A Study on Inference on Graphical Models and its Application to Channel Decoding Problems

グラフィカルモデル上の確率推論と通信路復号問題への応用に関する研究

February 2011

HORII Shunsuke

A Study on Inference on Graphical Models and its Application to Channel Decoding Problems

グラフィカルモデル上の確率推論と通信路復号問題への応用に関する研究

February 2011

Waseda University
Graduate School of Science and Engineering
Major in Mathematical Sciences, Research on Topology

HORII Shunsuke

Acknowledgments

本論文をまとめるにあたり，大変多くの御支援と御指導を賜りました．ここに深く感謝の意を表します．

まず，本論文の主査として御指導頂いた早稲田大学基幹理工学部応用数理学科 松嶋敏泰教授に心より感謝致します．松嶋教授には，著者が学部生時代よりお世話になり，研究はもとより，教育に対する姿勢や社会人としての心構えなど，数え切れないほどの御助言を頂きました．特に，研究においては，問題の本質を見抜き，物事を高い視点から見て考えるということの大切さを教わりました．著者が本論文をまとめることができたのも，松嶋教授の熱意ある御指導の賜物です．

また，論文の副査として有益な御助言を頂きました，早稲田大学理工学術院 平澤茂一名誉教授，早稲田大学基幹理工学部応用数理学科 大石進一教授に厚く御礼申し上げます．特に，平澤教授には大学院時代から長期間に渡り，貴重な時間を割いて熱心に議論をして頂き，また数多くの激励のお言葉を頂きました．それらは私が研究を続ける原動力になりました．

早稲田大学基幹理工学部応用数理学科松嶋研究室の諸先輩方にも厚く御礼申し上げます．横浜商科大学浮田善文准教授，北見工業大学前田康成助教，青山学院大学吉田隆弘助手，専修大学野村亮講師，日本電気株式会社峯松一彦氏，青山学院大学斎藤友彦助手，サイバー大学小泉大城助手，早稲田大学須子統太助教，東京工科大学小林直人助教の各氏には，日頃から研究室のゼミを通じて議論をして頂き，数多くの御助言を頂きました．特に須子助教には，著者が学部生時代より指導をして頂き，研究の進め方等多くのことを教わりました．

また，元早稲田大学創造理工学部経営システム工学科平澤研究室の皆様にも，研究室の枠を超えて議論をして頂き，数多くの御助言を頂きました．心より御礼申し上げます．

本研究成果の論文投稿で，有益な御助言を賜った査読委員と編集委員の皆様に感謝致します．また，情報理論とその応用学会をはじめとする数々の研究会で議論をして頂いた先生方に感謝致します．

本研究は以上をはじめとする多くの方々の御指導，御支援によって成し得たものです．お世話になった全ての皆様に心より御礼を申し上げます．

最後になりますが，著者の研究生活を暖かく見守り，支えてくれた家族に心より感謝の意を送ります．

2011 年 2 月

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Probabilistic inference is a very important problem that arises in several applications including channel coding theory, image processing and speech recognition. The probabilistic inference problem is defined as follows. We assume that the unobservable random variables, $\boldsymbol{x} = (x_1, \cdots, x_N)$, are generated from the probability distribution $p(\boldsymbol{x})$ and the observable random variables, $\boldsymbol{y}$, are generated from the conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$. Then, the probabilistic inference problem is the problem of estimating $\boldsymbol{x}$ given $\boldsymbol{y}$.

Depending on the criteria for evaluation, there are two optimum decision rules. One is the maximum a posterior (MAP) probability estimator, which maximizes the posterior distribution of unobservable variables for the given observable variables. Another is the maximum posterior marginal (MPM) estimator, which maximizes the marginals of the posterior distributions of unobservable variables for the given observable variables. With either rule, the optimum estimation is generally computationally difficult, i.e., in general, the computational complexity of these estimators is exponentially proportional to the number of unobservable variables.

A graphical model and probabilistic inference algorithms on the graph are used to tackle the probabilistic inference problem. A graphical model is a probabilistic model that expresses the statistical dependence structure of random variables in a graph. This model shows the factorized structure of the probability distribution. Several types of models can be used as graphical models, such as the Bayesian network, Markov random field, and factor graph. In this thesis, we mainly deal with the factor graph. A factor graph is a bipartite graph that expresses the factorization structure of the function. It contains some variable nodes, which correspond to variables, function nodes, which correspond to functions, and edge connections between the variable and function nodes.

The belief propagation (BP) algorithm is one of the most famous algorithms for solving probabilistic inference problems through graphical models. We can define the BP algorithm for the operations on the factor graph. The output of the algorithm coincides with the optimum estimator if the graphical model has no cycles. The graphical model and BP algorithm have provided excellent results when applied to error-correcting codes. Error-correcting codes are used to reliably transmit information through a noisy channel. The best codes currently available include turbo codes and low-density parity check (LDPC) codes, which are based on graphical models. Employing the BP algorithm as the decoding algorithm for turbo codes or LDPC codes, we can realize near-limit performance.

On the other hand, as an alternative decoding algorithm for LDPC codes, linear programming (LP) based decoding algorithms are attracting considerable attention. Given that we can consider the MAP estimation problem of binary linear codes as an instance of 0-1 integer programming problem, the LP decoder solves the relaxation problem of the 0-1 integer programming problem. Compared to the BP algorithm, the LP-based decoding algorithm has many attractive features. For example, in combinatorial optimization, there are many methods to improve the approximate performance of LP relaxation.

1.2 Purpose of Research

When the decoding problem of the binary linear code is considered as an example of the inference problem of the graphical model, the factor graph corresponds to this problem has some specific features. First, the functions in the factor graph can be classified in two classes. One is the indicator function, which is a function defined on a set, indicating the membership of an element in its subset; this function outputs value 1 for all elements of the subset and the value 0 for all elements not in the subset. The other function is called the non-indicator function. Each indicator function in the graph is connected to more than one variable node, while each non-indicator function is connected to only one variable node. On the other hand, the factor graph for general probabilistic inference problems except for the decoding problem of binary linear codes possibly has non-indicator functions connected to more than one variable node. For example, if we deal with the decoding problem of binary linear codes over the multiple-access channels, the factor graph correspond to the problem has such non-indicator functions. For such problems, we cannot obtain the inference algorithm based on LP in the same manner as we can for the decoding problem of binary linear codes over the single-user memoryless channel.

Contribution of this thesis are as follows:

1. We extend the LP-based inference algorithm to the inference problem on graphical models in addition to the decoding problem of binary linear codes over the single-user memoryless channel.

2. We demonstrate the approach to decreasing the computational complexity of the inference algorithms (LP-based and BP) by converting the graph structure.

3. We improve the LP-based inference algorithm using combinatorial optimization theory.

First, we developed the LP-based inference algorithm for the inference problems of the general factor graph. In this problem, the graph has non-indicator functions that are connected to more than one variable node. We have resolved this problem by introducing some auxiliary variables and changing the graph structure.

Second, we demonstrated that the computational complexity of the inference algorithms based on LP or BP can be decreased. The computational complexity of these algorithms depends on the structure of the factor graph. Since the structure of the factor graph depends on the way the function is factorized, different factorization results in a different factor graph.

Third, we improved the LP-based inference algorithm using combinatorial optimization theory. A feature of the LP-based inference algorithm is that if the optimum solution of the linear program is an integer, the solution is guaranteed to be MAP estimate, however, if the optimum solution is rational, it is not MAP estimate. Similar problems arising in general combinatorial optimization can be resolved using methods such as the branch-and-bound method, cutting-plane method, and branch-and-cut method. In this thesis, we apply these methods to the inference problem. We develop the inference algorithm based on the branch-and-cut algorithm.

This thesis is organized as follows.

In Chapter 2, we present some basic notations and discuss previous studies as preliminaries. We first formulate the probabilistic inference problems. Next, we discuss the factor graph and BP algorithm. Then, we formulate the decoding problems for various communication systems. We also provide that those problems can be considered as the probabilistic inference problems. Finally, we review the application of probabilistic inference algorithm for the decoding problems.

In Chapter 3, we provide the LP-based inference algorithm for general factor graphs. We first present the LP-based inference algorithm for the factor graph that does not include multi-degree non-indicator function nodes. Next, we propose the LP-based inference algorithm for the factor graph that includes multi-degree non-indicator function nodes.

In Chapter 4, we show that the computational complexity of the inference algorithms can be reduced by changing the factorization structure. First, we propose the BP algorithm based multiuser detection algorithm for a DS-CDMA channel. Next, we provide the LP-based decoding algorithm for Gaussian multiple-access channels.

In Chapter 5, we propose the branch-and-cut based decoding algorithm for the decoding problem of linear block codes over single-user memoryless channels. Next, we provide some numerical simulation results.

In Chapter 6, we conclude this thesis and discuss our future studies.

