FG decoders for (128,64) RM code and (128,64) polar code are given in Figure 4.11 The results are obtained for one pattern, which contains 1000 codewords, over BEC(0.35).

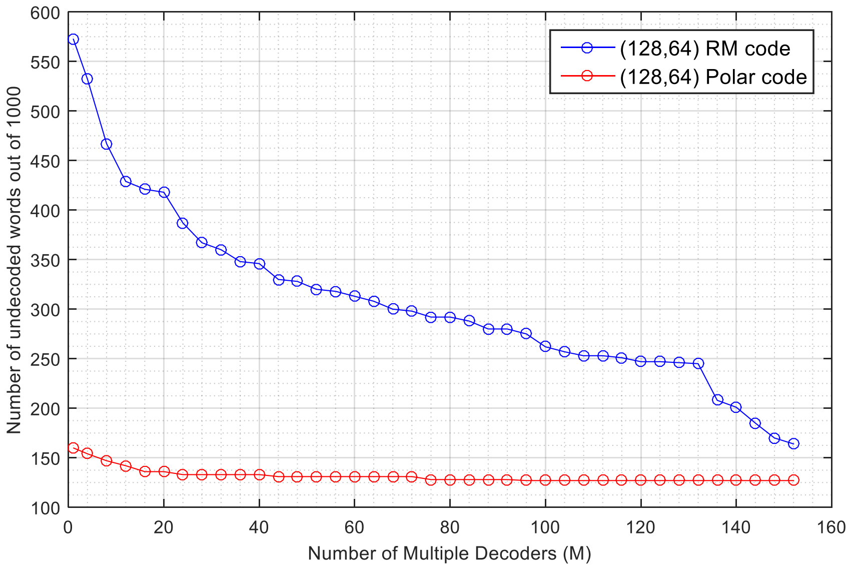


Figure 4.11: Performance of M-FG decoders for the (128,64) Reed-Muller codes and (128,64) polar codes over BEC(0.35), for $M = 1, 4, 8, \ldots, 152$

From Figure 4.11, one observes that the number of undecoded words at the M-FG decoder output of the (128, 64) RM code decreases nearly by 41% as M increases and 152-FG decoding performance of the RM code is approximately the same as the 1-FG performance of the adaptive polar code with similar parameters. So, multiple FG-BP decoding performances of RM codes can be good as polar codes with using multiple factor graphs, but it has much higher complexity compared to the polar codes.

4.6 BP Decoding Performances of Equi-CS Sets

In Section 4.4, single-FG and multi-FG BP decoding performances in the best equi-FV set have been evaluated for the (64, 32) adaptive polar code over BEC(0.35) and it was observed that M-FG decoding with large values of M doesn't provide any noticeable advantage. One may still be interested in exploring the performance of M-FG BP decoders with small values of M. In this section, we compare the performance of 4-FG decoders to those of the (best and worst) 1-FG decoders within the same set.

For the (64, 32) adaptive polar code over BEC(0.35), distribution of FV values are given in Table 4.8 of Section 4.4.1, where one observes that each equi-FV set can be divided into smaller sets with equal CS values. 48 FG's with the maximum FV ($=$ 88) are grouped into 12 equi-CS sets, and each equi-CS set contains the specific 4 FG's shown in Table 4.11.

In this section, we investigate the single and 4-FG BP decoding performances in these maximum-FV and equi-CS sets, that are composed of four FG's.

Table 4.11: Factor graphs in all equi-CS sets of the first equi-FV set of the (64, 32) adaptive polar code designed over BEC(0.35).