Chapter 2

Preliminaries

2.1 Probabilistic Inference Problem

In this section, we describe the formulation of probabilistic inference problem. Probabilistic inference problem is defined as follows. Let $X_n, n = 1, 2, \cdots, N$ be the random variables which take the values in the finite set $\mathcal{X}_n, n = 1, 2, \cdots, N$ with cardinality $|\mathcal{X}_n|$ and we denote those realizations as $x_n, n = 1, 2, \cdots, N$. We also define the set of random variables $\boldsymbol{X} \triangleq (X_1, X_2, \cdots, X_N)$ and it's realization $\boldsymbol{x} \triangleq (x_1, x_2, \cdots, x_N)$. The probability distribution function of $\boldsymbol{x}$ is denoted by $p(\boldsymbol{x})$. We assume that $\boldsymbol{x}$ is randomly generated according to the distribution $p(\boldsymbol{x})$ and we can't observe it. Let $\boldsymbol{Y} = (Y_1, Y_2, \cdots, Y_N)$ be the observable random variables and $\boldsymbol{y} = (y_1, y_2, \cdots, y_N)$ be its realizations where Y_n takes it's value in $\mathcal{Y}_n$ and it is generated according to the probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$.[1] The probabilistic inference problem is the problem to estimate the unobservable variables $\boldsymbol{x}$ given the observations $\boldsymbol{y}$.

According to the criteria for evaluation, there are two optimum decision rules. If we regard $\Pr\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\}$ as the error probability, we can minimize it by output the estimator $\hat{\boldsymbol{x}}$ as

$$\hat{\boldsymbol{x}} = \arg \max_{\boldsymbol{x} \in \mathcal{X}^N} p(\boldsymbol{x}|\boldsymbol{y}), \tag{2.1}$$

where we define $\mathcal{X}^N$ as $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_N$. This estimator is called Maximum A Posterior (MAP) estimator. On the other while, if we regard $\Pr(\hat{x}_n \neq x_n), n = 1, 2, \cdots, N$ as the error probability, we can minimize it by output the estimator $\hat{x}_n, n = 1, 2, \cdots, N$ as

$$\hat{x}_n = \arg \max_{x_n \in \mathcal{X}_n} p(x_n|\boldsymbol{y}) \quad n = 1, 2, \cdots, N, \tag{2.2}$$

where $p(x_n|\boldsymbol{y})$ is defined as

$$p(x_n|\boldsymbol{y}) = \sum_{x_1 \in \mathcal{X}_1} \cdots \sum_{x_{n-1} \in \mathcal{X}_{n-1}} \sum_{x_{n+1} \in \mathcal{X}_{n+1}} \cdots \sum_{x_N \in \mathcal{X}_N} p(\boldsymbol{x}|\boldsymbol{y}). \tag{2.3}$$

This estimator is called Maximum a Posterior Marginal (MPM) estimator.

[1] For simplicity, we assume that the number of unobservable variables and observable variables are equivalent. It is easy to remove this assumption.

In general, the computational complexity of both the MAP estimator and MPM estimator is $O(2^N)$ and hence we face the computational difficulty. To construct an efficient algorithm to compute the optimum estimator, we should utilize the structure of the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$.

2.2 Factor Graph and Belief Propagation Algorithm

The Belief Propagation (BP) algorithm is one of the most famous algorithm to calculate the MAP estimator or MPM estimator efficiently. In general, if we use the terminology "BP algorithm", it indicates two algorithms. One is the Sum-Product (SP) algorithm and another is the Max-Product (MP) algorithm.[2] Both the SP algorithm and the MP algorithm can be defined on the graph named "factor graph" [1]. The factor graph is a graph that express the structure of factorization of a function. If the factor graph correspond to the posterior distribution function has no cycles, SP algorithm outputs the MPM estimate and MP algorithm outputs the MAP estimate. On the other hand, if the factor graph has any cycles, both the SP algorithm and MP algorithm don't output the optimum estimate in general, however, it is known that these algorithms perform well in many cases.[3]

2.2.1 Factor Graph

Let $\{X_1, X_2, \dots, X_N\}$ be a set of N discrete random variables and let x_n be the possible realizations of random variable X_n. Let $p(\boldsymbol{x}|\boldsymbol{y})$ be the posterior probability distribution. We consider the case when $p(\boldsymbol{x}|\boldsymbol{y})$ can be factorized into a product of functions as follows,

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z}\prod_{a\in\mathcal{A}} f_a(\boldsymbol{x}_a), \tag{2.4}$$

where $\mathcal{A}$ is a discrete index set. For $a \in \mathcal{A}$, we defined $\mathcal{N}(a) = \left\{a_0, a_1, a_{|\mathcal{N}(a)|-1}\right\}$ and $\boldsymbol{x}_a = (x_{a_0}, x_{a_1}, \cdots, x_{a_{|\mathcal{N}(a)|-1}})$. $\boldsymbol{x}_a$ is an argument of f_a and Z is a constant which is not affected by $\boldsymbol{x}$.

Definition 2.1 [1] A *factor graph* is a bipartite graph that expresses the structure of the factorization. A factor graph has *variable nodes* (drawn as a circle), which represent the variables $\{x_n\}_{n=1,2,\cdots,N}$, *factor nodes* (drawn as a square), which represent the functions $\{f_a\}_{a\in\mathcal{A}}$, and edge-connections between variable nodes x_n and factor node f_a if and only if x_n is an argument of f_a.

Example 2.1 The graph in Fig. 2.1 is the factor graph corresponds to the posterior probability function,

$$p(x_1, x_2, x_3, x_4|\boldsymbol{y}) = \frac{1}{Z} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4). \tag{2.5}$$

[2] The MP algorithm is also called Mini-Sum algorithm since if we take the negative logarithm of the function, the problem to seek the maximum is reduced to the problem to seek the minimum and the product operation is changed to the summation operation.

[3] SP algorithm is also often used as an approximation algorithm to compute the MAP estimate.

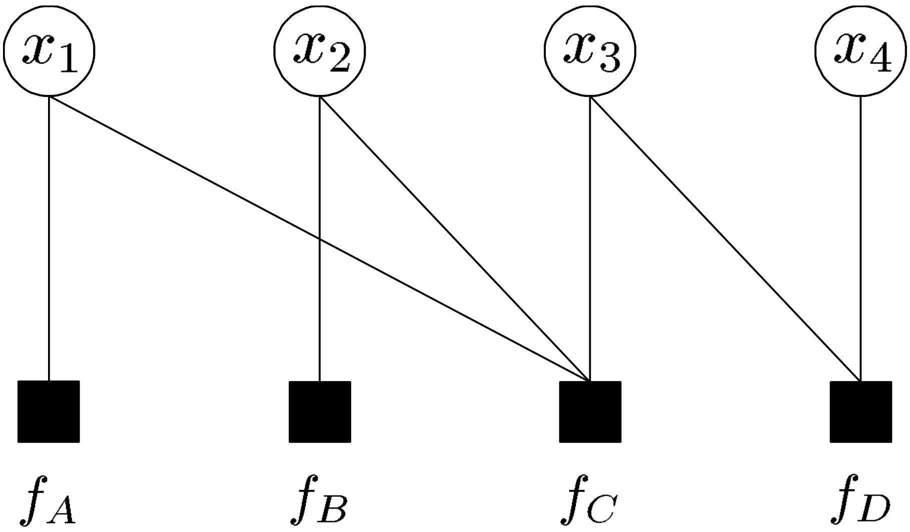

Figure 2.1: An example of factor graph for the posterior probability function, $p(x_1, x_2, x_3, x_4|\boldsymbol{y}) = \frac{1}{Z} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4)$.

In this example, $\mathcal{A} = \{A, B, C, D\}$, $\boldsymbol{x}_A = (x_1)$, $\boldsymbol{x}_B = (x_2)$, $\boldsymbol{x}_C = (x_1, x_2, x_3)$, and $\boldsymbol{x}_D = (x_3, x_4)$.

2.2.2 The Sum-Product Algorithm

Sum-Product (SP) algorithm is an algorithm to (approximately) compute the marginal of a function. Let $p(x_n|\boldsymbol{y})$ denote the marginal probability function obtained by marginalizing $P(\boldsymbol{x}|\boldsymbol{y})$ onto the variable x_n, i.e.,

$$p(x_n|\boldsymbol{y}) = \sum_{x_1 \in \mathcal{X}_1} \cdots \sum_{x_{n-1} \in \mathcal{X}_{n-1}} \sum_{x_{n+1} \in \mathcal{X}_{n+1}} \cdots \sum_{x_N \in \mathcal{X}_N} p(\boldsymbol{x}|\boldsymbol{y}). \tag{2.6}$$

The algorithm can be defined in terms of the set of operations on the factor graph [1].

The SP algorithm consists of *messages* $m_{n \to a}(x_n)$ from variable nodes to their neighboring factor nodes and messages $m_{a \to n}(x_n)$ from factor nodes to their neighboring variable nodes, and *beliefs* $q_n(x_n)$ of variable nodes. The messages and belief are vectors over the possible realizations of x_n.