Equi-CS set no.	**Factor graph**	**FV**	**CS**
1	654321, 654231, 645321, 645231	88	31.6594
2	653421, 652431, 643521, 642531	88	31.5098
3	654312, 654213, 645312, 645213	88	31.4286
4	653241, 652341, 643251, 642351	88	31.2945
5	653412, 652413, 643512, 642513	88	31.2641
6	654132, 654123, 645132, 645123	88	31.2133
7	651432, 651423, 641532, 641523	88	30.8992
8	653214, 652314, 643215, 642315	88	30.8180
9	653142, 652143, 643152, 642153	88	30.8126
10	651342, 651243, 641352, 641253	88	30.6630
11	653124, 652134, 643125, 642135	88	30.5818
12	651324, 651234, 641325, 641235	88	30.4173

Within each equi-CS set, we evaluate three performance figures: The number of undecoded words by *i*) the best 1-FG, *ii*) the worst 1-FG and *iii*) 4-FG BP decoders. Table 4.12 shows the average performance over 20 different erasure patterns in each equi-CS set. “Best 1-FG” and “Worst 1-FG” columns show the best and the worst single-FG decoding performances of the factor graphs in each set. 4-FG decoder gives the multiple BP decoding performance of 4 factor graphs. Each erasure pattern consists of 1000 erased codewords and the average BP decoding performances over 20 patterns are shown in Table 4.12, in terms of the number of undecoded words among 1000 erased codewords for the BEC(0.35).

Table 4.12: Average BP decoding performance over 20 erasure patterns of the best, worst and 4-FG decoders of equi-CS sets for the (64, 32) code over BEC(0.35) (average results out of 1000 erased codewords over 20 trials).

Equi-CS set no.	CS	FV	Best 1-FG in the equi-CS set	Worst 1-FG in the equi-CS set	4-FG decoder in the equi-CS set
1	31.66	88	181.75	185.90	168.45
2	31.51	88	184.70	193.10	155.65
3	31.43	88	182.95	190.05	148.40
4	31.29	88	190.75	199.15	159.50
5	31.26	88	186.50	198.95	**136.50**
6	31.21	88	207.25	213.75	176.80
7	30.90	88	243.60	254.15	189.75
8	30.82	88	205.80	216.85	153.55
9	30.81	88	215.60	231.55	140.15
10	30.66	88	251.20	268.55	161.95
11	30.58	88	226.05	240.20	144.00
12	30.42	88	262.40	276.85	170.10

Factor graphs, which have the highest CS values, suit the philosophy of a polar code that chooses the highest-capacity paths. We expect that the BP decoders using these factor graphs to perform better and faster. In Table 4.12, the best single BP decoding performance is obtained by a factor graph in the equi-CS set #1 (182 undecoded words out of 1000 erased words). So, the factor graph with the highest CS and FV values gives the best single BP decoding performance as expected.

On the other hand, if we look at the multiple-BP decoding performances, one can observe that the best 4-FG BP decoding performance belongs to the equi-CS set #5 (136 undecoded words out of 1000 erased words) with ≈5% performance improvement over the best single-FG BP decoder. Startlingly, 4-FG BP decoding performance of the highest equi-CS set is not as good as that of the equi-CS set #5.

This experiment is repeated for different erasure rates to determine whether 4-FG BP decoding performance of the equi-CS set #5 is the best. Equi-CS sets that we obtain for ϵ =0.35 are not the same for all erasure rates. We have observed that each CS value is shared by 12 different factor graphs from $\epsilon = 0.1$ to 0.25. Starting from

ϵ $= 0.3$ up to 0.5, factor graphs are divided into equi-CS sets with 4 graphs and FG's in each equi-CS set are identical to the equi-CS sets for ϵ $= 0.35$, but the order of equi-CS sets, that is given in Table 4.11, is not the same for all erasure rates. Average 4-FG decoding performances of 12 equi-CS sets over 20 erasure patterns are given in Table 4.13 for 5 different erasure rates; $\epsilon =$ 0.30, 0.35, 0.40, 0.45 and 0.50. Equi-CS set numbers are fixed for all erasure rates according to order of equi-CS sets for $\epsilon =$ 0.35. Surprisingly, average 4-FG BP decoding performance of the equi-CS set #5 is the best for all erasure rates.