The messages are updated as follows:

$$m^{(t)}_{a \to n}(x_n) = \sum_{\boldsymbol{x}_a \backslash x_n} f_a(\boldsymbol{x}_a) \prod_{n' \in \mathcal{N}(a) \backslash n} m^{(t-1)}_{n' \to a}(x_{n'}), \tag{2.7}$$

$$m^{(t-1)}_{n \to a}(x_n) = \alpha_{n,a} \prod_{a' \in \mathcal{N}(n) \backslash a} m^{(t-1)}_{a' \to n}(x_n), \tag{2.8}$$

where $\mathcal{N}(a) \setminus n$ denotes all neighboring nodes of the node a except for the node n, and $\mathcal{N}(n) \setminus a$ is defined in a similar way. The operation $\sum_{\boldsymbol{x}_a \backslash x_n}$ denotes a sum over all the variables $\boldsymbol{x}_a$ except for x_n. Normalization constant $\alpha_{n,a}$ is determined so that $\sum_{x_n} m_{n \to a}(x_n) = 1$ holds. In general, the messages are initialized to $m^{(0)}_{a\to n}(x_n) = 1$ and $m^{(0)}_{n\to a}(x_n) = 1$ for all factor nodes a, variable nodes n, and possible values of x_n.

The beliefs are computed as

$$q^{(t)}_n(x_n) = \alpha_n \prod_{a\in\mathcal{N}(n)} m^{(t)}_{a\to n}(x_n), \tag{2.9}$$

where α_n is a normalization constant which is determined so that $\sum_{x_n} q_n(x_n) = 1$ holds. Messages are updated iteratively until they (hopefully) converge, then compute the beliefs from (2.9) and use them as approximations to the exact marginal probability functions $\{p(x_n|\boldsymbol{y})\}$. The SP algorithm calculates the exact marginal probability functions when the factor graph has no cycles.

In contrast with the case when the factor graph has no cycles, the results of the SP algorithm to the factor graph with cycles cannot be exact marginal probability functions. However, according to the research about the decoding problems for LDPC codes, the SP algorithm often gives good approximation results, especially when the number of cycles is small.

2.3 Channel Models

2.3.1 Single-User Channel

An information source emits a digital signal called an information symbol. A sequence of information symbols with length K called an information vector, which is denoted by $\boldsymbol{b} = (b_1, b_2, \cdots, b_K) \in \mathcal{B}^K$, where $\mathcal{B}$ is a finite set which represents the information symbol. The encoding function is a mapping from $\mathcal{B}^K$ to $\mathcal{X}^N$, where $\mathcal{X}$ denotes a finite set and N denotes the length of the codeword. The range of the mapping is called a *code*, which is denoted by $\mathcal{C}$, and the elements in $\mathcal{C}$ is called the codewords. In general, the encoding function is an injection mapping, i.e., there is a one-to-one correspondence between a information vector and codeword.

The transmitter encodes the information vector $\boldsymbol{b} \in \mathcal{B}^K$ to the codeword $\boldsymbol{x} \in \mathcal{C}$ by using the encoder. A codeword $\boldsymbol{x} \in \mathcal{C}$ is sent to the destination through a noisy channel. The receiver receives a noise-corrupted sequence $\boldsymbol{y} = (y_1, y_2, \cdots, y_N) \in \mathcal{Y}^N$, where $\mathcal{Y}$ is a set of received signal.

The receiver estimates the transmitted information vector by using the information of given received sequence. Since the encoding function is an injection, it is equivalent to estimate the transmitted codeword. In this thesis, we assume that the decoder estimates the transmitted codeword and this procedure is called decoding.

Probabilistic Model

We assume that the information vectors are generated according to a probability distribution $p(\boldsymbol{b})$ and the process of the noise generation on the channel is governed by a conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$. Since the information vector is considered as a random variable, the codeword $\boldsymbol{x}$ is also considered as a random variable. We denote the probability distribution of $\boldsymbol{x}$ as $p(\boldsymbol{x})$.

Assuming that the source coding is sufficiently implemented, we can assume that the information vector $\boldsymbol{b}$ is uniformly distributed, i.e.,

$$p(\boldsymbol{b}) = \frac{1}{|\mathcal{B}|^K}. \tag{2.10}$$

According to the property of the encoding function, we can also describe the probability distribution of $\boldsymbol{x}$ as

$$p(\boldsymbol{x}) = \begin{cases} 1/|\mathcal{B}|^K & \text{if } \boldsymbol{x} \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases}. \tag{2.11}$$

The channel is called memoryless when $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{n=1}^N p(y_n|x_n)$ is satisfied, where $p(y_n|x_n)$ is a conditional probability distribution of the received symbol y_n given the codeword symbol x_n.

Example 2.2 Binary-Input Additive White Gaussian Noise (BIAWGN) Channel is widely used as a model for the memoryless channel. Let the codeword alphabet $\mathcal{X}$ is $\{0, 1\}$. Each codeword symbols is transformed by the binary-bipolar conversion, that is, each $\{0, 1\}$ symbol is converted to $\{+1, -1\}$. The converted signal of x_n is denoted by $\tilde{x}_n$. The received symbol y_n takes on the real values, i.e., $\mathcal{Y} = \mathbb{R}$. The received symbol y_n is modeled as,

$$y_n = x_n + \epsilon_n, \tag{2.12}$$

where ϵ_n is a Gaussian random variable with zero mean and variance σ^2. The conditional probability distribution $p(y_n|x_n)$ of the BIAWGN channel is denoted by

$$p(y_n|x_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_n - \tilde{x}_n)^2}{2\sigma^2}\right). \tag{2.13}$$

2.3.2 Multiple-Access Channel

Multiple-access channel is a channel that some senders send information to a common receiver. As an example, a communication system between a set of cell phones and base station can be modeled as multiple-access channel. Here, we describe the U-user multiple-access channel.

Each user u encodes its information vector $\boldsymbol{b}_u = (b_{u,1}, b_{u,2}, \cdots, b_{u,K_u}) \in \mathcal{B}^{K_u}$ to $\mathcal{X}^N$, where K_u is the length of the information vector of user u. The code of user u is denoted by $\mathcal{C}_u$. The receiver estimates the transmitted information vectors of codewords of all users.

Probabilistic Model

As in the case for single-user channel, we assume that the information vectors are generated according to a uniform distribution, i.e.,

$$p(\boldsymbol{b}_u) = \frac{1}{|\mathcal{B}|^{K_u}}, \quad u = 1, 2, \cdots, U, \tag{2.14}$$

$$p(\boldsymbol{x}_u) = \begin{cases} 1/|\mathcal{B}|^{K_u} & \text{if } \boldsymbol{x}_u \in \mathcal{C}_u \\ 0 & \text{otherwise} \end{cases}, \quad u = 1, 2, \cdots, U. \tag{2.15}$$

We also assume that the received vector $\boldsymbol{y}$ is generated from the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U)$. The channel is called memoryless when it holds $p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) = \prod_{n=1}^{N} p(y_n|x_{1,n}, x_{2,n}, \cdots, x_{U,n})$, where $p(y_n|x_{1,n}, x_{2,n}, \cdots, x_{U,n})$ is a conditional probability distribution of the received symbol y_n given the codeword symbols $(x_{1,n}, x_{2,n}, \cdots, x_{U,n})$.

Example 2.3 Gaussian multiple-access channel is widely used as a model for the memoryless multiple-access channel. Let the codeword alphabet $\mathcal{X}$ is $\{0, 1\}$. Each codeword symbols is transformed by the binary-bipolar conversion, that is, each $\{0, 1\}$ symbol is converted to $\{+1, -1\}$. The converted signal of x_n is denoted by $\tilde{x}_n$. The received symbol y_n takes on the real values, i.e., $\mathcal{Y} = \mathbb{R}$. The received symbol y_n is modeled as,

$$y_n = \sum_{u=1}^{U} \tilde{x}_{u,n} + \epsilon_n, \tag{2.16}$$

where ϵ_n is a Gaussian random variable with zero mean and variance σ^2. The conditional probability distribution $p(y_n|x_{1,n}, x_{2,n}, \cdots, x_{U,n})$ is denoted by

$$p(y_n|x_{1,n}, x_{2,n}, \cdots, x_{U,n}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_n - \sum_{u=1}^{U} \tilde{x}_{u,n})^2}{2\sigma^2}\right). \tag{2.17}$$

2.4 Linear Block Codes

Here, we describe about the *linear block code*. Let $\mathbb{F}_q$ be a finite field with q elements. We assume that $\mathcal{B} = \mathcal{X} = \mathbb{F}_q$. When a code is defined as a linear subspace of the vector space of $\mathbb{F}_q^N$, it is called a *linear code.*

Linear block codes are linear codes such that an information vector is encoded independently of the other information vectors. Since a linear block code $\mathcal{C}$ of length N over F_q is a subspace of F_q^N, there must exist K so that $\mathcal{C}$ has a dimension K. A $K \times N$ matrix G is called a *generator matrix* if whose rows form a linearly independent basis for K dimensional subspace of F_q^N. Conversely, given a matrix $G \in \mathbb{F}_q^{K\times N}$ of rank K we can associate with it the code $\mathcal{C}$ as

$$\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{F}_q^N : \boldsymbol{x} = \boldsymbol{b}G,\ \boldsymbol{b} \in \mathbb{F}_q^K\right\}. \tag{2.18}$$

The matrix $H \in \mathbb{F}_q^{M\times N}$, whose dimension is $N-K$, is called *parity check matrix* when it satisfies

$$GH^T = \boldsymbol{O}. \tag{2.19}$$

We can also define a code $\mathcal{C}$ by a parity check matrix H as

$$\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{F}_q^N : H\boldsymbol{x}^T = \boldsymbol{0}^T\right\}. \tag{2.20}$$

Let H_{mn} be the value of (m, n) element of H. We define the sets $\mathcal{N}(m)$ and $\mathcal{M}(n)$ as

$$\mathcal{N}(m) = \{n : H_{mn} \neq 0\}, \quad \mathcal{M}(n) = \{m : H_{mn} \neq 0\}. \tag{2.21}$$

The m-th row of H indicates that $\sum_{n\in\mathcal{N}(m)} x_n = 0$. The constraint is called a parity-check constraint.

Linear code is called *binary linear code* when the number of elements of the field q is 2. In the case of binary linear code, we can consider $\mathbb{F}_2 = \{0,1\}$ defining addition and multiplication by Table 2.1.

Table 2.1: Addition and multiplication tables for $\mathbb{F}_2$.

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

2.5 Communication Systems and Channel Decoding Problems

The communication systems consist of coding/decoding scheme and channel model. Given that there are various coding/decoding schemes and channel models, we can consider various communication systems. In this section, we describe the communication systems which are dealt in this thesis.

2.5.1 Decoding Problem of the Binary Linear Codes over Memoryless Channels

Here, we consider the single-user communication system that the encoder employ the binary linear codes and the channel is modeled as the memoryless channel which is depicted in fig. 2.2.

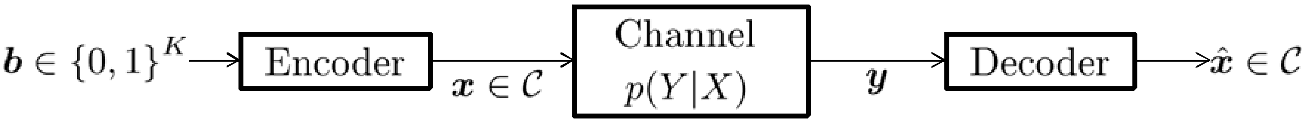
$\boldsymbol{b} \in \{0,1\}^K$
Encoder
$\boldsymbol{x} \in \mathcal{C}$
Channel
$p(Y|X)$
$\boldsymbol{y}$
Decoder
$\hat{\boldsymbol{x}} \in \mathcal{C}$

Figure 2.2: Single-user communication system with binary linear codes and memoryless channel.

The encoder encodes the information vector $\boldsymbol{b} \in \{0,1\}^K$ to the codeword $\boldsymbol{x} \in \mathcal{C}$, where $\mathcal{C}$ is the binary linear code with parity check matrix $H \in \{0,1\}^{M\times N}$. The transmitted codeword $\boldsymbol{x}$ go through the memoryless channel $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{n=1}^{N} p(y_n|x_n)$. The decoder receive the noise corrupted sequence $\boldsymbol{y}$ and estimate the transmitted codeword $\boldsymbol{x}$. First, we define the two kind of error probability, *block error probability* and *bit error probability.*

Definition 2.2 Let $\hat{\boldsymbol{x}}$ be the estimator for the transmitted codeword $\boldsymbol{x}$. Then the block error probability is defined as

$$P_B = \Pr\left\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\right\}. \tag{2.22}$$

Let $\hat{x}_n$ be the estimator for the n-th bit of transmitted codeword x_n. Then the bit error probability is defined as

$$P_{b,n} = \Pr\left\{\hat{x}_n \neq x_n\right\}. \tag{2.23}$$

In order to minimize the block error probability, the decoder must output the hypothesized information vector $\boldsymbol{x}$ which maximizes the posterior probability $p(\boldsymbol{x}|\boldsymbol{y})$. This type of decoding is called the *Maximum a Posterior probability (MAP) decoding.*

Definition 2.3 MAP decoding rule is to output $\hat{\boldsymbol{x}}$ which is defined as

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{C}} p(\boldsymbol{x}|\boldsymbol{y}). \tag{2.24}$$

If we assume that each codeword in $\mathcal{C}$ has an equal probability to be transmitted, the MAP decoding is reduced to maximize the likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$. This type of decoding rule is called the *Maximum Likelihood (ML) Decoding.*

Definition 2.4 ML decoding rule is to output $\hat{\boldsymbol{x}}$ which is defined as

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{C}} p(\boldsymbol{y}|\boldsymbol{x}). \tag{2.25}$$

In a similar fashion, to minimize the bit error probability, the decoder must output the hypothesized information symbol x_n which maximizes the posterior marginal probability $p(x_n|\boldsymbol{y})$. This type of decoding is called the *Maximum a Posterior Marginal probability (MPM) decoding.*

Definition 2.5 MPM decoding rule is to output $\hat{x}_n$ which is defined as

$$\hat{x}_n = \arg\max_{x_n\in\{0,1\}} p(x_n|\boldsymbol{y}). \tag{2.26}$$

MAP decoder minimizes the block error probability and MPM decoder minimizes the bit error probability, however, the complexity of these decoders is exponentially proportional to the length of the information vector, i.e., $O(2^K)$. Thus the MAP decoding and MPM decoding are impractical.

2.5.2 Decoding Problems for Synchronous DS-CDMA System

When the *spreading codes* are used for the multiple-access channels, the communication system is called Direct Sequence Code Division Multiple Access (DS-CDMA) system. First, we explain the spreading code. We assume that $\mathcal{B} = \mathcal{X} = \{-1, +1\}$ and the length of information vector

K equals to 1. The spreading codes are the set of binary sequences which is defined by the signature sequence $\boldsymbol{s} \in \left\{-1/\sqrt{N}, +1/\sqrt{N}\right\}^N$. The spreading codes are defined as

$$\mathcal{C} = \left\{b\boldsymbol{s} \in \left\{-1/\sqrt{N}, +1/\sqrt{N}\right\}^N : b \in \{-1, +1\}\right\}. \tag{2.27}$$

The encoder encodes the information bit b to $b\boldsymbol{s}$.

Here, we consider a U-user synchronous DS-CDMA system which is depicted in 2.3.[4]

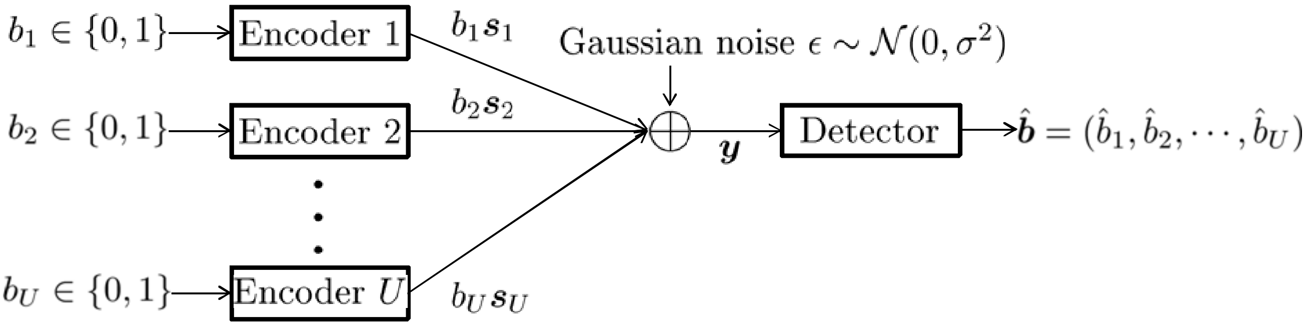

Figure 2.3: DS-CDMA model with Gaussian noise.

Each user u uses own signature sequence $\boldsymbol{s}_u \in \{-1, +1\}^N$ to encodes its information symbol. We also assume that the noise distribution is Gaussian. Then the received signal in a U-user synchronous CDMA system is described as

$$y_n = \sum_{u=1}^{U} b_u s_{u,n} + \epsilon_n, \qquad (n = 1, 2, \cdots, N), \tag{2.28}$$

where y_n is the received signal at chip interval n, and $b_u \in \{-1, 1\}$, and $\{s_{u,n} : n = 1, 2, \ldots, N\}$ denote the information bit and normalized unit energy signature sequence of user u respectively, and filtered noise $\{\epsilon_n\}$ are Gaussian with 0 mean and variance σ^2.

The decoding problem for the DS-CDMA system is often called the multiuser detection problem and decoder is called the multiuser detector. The multiuser detection problem is the problem to estimate the information bit sequence $\boldsymbol{b} = (b_1, b_2, \cdots, b_U)$ from the received signals $\boldsymbol{y} = (y_1, y_2, \cdots, y_N)$. We assume that the detector knows the signature sequences of all users.

As in the case for the decoding problem of the binary linear codes over memoryless channels, we define the two kind of error probabilities.