Table 4.13: Average BP decoding performance over 20 erasure patterns of the 4-FG decoders of equi-CS sets for the (64, 32) code over 5 different erasure rates. (average results out of 1000 erased codewords over 20 trials).

Equi-CS set no. as to ϵ = 0.35	**4-FG decoder for ϵ = 0.30**	**4-FG decoder for ϵ = 0.35**	**4-FG decoder for ϵ = 0.40**	**4-FG decoder for ϵ = 0.45**	**4-FG decoder for ϵ = 0.50**
1	65.10	168.45	376.10	629.05	843.15
2	58.85	155.65	357.70	609.20	824.25
3	54.15	148.40	346.50	604.70	819.10
4	59.85	159.50	363.25	617.10	836.05
5	**48.20**	**136.50**	**330.85**	**586.25**	**805.50**
6	66.85	176.80	390.60	645.55	842.70
7	73.80	189.75	409.10	597.50	853.45
8	53.70	153.55	337.35	664.15	838.60
9	48.70	140.15	358.30	615.40	810.15
10	59.35	161.95	377.45	638.25	841.20
11	49.15	144.00	343.25	604.55	826.50
12	60.50	170.10	390.35	652.50	859.25

In Section 4.4.1, while using the maximum equi-FV set, we have decided that the multiple-FG decoder with 12 factor graphs is sufficient and there remains nearly 133 undecoded words out of 1000 erased words with 12-FG decoder. So, M-FG BP decoding performance of the best equi-CS set is approximately the same as the maximum equi-FV set for $N = 64$, with 3 times lower complexity. Performances of equi-CS sets and equi-FV sets for the (64, 32) adaptive polar code over BEC(0.35) are summarized in Table 4.14.

Table 4.14: Performances of some M-FG BP decoders for the (64, 32) adaptive polar code over BEC(0.35) by using maximum-FV-valued FG's (average results out of 1000 erased codewords over 20 trials).

Set Choice	48-FG decoder using all FG's in the equi-FV set #1	12-FG decoder using 12 FG's with the highest CS	4-FG decoder using the equi-CS set #1	4-FG decoder using the equi-CS set #5
Number of undecoded words out of 1000	127	133	168	136

We observe that, with increasing block length N, the number of the factor graphs in equi-CS sets decreases as shown in Table 4.15 for the (32, 16), (64, 32), (128, 64), (256, 128) and (512, 256) adaptive polar codes over BEC(0.35).

Table 4.15: Number of factor graphs in the maximum equi-FV set and in equi-CS sets from block length $N = 32$ to $N = 512$ for adaptive polar codes over BEC(0.35).

	(32,16) polar code	(64,32) polar code	(128,64) polar code	(256,128) polar code	(512,256) polar code
Number of FG's in the maximum equi-FV set	12	48	72	36	24
Number of FG's in equi-CS sets	12	4	4	2	1

Therefore, for the M-FG decoding of larger block length-codes, equi-CS sets are not meaningful. Instead, choosing a small number of highest-CS-valued FG's within the maximum equi-FV set seems reasonable.

4.7 Comparison of the Multi-FG BP Decoding Performance of the (64, 32) Adaptive Polar Code with Doğan's Work

In Doğan's work [Doğan, 2015], factor graphs with large CS and FV values are used for multi-FG BP decoding. He uses a pre-trained criteria, while choosing the best performing factor graphs. For a (64, 32) adaptive polar code over a BEC(0.34375), his genie-chosen 5 factor graphs are given in Table 4.16 (reproduced from [Doğan, 2015]). In his simulations, 1000 codewords are generated and the number of erasures is fixed to 22 for each word.

Table 4.16: 1-FG to 5-FG BP decoding performances (in terms of the number of undecoded words out of 1000 erased words) of Doğan's genie-chosen 5 factor graphs for the (64, 32) code with 22 erasures per codeword reproduced from [Doğan, 2015].

Factor Graph	FV	CS	Number of undecoded words	Number of remaining undecoded words
624531	84	31.388	100	100
643251	88	31.294	91	75
643512	88	31.264	87	53
645132	88	31.213	101	47
624513	84	31.112	102	46

In Table 4.16, the number of undecoded words column shows the single-FG BP decoding performances of each factor graph, whereas the last column shows the cumulative performance of the related row and preceding ones. By using all 5 FG's, number of undecoded words out of 1000 erased words is decreased up to 46.

Since the choice of such sets depending on previously recorded performance is not so practical, we prefer a more systematic selection, such as using the FG's with the highest number of frozen variables, ranked in decreasing order of CS values in M-FG decoders. We have already evaluated the M-FG decoding performances of the (64, 32) adaptive polar code over BEC(0.35) for such FG's, chosen with respect to the criterion of "the highest-FV followed by high-CS values", in Section 4.4.1.

To make a more detailed and fair comparison with Table 4.16, we generate 1000 erased words by erasing exactly 22 bits of each codeword of the (64, 32) code as in [Doğan, 2015]. Then, we perform 1-FG and *M*-FG BP decoding using the FG's of the maximum-FV set ranked in decreasing order of CS values. Corresponding BP decoding performances of the first 20 highest-CS-valued FG’s within the maximum equi-FV set are given in Table 4.17 in terms of the number of undecoded words.

Table 4.17: BP decoding performances (in terms of the number of undecoded words out of 1000 erased words) of the first 20 highest-CS-valued FG’s with maximum FV for the (64, 32) code with 22 erasures per codeword.

Factor Graph	**FV**	**CS**	**Number of undecoded words**	**Number of remaining undecoded words**
654321	88	31.699	90	90
654231	88	31.699	95	77
645231	88	31.699	95	77
645321	88	31.699	90	77
643521	88	31.560	94	73
642531	88	31.560	93	66
653421	88	31.560	98	66
652431	88	31.560	97	66
645213	88	31.485	92	52
645312	88	31.485	97	46
654312	88	31.485	97	46
654213	88	31.485	92	46
643251	88	31.355	98	46
642351	88	31.355	99	45
653241	88	31.355	104	45
652341	88	31.355	104	45
643512	88	31.331	104	45
642513	88	31.331	93	45
653412	88	31.331	106	45
652413	88	31.331	93	45

Best single BP decoding performance is obtained by FG’s with the highest CS and FV values (90 undecoded words out of 1000 erased words). In Table 4.17, last column shows that the remaining number of undecoded words does not change significantly for $M \geq 10$, and 10-FG performance is similar to that of Doğan’s genie-chosen set, which corresponds to ≈5% performance improvement over single-FG decoders.

CHAPTER 5

CONCLUSION

For a polar code with blocklength $N = 2^n$, $n!$ different factor graph representations can be constituted by changing the order of stages with respect to each other. In this thesis, we discuss whether the generator matrix G_P of the (N, K) polar code and the sum of capacities of the K active channels are invariant under these stage permutations. Invariance of G_P ensures that corresponding factor graphs generate the same linear code; whereas keeping the capacity-sum at its maximum value supports the polar code idea of activating the maximum-capacity channels, while freezing the low-capacity ones.

Transformation matrix G_N of the factor graph, which is the n-th Kronecker product $G_N = F^{\otimes n}$ of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, contains Z-type of connections with 2 inputs and 2 outputs. Naming the stages of a factor graph as 1, 2, …, n [Doğan 2015], where Stage 1 contains the smallest Z's with adjacent inputs and Stage n contains the largest Z's that connect input nodes separated by $2^{n-1} = N/2$; one may distinguish an FG with its related stage order. For instance, the "n-…2-1" factor graph is the reference FG, and its transformation matrix G_N can be decomposed into stage matrices $G_N^{(i)}$; such that $G_N = G_N^{(n)} \ldots G_N^{(2)} G_N^{(1)}$. In Chapter 3, we prove that stage matrix multiplication $G_N^{(i)} G_N^{(j)}$ is commutative; therefore, G_N is invariant under stage permutations. In other words, all $n!$ factor graphs have the same transformation matrix G_N.