Definition 2.6 Let $\hat{\boldsymbol{b}}$ be the estimator for the transmitted bit sequence $\boldsymbol{b}$. Then, the block error probability for DS-CDMA is defined as

$$P_{B,CDMA} = \Pr\left\{\hat{\boldsymbol{b}} \neq \boldsymbol{b}\right\} \tag{2.29}$$

[4] The system is called synchronous when a received symbol y_n depends on only $\{x_{1,n}, x_{2,n}, \cdots, x_{U,n}\}$. Otherwise, the system is called asynchronous. We consider only the synchronous systems in the thesis.

Let $\hat{b}_u$ be the estimate of b_u for all $u = 1, 2, \cdots, U$. The bit error probability for DS-CDMA is defined as

$$P_{b,u,CDMA} = \Pr\left\{\hat{b}_u \neq b_u\right\} \tag{2.30}$$

We also define the MAP detector, ML detector and MPM detector as follows.

Definition 2.7 The MAP detector, ML detector and MPM detector are defined as the detectors which output

$$\hat{\boldsymbol{b}} = \arg \max_{\boldsymbol{b}\in\{-1,+1\}^N} p(\boldsymbol{b}|\boldsymbol{y}), \tag{2.31}$$

$$\hat{\boldsymbol{b}} = \arg \max_{\boldsymbol{b}\in\{-1,+1\}^N} p(\boldsymbol{y}|\boldsymbol{b}), \tag{2.32}$$

$$\hat{b}_u = \arg \max_{b_u\in\{-1,+1\}} p(b_u|\boldsymbol{y}), \tag{2.33}$$

respectively.

MAP detector minimizes the block error probability and MPM detector minimizes the bit error probability, however, the complexity of those detectors are exponential in the number of users $O(2^U)$. Therefore, these detectors are impractical when the number of users is large.

2.5.3 Decoding Problems of Binary Linear Codes over Memoryless Multiple-Access Channels

Here, we describe the U-user multiple-access communication system that each user employ the binary linear codes and the channel is memoryless which is depicted fig. 2.4.

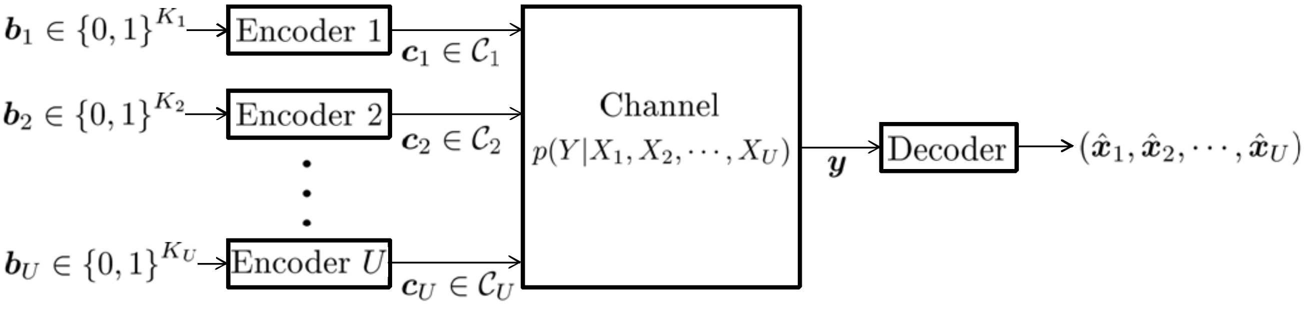
$\boldsymbol{b}_1 \in \{0,1\}^{K_1}$
Encoder 1
$\boldsymbol{c}_1 \in \mathcal{C}_1$
$\boldsymbol{b}_2 \in \{0,1\}^{K_2}$
Encoder 2
$\boldsymbol{c}_2 \in \mathcal{C}_2$
$\boldsymbol{b}_U \in \{0,1\}^{K_U}$
Encoder U
$\boldsymbol{c}_U \in \mathcal{C}_U$
Channel
$p(Y|X_1, X_2, \cdots, X_U)$
$\boldsymbol{y}$
Decoder
$(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_U)$

Figure 2.4: Multiple-access channel with binary linear codes and memoryless channel.

Each encoder encodes the information vector $\boldsymbol{b}_u$ to the codeword $\boldsymbol{x}_u \in \mathcal{C}_u$, where $\mathcal{C}_u$ is the binary linear code with parity check matrix $H^{(u)} \in \{0,1\}^{M_u \times N}$. All encoders sent the codewords in parallel and the transmitted codewords $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U)$ go through the memoryless multiple-access channel $p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) = \prod_{n=1}^{N} p(y_n|x_{1,n}, x_{2,n}, \cdots, x_{U,n})$. The decoder receive the sequence $\boldsymbol{y}$ and estimate the transmitted codewords $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U)$. The error probabilities are defined as follows.

Definition 2.8 The block error probability and the bit error probability are defined as,[5]

$$P_{B,MAC} = \Pr\left\{(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_U) \neq (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U)\right\} \tag{2.34}$$

$$P_{b,u,n,MAC} = \Pr\left\{\hat{x}_{u,n} \neq x_{u,n}\right\} \tag{2.35}$$

The optimum decoding rules are defined as follows.

Definition 2.9 The MAP decoding rule and MPM decoding rule are defined as rules to output,

$$(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_U) = \arg \max_{(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) \in \mathcal{C}_1 \times \mathcal{C}_2 \times \cdots \times \mathcal{C}_U} p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U | \boldsymbol{y}), \tag{2.36}$$

$$\hat{x}_{u,n} = \arg \max_{x_{u,n} \in \{0,1\}} p(x_{u,n} | \boldsymbol{y}), \tag{2.37}$$

respectively.

MAP decoder minimizes the block error probability and MPM decoder minimizes the bit error probability, however, the complexity of those decoders are $O(2^{UN})$. Therefore, those decoders are generally impractical.

2.6 Factor Graph Representations for Decoding Problems

In this section, we show that we can regard the decoding problems introduced in the previous section as the probabilistic inference problems. We also present the factor graphs of the problems.

2.6.1 Factor Graph for the Decoding Problem of the Binary Linear Codes over Memoryless Channels

The prior distribution $p(\boldsymbol{x})$ is described in (2.11). We note that $\boldsymbol{x}$ is in $\mathcal{C}$ if and only if $H\boldsymbol{x}^T = \boldsymbol{0}$ mod 2 is satisfied. We can translate the condition to

$$\sum_{n \in \mathcal{N}(m)} x_n = 0 \bmod 2,\ m = 1, 2, \cdots, M, \tag{2.38}$$

where $\mathcal{N}(m) = \{n : H_{mn} = 1\}$. As a result, the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is factorized as

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}) \tag{2.39}$$

$$= \frac{1}{Z} \prod_{n=1}^{N} p(y_n|x_n) \prod_{m=1}^{M} f_{I_j}(\boldsymbol{x}_{I_m}), \tag{2.40}$$

[5] We can also consider the error probability, $\Pr\{\hat{\boldsymbol{x}}_u \neq \boldsymbol{x}_u\}$. However, if we want to minimize this probability, we must solve the maximization problem and marginalization problem jointly. Since those kind of problems are not the scope of our study, we do not deal with the problem in the thesis.

where $\boldsymbol{x}_{I_m}$ is defined as $\boldsymbol{x}_{I_m} = (x_n)_{n\in\mathcal{N}(m)}$ and f_{I_m} is defined as

$$f_{I_m}(\boldsymbol{x}_{I_m}) = \begin{cases} 1 & \text{if } \sum_{n\in\mathcal{N}(m)} x_n = 0 \bmod 2, \\ 0 & \text{otherwise.} \end{cases} \qquad m = 1, 2, \cdots, M. \tag{2.41}$$

The factor graph for above factorization is depicted in Figure 2.5.

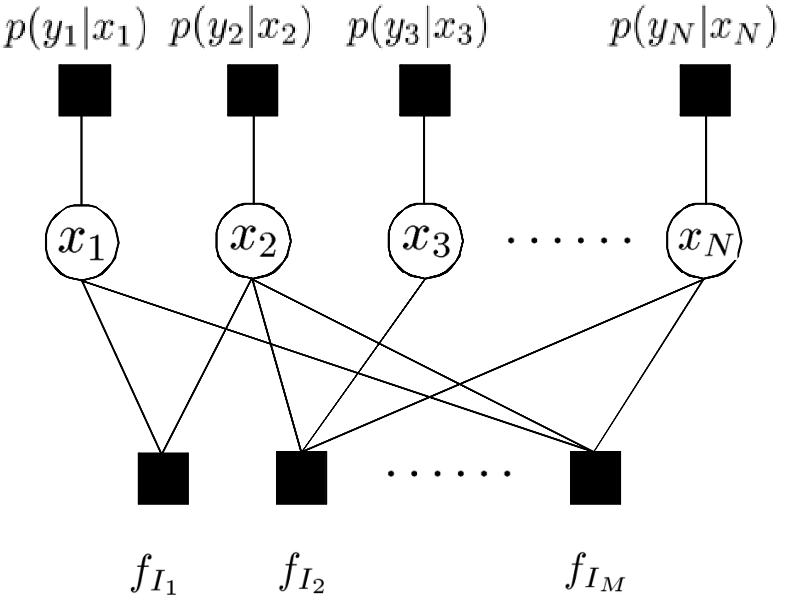

Figure 2.5: Factor graph for the decoding problem of binary linear codes over memoryless channels.

2.6.2 Factor Graph for the Decoding Problem for Synchronous DS-CDMA System

Based on the coding system and channel model (2.28), probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ is described as

$$p(\boldsymbol{y}|\boldsymbol{b}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{b}), \tag{2.42}$$

$$p(y_n|\boldsymbol{b}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(y_n - \sum_{u=1}^{U} b_u s_{u,n}\right)\right\}. \tag{2.43}$$

Given that the prior distribution $p(\boldsymbol{b})$ is uniform, the posterior distribution $p(\boldsymbol{y}|\boldsymbol{x})$ is

$$p(\boldsymbol{b}|\boldsymbol{y}) = \frac{1}{Z}\prod_{n=1}^{N} p(y_n|\boldsymbol{b}). \tag{2.44}$$

The factor graph for above factorization is depicted in Figure 2.6.

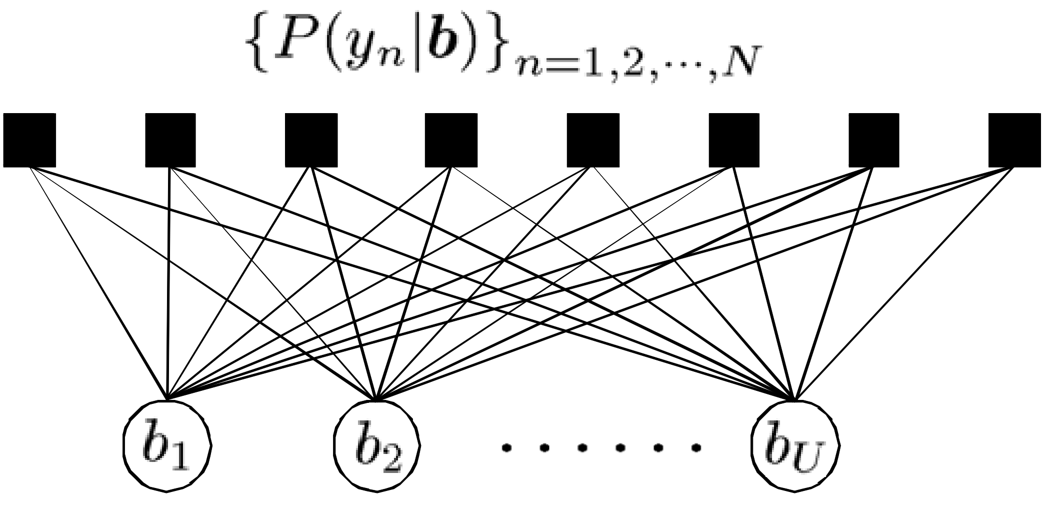

Figure 2.6: Factor graph for the decoding problem for Synchronous DS-CDMA system.

2.6.3 Factor Graph for the Decoding Problem of Binary Linear Codes over Memoryless Multiple-Access Channels

Since each encoder encodes the information vector independently, we can decompose the prior distribution $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) = \prod_{u=1}^{U} p(\boldsymbol{x}_u)$. Each component $p(\boldsymbol{x}_u)$ is described as (2.15). As in the case for the single-user channel case, the condition $\boldsymbol{x}_u \in \mathcal{C}_u$ is satisfied if and only if

$$\sum_{n \in \mathcal{N}_u(m)} x_{u,n} = 0 \text{ mod } 2, \ m = 1, 2, \cdots, M_u, \tag{2.45}$$

where $\mathcal{N}_u(m) = \left\{ n : H_{m,n}^{(u)} = 1 \right\}$.

As a result, the posterior distribution $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U|\boldsymbol{y})$ is given by

$$\begin{aligned} p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U|\boldsymbol{y}) &= \frac{1}{Z} p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) \prod_{u=1}^{U} p(\boldsymbol{x}_u) && (2.46) \\ &= \frac{1}{Z} \prod_{n=1}^{N} p(y_n|x_{1,n}, x_{2,n}, \cdots, x_{U,n}) \prod_{u=1}^{U} \prod_{m=1}^{M_u} f_{I_{u,m}}(\boldsymbol{x}_{I_{u,m}}) && (2.47) \end{aligned}$$

where $\boldsymbol{x}_{I_{u,m}} = (x_{u,n})_{n \in \mathcal{N}_u(m)}$ and

$$f_{I_{u,m}}(\boldsymbol{x}_{I_{u,m}}) = \begin{cases} 1 & \text{if } \sum_{n \in \mathcal{N}_u(m)} x_{u,n} = 0, \text{ mod } 2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.48}$$

The factor graph for above factorization is depicted in Figure 2.7.

2.7 Decoding Algorithms based on the Sum-Product Algorithm

In the previous section we showed that we can consider the decoding problems arising in various communication systems can be considered as the probabilistic inference problems. We also

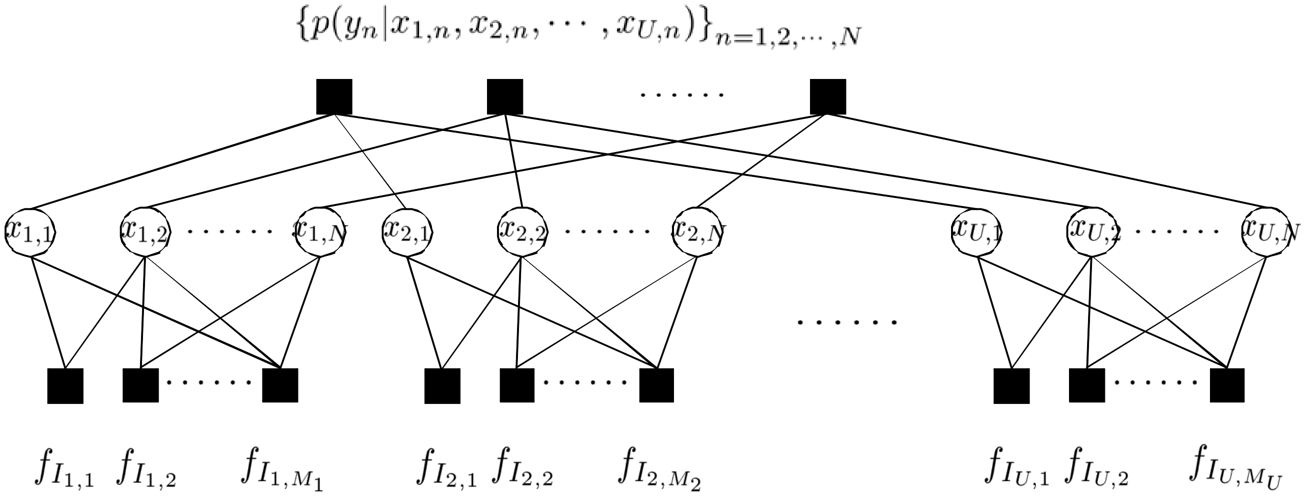

Figure 2.7: Factor graph for the decoding problem of binary linear codes over memoryless multiple-access channels.

showed the factor graphs correspond to those problems. Given that the SP algorithm is defined as the set of operations on the factor graph, we can apply the SP algorithm to those problems. In this section we describe the decoding algorithms based on the SP algorithm.

2.7.1 Sum-Product Algorithm based Decoding Algorithm for the Binary Linear Codes over Memoryless Channels

We describe the procedure of the SP decoding proposed in [2]. This procedure calculates the messages in the logarithmic domain and it simplifies the factor node process by considering the binary parity check constraints.

Initialization:

For all pairs (m, n) such that $H_{mn} = 1$,

$$\begin{cases} q_{mn}^{0,(0)} = p(y_n|0) \\ q_{mn}^{1,(0)} = p(y_n|1). \end{cases} \tag{2.49}$$

Factor Node Process:

For all pairs (m, n) such that $H_{mn} = 1$,

$$\begin{cases} r_{mn}^{0,(t)} = ((1 + \delta_{mn}^{(t)})/2) \\ r_{mn}^{1,(t)} = ((1 - \delta_{mn}^{(t)})/2). \end{cases} \tag{2.50}$$

where

$$\delta_{mn}^{(t)} = \prod_{n' \in \mathcal{N}_c(m) \backslash n} (q_{mn'}^{0,(t)} - q_{mn'}^{1,(t)}). \tag{2.51}$$

Variable Node Process:

For all pairs (m,n) such that $H_{mn}=1$,

$$\left\{\begin{array}{l} q_{mn}^{0,(t)} = \alpha_{mn} p(y_n|0) \prod_{m' \in \mathcal{N}_v(n) \backslash m} r_{m'n}^{0,(t-1)} \\ q_{mn}^{1,(t)} = \alpha_{mn} p(y_n|1) \prod_{m' \in \mathcal{N}_v(n) \backslash m} r_{m'n}^{1,(t-1)} \end{array}\right. \tag{2.52}$$

where α_{mn} is a normalizing constant such that $q_{mn}^{0,(t)} + q_{mn}^{1,(t)} = 1$.