Generator matrix G_P of an (N, K) polar code is constructed by selecting K rows of the transformation matrix G_N. Selected row indices correspond to the active information bits and the remaining $N - K$ bits are frozen. Hence, with the same frozen and active bits, the generator matrix G_P is invariant under stage permutations, and all $n!$ FG encoders generate the same linear code. However, corresponding CS and FV values may differ from one FG of a polar code to the other.

For the FG implementation of an (N, K) code, the "capacity sum-CS" is defined as the sum of the capacities of the K transmission paths [Doğan, 2015] and FV is the "total number of frozen variables". The positions of the selected K channels are fixed for all $n!$ FG's; but the capacity sum (CS) of these K transmission paths may change with changing stage order. Likewise, the number of frozen variables (FV) depends on the stage order of the factor graph and it can change from one FG to another. We use the CS and FV values of a factor graph in differentiating how efficient the related FG decoder of the polar code is; because keeping the capacity-sum at its maximum value supports the polar code concept of activating the maximum-capacity channels, while freezing the low-capacity ones. Factor graphs with the same FV value are grouped into "equi-FV sets" and those with the same CS value are gathered into "equi-CS sets".

In Chapter 4, we evaluate and compare BP decoding performances on single (1-FG) and multiple factor graph (M-FG) decoders of the RM and polar codes over a binary erasure channel. For polar codes, we also examine the choice of the best values of M for multiple-FG BP decoding over the BEC and investigate the choice of the best set of FG's for multiple-FG BP decoding.

For polar codes, single-FG decoder gives significantly better performance with increasing number of frozen variables. For the multiple-FG decoders, as differently from Doğan's choice of FG's, we restrict our search space to FG's with the highest number of frozen variables. Increasing the number of FG's in M-FG decoders does not result in significant performance improvement over 1-FG's, i.e., for the (128, 64) adaptive polar codes over BEC(0.35), there is less than 3% performance improvement of the 16-FG decoder over the single-FG decoder; which is not enough to justify the use of 16-FG decoder instead of 1-FG

For the (64, 32) adaptive polar codes, reducing the set of FG’s in multiple-FG decoders to 4-element equi-CS sets gives better results. 4-FG BP decoding performance of the best performing equi-CS set, is almost similar to that of the 12-FG decoder with 12 highest-CS-valued FG’s within the maximum equi-FV set, although it has 3 times lower complexity. For the M-FG decoding of larger block length-codes, equi-CS sets disappear, and selection of a small number of highest-CS-valued FG’s within the maximum equi-FV set seems promising.

For the RM codes, all $n!$ different factor graphs have the same FV and the same CS values. Because of the constant values of CS and FV, single-FG decoding performances of all FG’s are similar. As opposed to polar codes, M-FG decoding performance of Reed-Muller codes improves considerably with increasing M; however, it can approach the 1-FG performance of the adaptive polar code with similar parameters for quite high M ($\approx$152) over BEC(0.35).

Factor graphs, which have the highest CS values, conform the attitude of polar codes that chooses the highest-capacity paths; hence, one expects that BP decoders using these factor graphs to perform better. However, for the (64, 32) polar codes over binary erasure channels with various erasure rates, we have observed in Section 4.6 that the 4-FG decoder, which uses the fifth-highest-CS-valued equi-CS set, performs slightly better than the 4-FG decoder that employs the highest-CS-valued equi-CS set. Many trials with different channels and different erasure patterns resulted in similar results, equi-CS set #5 always had this small advantage over set #1 for an unexplained reason.