Marginalization Process:

For all pairs (m,n) such that $H_{mn}=1$,

$$\left\{\begin{array}{l} q_{n}^{0,(t)} = \alpha_{n} p(y_n|0) \prod_{m' \in \mathcal{N}_v(n)} r_{m'n}^{0,(t)} \\ q_{n}^{1,(t)} = \alpha_{n} p(y_n|1) \prod_{m' \in \mathcal{N}_v(n)} r_{m'n}^{1,(t)} \end{array}\right. \tag{2.53}$$

where α_n is a normalizing constant such that $q_n^{0,(t)} + q_n^{1,(t)} = 1$.

Decoding Process:

For all n,

$$\hat{x}_n = \left\{\begin{array}{ll} 0 & q_n^{0,(t)} \geq q_n^{1,(t)} \\ 1 & q_n^{0,(t)} < q_n^{1,(t)}. \end{array}\right. \tag{2.54}$$

If $H\boldsymbol{x} = 0$ or the number of iterations exceeds a predetermined value, then this algorithm halts, else go to the factor node process.

2.7.2 Sum-Product Algorithm based Decoding Algorithm for the DS-CDMA System

According to (2.33), it is easy to see that the detection problem can be regarded as a probabilistic inference problem. To that end, Tanaka and Okada suggested to apply BP algorithm for the detection problem [3].

They used factorization (2.44). Corresponding factor graph is expressed as Fig.2.6. The factor graph would be a complete bipartite graph and U variable nodes represent the information bits and N factor nodes represent the likelihood functions for received chip.

Applying SP algorithm for the factor graph, message updating rules are described as

$$m_{n\to u}^{(t)}(b_u = \pm 1) = \sum_{\{b_{u'}:u' \neq u\}} \left\{ P\left(y_n | b_u {=} \pm 1, \{b_{u'} : u' \neq u\}\right) \prod_{u' \neq u} m_{u' \to n}^{(t-1)}(b_u) \right\}, \tag{2.55}$$

$$m_{u\to n}^{(t-1)}(b_u = \pm 1) = \prod_{n' \neq n} m_{n' \to u}^{(t-1)}(b_u = \pm 1), \tag{2.56}$$

and beliefs are computed as

$$q_u^{(t)}(b_u = \pm 1) = \alpha_u \prod_{n} m_{n \to u}^{(t)}(b_u = \pm 1), \tag{2.57}$$

where α_u is determined so that $q_u^{(t)}(b_u = +1) + q_u^{(t)}(b_u = -1) = 1$ holds.

Although the SP algorithm is defined as above, it requires $O(2^{U-1})$ operations to compute (2.55). Consequently, Tanaka and Okada reduced the computational complexity by approximating the algorithm. In their approximation, instead of using soft bits $n_{u\to n}(b_u)$ in (2.55), they use the following hard bits

$$\hat{b}_u^{(t)} = \mathrm{sgn}\left[q_u^{(t)}(b_u = +1) - q_u^{(t)}(b_u = -1)\right], \tag{2.58}$$

where $\mathrm{sgn}(a) = 1$ if $a > 0$ and else $\mathrm{sgn}(a) = -1$. Resulting approximation is described as

$$m_{n\to u}^{(t)}(b_u{=}\pm1) \approx p\left(r_n|b_u{=}\pm1, \left\{\hat{b}_{u'}^{(t-1)} : u'{\neq}u\right\}\right). \tag{2.59}$$

As a result, the following update rule for the temporary decision $\hat{b}_u^{(t)}$ of information bits at iteration stage t is derived [3]:

$$\hat{b}_u^{(t)} = \mathrm{sgn}\left[h_u - \sum_{u'\neq u} W_{uu'}\hat{b}_{u'}^{(t-1)}\right], \tag{2.60}$$

where

$$h_u = \sum_{n=1}^{N} s_{u,n}y_n, \qquad W_{uu'} = \sum_{n=1}^{N} s_{u,n}s_{u',n}, \tag{2.61}$$

h_u is the output of the matched filter for user u, and $W_{uu'}$ is the uu'-element of the cross correlation matrix $\boldsymbol{W}$ of the signature sequences.

In [3], it is indicated that the above update rule is the same rule as Parallel Interference Canceller (PIC) [4]. The computational complexity of PIC is $O(U^2)$.

2.7.3 Sum-Product Algorithm based Decoding Algorithm for the Binary Linear Codes over Memoryless Multiple-Access Channels

In [5] [6] [7], the authors presented frameworks for the design of LDPC codes and SP algorithm based decoding for multiple-access channels.

Here we describe the SP decoding algorithm for multiple-access channels.[6] First, we describe the SP algorithm for the subgraph corresponding to the code of user u. The algorithm takes as input the temporary approximate posterior probabilities of all users $\left\{\left(q_{u,n}^{(0)}, q_{u,n}^{(1)}\right)\right\}_{u=1,2,\cdots,U}$ and outputs the updated approximate posterior probability $\left(q_{u,n}^{(0)}, q_{u,n}^{(1)}\right)$ of user u.

SP algorithm for the subgraph corresponding to the code of user u

[6] In the papers [5] [6] [7], the authors don't describe the algorithm detail. Therefore the algorithm described here is an original one.

Initialization:
For $n = 1, 2, \cdots, N$,

$$e^{0}_{u,n} = \alpha \left(\sum_{(x_{u',n})_{u'\neq u}\in\{0,1\}^{U-1}} \Pr\left(y_n | x_{u,n} = 0, (x_{u',n})_{u'\neq u}\right) \prod_{u'\neq u} q^{(x_{u',n})}_{u',n} \right), \tag{2.62}$$

$$e^{1}_{u,n} = \alpha \left(\sum_{(x_{u',n})_{u\neq u}\in\{0,1\}^{U-1}} \Pr\left(y_n | x_{u,n} = 1, (x_{u',n})_{u'\neq u}\right) \prod_{u'\neq u} q^{(x_{u',n})}_{u',n} \right), \tag{2.63}$$

For all pairs (m, n) such that $H^{(u)}_{m,n} = 1$,

$$q^{0,(0)}_{mn,u} = e^{0}_{u,n}, \tag{2.64}$$

$$q^{1,(0)}_{mn,u} = e^{1}_{u,n}. \tag{2.65}$$

Factor node process:
For all pairs (m, n) such that $H^{(u)}_{m,n} = 1$,

$$r^{0,(t)}_{mn,u} = ((1 + \delta^{(t)}_{mn,u})/2), \tag{2.66}$$

$$r^{1,(t)}_{mn,u} = ((1 - \delta^{(t)}_{mn,u})/2), \tag{2.67}$$

where

$$\delta^{(t)}_{mn,u} = \prod_{n'\in\mathcal{N}_u(m)\setminus n} \left(q^{0,(t)}_{mn',u} - q^{1,(t)}_{mn',u} \right). \tag{2.68}$$

Variable node process:
For all pairs (m, n) such that $H^{(u)}_{m,n} = 1$,

$$q^{0,(t+1)}_{mn,u} = \alpha \left(e^{0}_{u,n} \prod_{m'\in\mathcal{M}_u(n)\setminus m} r^{0,(t)}_{m'n,u} \right), \tag{2.69}$$

$$q^{1,(t+1)}_{mn,u} = \alpha \left(e^{1}_{u,n} \prod_{m'\in\mathcal{M}_u(n)\setminus j} r^{1,(t)}_{m'n,u} \right). \tag{2.70}$$

where $\mathcal{M}_u(n) = \left\{ m : H^{(u)}_{m,n} = 1 \right\}$. The algorithm iteratively process the factor node process and variable node process for a predetermined number. The algorithm then outputs the approximate posterior probability as,

$$q^{0,(t)}_{u,n} = \alpha \left(e^{0}_{u,n} \prod_{m\in\mathcal{M}_u(n)} r^{0,(t)}_{mn,u} \right), \tag{2.71}$$

$$q^{1,(t)}_{u,n} = \alpha \left(e^{0}_{u,n} \prod_{m\in\mathcal{M}_u(n)} r^{1,(t)}_{mn,u} \right). \tag{2.72}$$

The SP algorithm for multiple-access channels is described as follows.

```
SP algorithm for multiple-access channels
Initialization:
For $u = 1, 2, \cdots, U$ and $n = 1, 2, \cdots, N$,
    $q_{u,n}^{0,(0)} = q_{u,n}^{1,(0)} = 1/2.$        (2.73)
Main loop:
  for $t = 1, 2, \cdots, T_1$ do
    for $u = 1, 2, \cdots, U$ do
      Implement the SP algorithm for the subgraph corresponding to the code of user $u$. The number of iterations is set to $T_2$.
    end for
  end for
Decoding process:
For all $u = 1, 2, \cdots, U$ and $n = 1, 2, \cdots, N$,
    $\hat{x}_{u,n} = \begin{cases} 0 & q_{u,n}^{0,(t)} \geq q_{u,n}^{1,(t)} \\ 1 & q_{u,n}^{0,(t)} < q_{u,n}^{1,(t)} \end{cases}$        (2.74)
```