It is a future work to examine the structure of FG’s in the best performing sets, and determine better selection criteria for multiple-FG decoders. Invariability of CS and FV values with respect to the factor graph stage permutations of Reed-Muller codes also needs to be proved.

REFERENCES

[Abbas et al., 2015] Abbas, S. M., Fan, Y., Chen, J., & Tsui, C. Y. (2015). Low complexity belief propagation polar code decoder. *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, 2015–December, 6–11.

[Abbas et al., 2017] Abbas, S. M., Fan, Y. Z., Chen, J., & Tsui, C. Y. (2017). High-Throughput and Energy-Efficient Belief Propagation Polar Code Decoder. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(3), 1098–1111.

[Arıkan, 2008] Arıkan, E. (2008). A performance comparison of polar codes and reed-muller codes. *IEEE Communications Letters*, 12(6), 447–449.

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

[Arıkan, 2010] Arıkan, E. (2010). Polar codes: A pipelined implementation. 4th International Symposium on Broadband Communication (ISBC 2010), 2–4. ing Codes for Symmetric, *IEEE Transactions on Information Theory*, 55(7), 3051–3073.

[Berrou, 1993] Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo-codes. *Proceedings of ICC '93 - IEEE International Conference on Communications*, 2(1), 1064–1070. (1950).

[Bose and Ray-Chaudhuri, 1960] Bose, R. C. and Ray-Chaudhuri, D. K. (1960). On a class of error correcting binary group codes. *Information and control*, 3(1):68– 79.

[Costello, 2007] Forney, G. D., & Costello, D. J. (2007). Channel coding: The road to channel capacity. *Proceedings of the IEEE*, 95(6), 1150–1177.

[Doğan, 2015] Doğan, O. (2015). An Investigation on Belief Propagation Decoding of Polar Codes, MSc. Thesis.

[Eslami and Pishro-Nik, 2013] 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 and Barry, 2014] Fayyaz, U. U., & Barry, J. R. (2014). Low-Complexity Soft-Output Decoding of Polar Codes. *IEEE Journal on Selected Areas in Communications*, 32(5), 958–966.

[Forney, 2001] Forney, J. (2001). Codes on graphs: Normal realizations. *IEEE Transactions on Information Theory*, 47(2), 520–548.

[Golay, 1949] Golay, M. J. (1949). Notes on digital coding.

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

[Hussami, 2009] Hussami, N., Korada, S. B., and Urbanke, R. (2009). Performance of polar codes for channel and source coding. *In Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 1488–1492.

[Korada, 2009] Korada, S. B. (2009). Polar codes for channel and source coding., PhD thesis.

[Lin et al., 2015] Lin, J., Xiong, C., & Yan, Z. (2015). Reduced complexity belief propagation decoders for polar codes. *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, 2015–December(2).

[MacKay and Neal, 1996] MacKay, D. J. C., & Neal, R. M. (1996). Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 33(6), 457.

[Muller, 1954] Muller, D. E. (1954). Application of Boolean algebra to switching circuit design and to error detection. *Electronic Computers, Transactions of the IRE Professional Group on*, (3):6–12.

[Reed, 1954] 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 and Solomon, 1960] Reed, I. S. and Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathe- matics*, 8(2):300–304.

[Sha et al., 2016] Sha, J., Lin, J., & Wang, Z. (2016). Stage-combined belief propagation decoding of polar codes. *IEEE International Symposium on Circuits and Systems* (Vol. 2016–July, pp. 421–424).

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. T*he Bell System Technical Journal*, 27(July 1928), 379–423.

[Vangala et al., 2014] H. Vangala, E. Viterbo, & Y. Hong. (2014). Permuted Successive Cancellation Decoder for Polar Codes. *Information Theory and Its Applications (ISITA), 2014 International Symposium on*, (C), 438–442.

[Xu et al., 2015] Xu, J., Che, T., & Choi, G. (2015). XJ-BP: Express journey belief propagation decoding for polar codes. *2015 IEEE Global Communications Conference*, GLOBECOM 2015.

[Yuan and Parhi, 2013] Yuan, B., & Parhi, K. K. (2013). Architecture optimizations for BP polar decoders. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (1), 2654–2658.

[Zhang, 2014] Zhang, Y., Zhang, Q., Pan, X., Ye, Z., & Gong, C. (2014). A simplified belief propagation decoder for polar codes. *2014 IEEE International Wireless Symposium, IWS 2014*, 2–5.

APPENDIX A

FACTOR GRAPHS OF THE MAXIMUM EQUI-FV SET AND FV DISTRIBUTION FOR THE (128, 64) ADAPTIVE POLAR CODE

For the (128, 64) adaptive polar code over BEC(0.35), $n! = 7!$ = 5040 different FG’s can be implemented. In Section 4.4.2, BP decoding performance, using FG's of the maximum equi-FV set, for the (128, 64) adaptive polar code over BEC(0.35) is evaluated. Factor graphs that have the highest FV value are presented in Table A.1 and distribution of the number of frozen variables among 5040 factor graphs is given in Table A.2.

Table A.1: Factor graphs in the maximum equi-FV set for the (128, 64) adaptive polar code with ϵ=0.35.

Factor Graphs	FV	CS
6745321, 6754321, 7645321, 7654321	210	63.7043
6743521, 6753421, 7643521, 7653421	210	63.5861
6745231, 6754231, 7645231, 7654231	210	63.4057
6745312, 6754312, 7645312, 7654312	210	63.3012
6743512, 6753412, 7643512, 7653412	210	63.1289
6743251, 6753241, 7643251, 7653241	210	63.0892
6742531 ,6752431, 7642531, 7652431	210	63.0049
6742351, 6752341, 7652341, 7642351	210	62.8065
6745132, 6754132, 7645132, 7654132	210	62.5524
6745213, 6754213, 7645213, 7654213	210	62.5195
6743152, 6753142, 7653142, 7643152	210	62.1039
6745123, 6754123, 7645123, 7654123	210	62.0692
6743215, 6753214, 7643215, 7653214	210	61.9594
6742513, 6752413, 7642513, 7652413	210	61.9120
6742315, 6752314, 7652314, 7642315	210	61.5242
6743125, 6753124, 7643125, 7653124	210	61.4313
6742153, 6752143, 7642153, 7652143	210	60.6754
6742135, 6752134, 7642135, 7652134	210	60.4860

Table A.2: Distribution of the number of frozen variables (FV) for the (128, 64) adaptive polar code with erasure rate ϵ =0.35 among 5040 factor graphs.

FV	Number of the FG’s	Maximum and Minimum CS Values
210	72	63.7043, 60.4860
206	72	63.6297, 60.3828
202	120	63.5654, 60.1537
198	96	63.5122, 60.4111
194	192	63.4479, 60.1236
190	120	63.3734, 59.9573
186	264	63.1533, 59.2942
182	168	62.8258, 58.7738
178	192	62.4509, 58.6319
174	144	62.3322, 58.1320
170	336	62.1136, 58.1184
166	168	61.6304, 58.0956
162	288	61.7713, 57.6797
158	120	61.7147, 57.4161
154	288	61.5782, 57.4785
150	48	61.0129, 57.3914
146	456	61.4195, 57.2532
142	120	61.3008, 57.2943
138	384	61.0700, 56.7822
134	96	61.1300, 56.9483
130	312	60.8570, 56.4660
126	48	57.9701, 56.5059
122	288	59.8194, 56.2410
118	144	59.5619, 56.6240
114	96	58.9773, 56.6316
110	96	59.0316, 56.2428
106	192	58.9414, 55.9530
98	120	57.7028, 55.8789