2.8 Linear Programming based Decoding Algorithm for Binary Linear Codes over Memoryless Channels

Recently, Feldman et al. proposed the linear programming based decoding algorithm for binary linear codes over memoryless channels [8]. In this section we described the algorithm. Remember that the ML decoding problem is defined as

$$\arg\max_{\boldsymbol{x}\in\mathcal{C}} p(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{x}\in\mathcal{C}} \sum_{n=1}^{N} \ln p(y_n|x_n). \tag{2.75}$$

This problem is equivalent to the following linear programming problem

$$\begin{array}{ll} \text{minimize} & \boldsymbol{\gamma}^T \boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \in \text{conv}(\mathcal{C}), \end{array} \tag{2.76}$$

where $\text{conv}(\mathcal{C})$ is the convex hull of all codewords in $\mathcal{C}$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_N)$ is the vector of log-likelihood ratios defined as

$$\gamma_n = \ln\left(\frac{p(y_n|x_n = 0)}{p(y_n|x_n = 1)}\right). \tag{2.77}$$

Given that (2.76) is a linear programming (LP) problem, it is solvable in polynomial time on the number of variables and constraints, however, generally the number of constraints is exponential in the code length n. As an approximation to ML decoding , Feldman et al. proposed a relaxed version of the problem [8]. They consider the convex hull of the local codewords defined by each row of the parity check matrix. The intersection of them defines the polytope $\mathcal{P}$. The LP decoder minimizes $\boldsymbol{\gamma}^T \boldsymbol{x}$ over the polytope $\mathcal{P}$. The LP decoder has the property that if it outputs an integer solution, it is guaranteed to be an ML codeword. This property is called ML certificate property.

We provide an explicit inequality description of the relaxed polytope $\mathcal{P}$ which is explained in [8]. For every check $m = 1, 2, \cdots, M$, every configuration of the set of neighboring variables $\mathcal{N}(m)$ must satisfy the following constraints for all subsets $S \subseteq \mathcal{N}(m)$, $|S|$ is odd,

$$\sum_{n \in S} x_n - \sum_{n \in \mathcal{N}(m) \backslash S} x_n \leq |S| - 1. \tag{2.78}$$

We also need to add a set of $2N$ box inequalities, i.e.,

$$0 \leq x_n \leq 1 \quad n = 1, 2, \cdots, N. \tag{2.79}$$

As a result, relaxed LP is described as follows:

$$\begin{array}{ll} \text{minimize} & \boldsymbol{\gamma}^T \boldsymbol{x} \\ \text{subject to} & 0 \leq x_n \leq 1,\ n = 1, 2, \cdots, N, \\ & m = 1, 2, \cdots, M,\ \forall S \text{ s.t. } S \subseteq \mathcal{N}(m),\ |S| \text{ is odd}, \\ & \sum_{n \in S} x_n - \sum_{n \in \mathcal{N}(m) \backslash S} x_n \leq |S| - 1. \end{array} \tag{2.80}$$

The number of constraints in the problem is reduced to $\sum_{m=1}^{M} 2^{|\mathcal{N}(m)|-1} + 2N$.

2.9 Contribution of the Thesis

In chapter we review the problem setting of the probabilistic inference problems and SP algorithm which (approximately) solve the problem efficiently. We also have seen that various channel decoding problems can be considered as instances of the probabilistic inference problems. As a consequence, we can consider to apply the SP algorithm to those decoding problems. In fact, SP algorithm effectively works for the decoding problem of binary linear codes over memoryless channels. We can apply the SP algorithm for other problems that are expressed by the factor graph. On the other hand, the LP-based decoding algorithm for the decoding problem of binary linear codes over memoryless channels is recently developed. It is known that the LP-based decoding algorithm has many attractive features. Then the question is prompted. "Can we extend the LP-based inference algorithm for general probabilistic inference problems ?". In chapter 3, we show that the answer of the question is "Yes". We will provide the LP-based inference algorithm for general factor graphs. Based on the work, we can apply the LP-based inference algorithms for various decoding problems which include decoding problem on DS-CDMA systems and decoding problem of binary linear codes over memoryless multiple-access channels.

Compared to the SP algorithm, the LP-based inference algorithm have many desirable properties. For example, the output of the LP-based inference algorithm is integer, it is guaranteed to be optimal.

The both of the SP algorithm and LP-based inference algorithm have the complexity exponentially proportional to the number of variable nodes connected to factor node. Therefore those algorithm are impractical in many cases. For example, we have seen that we can't apply the SP algorithm for the multiuser detection problems for DS-CDMA systems. In chapter 4, we can reduce the computational complexity of those algorithms for some cases by converting the factor graph structure. We will show that we can apply the SP algorithm for the multiuser detection problems for DS-CDMA systems. We also present the reduced complexity LP-based decoding algorithms for the decoding problem of binary linear codes over the Gaussian multiple-access channels.

The LP-based inference algorithm have the attractive feature that if the output of the algorithm is an integer, it is guaranteed to be optimal. In other words, if the output of the algorithm is not an integer, we can prove the output is not optimal. By utilizing the property, we can propose the improved algorithm. In chapter 5, we propose a new decoding algorithm for binary linear codes over memoryless channels. The proposed algorithm is based on the branch-and-cut algorithm, which is originally proposed in the theory of combinatorial optimization.

Chapter 3

Linear Programming based Inference Algorithm for General Factor Graph

3.1 Introduction

In this chapter, we present the linear programming (LP) -based inference algorithm for general factor graph. In chapter 2, we saw that the decoding problem of the binary linear code can be considered as an example of the probabilistic inference problem. The Sum-Product (SP) algorithm is the algorithm defined on the factor graph and it outputs the optimum solution when the factor graph has no cycles. When the factor graph has any cycles, the SP algorithm does not output the optimum solution, however, the effectiveness of the algorithm for that situation is examined through numerical simulations or analyses.

On the other hand the LP decoding is also an effective decoding algorithm. There are many important connections between the LP decoding and the sum-product decoding. The LP decoding has been attracting much attention from its mathematical tractability.

If we consider the decoding problem as an example of the probabilistic inference problem, the factor graph corresponds to the problem has some specific structures. First, we can see that the functions in the factor graph can be classified into two classes, indicator functions (which has the range $\{0,1\}$ and it indicates membership is in a set or not) and non-indicator functions (functions other than the indicator function). Second, each non-indicator function is connected to only one variable node. The factor graph corresponds to the general probabilistic inference problems possibly has non-indicator functions which are connected to more than one variable node (We call such functions *multi-degree non-indicator function*). For example, if we describe the factor graph corresponds to the decoding problem of binary linear code for multiple-access channels, the graph has such functions. For such problems, we couldn't obtain the inference algorithm based on the LP in the same manner as the derivation of LP decoding.

In this chapter, we develop the LP-based inference algorithm for the factor graph which has the multi-degree non-indicator functions. The basic idea is to reduce the problem to the inference problem that the factor graph corresponds to the problem does not have the multi-degree non-indicator functions connected to more than one variable node.

The rest of the chapter is organized as follows. In section 3.2, we reformulate the probabilistic inference problem. In section 3.3, we demonstrate how the LP-based inference algorithm is derived. Then we develop the LP-based inference algorithm in section 3.4. We give a conclusion in section 3.6.

3.2 Reformulation of the Probabilistic Inference Problem

In this section, we reformulate the probabilistic inference problem. We assume that the functions $f_a, a \in \mathcal{A}$ in the factorization (2.4) are classified into two classes. One is the *indicator function* and the another is the *non-indicator function*. Describing the indicator functions in $f_a, a \in \mathcal{A}$ as $f_{I_j}, j \in \mathcal{J}$ and the non-indicator functions as $f_{R_l}, l \in \mathcal{L}$, the factorization (2.4) is reduced to

$$\frac{1}{Z} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}) = \frac{1}{Z} \prod_{j \in \mathcal{J}} f_{I_j}(\boldsymbol{x}_{I_j}) \prod_{l \in \mathcal{L}} f_{R_l}(\boldsymbol{x}_{R_l}) \tag{3.1}$$

For each $j \in \mathcal{J}$ and $l \in \mathcal{L}$, let $\mathcal{N}(j) = \left\{j_1, j_2, \cdots, j_{|\mathcal{N}(j)|}\right\}$ and $\mathcal{N}(l) = \left\{l_1, l_2, \cdots, l_{|\mathcal{N}(l)|}\right\}$ denote the indices of the membership of the function f_{I_j} and f_{R_l}, respectively. More specifically, $\boldsymbol{x}_{I_j}$ and $\boldsymbol{x}_{R_l}$ are defined as $\boldsymbol{x}_{I_j} = (x_{j_1}, x_{j_2}, \cdots, x_{j_{|\mathcal{N}(j)|}})$ and $\boldsymbol{x}_{R_l} = (x_{l_1}, x_{l_2}, \cdots, x_{l_{|\mathcal{N}(l)|}})$, respectively. Since we deal with the probability distribution function, we can assume that the range of the non-indicator functions $f_{R_l}, l \in \mathcal{L}$ are nonnegative real values. We define the function $g(\boldsymbol{x})$ as

$$g(\boldsymbol{x}) = \prod_{j \in \mathcal{J}} f_{I_j}(\boldsymbol{x}_{I_j}) \prod_{l \in \mathcal{L}} f_{R_l}(\boldsymbol{x}_l). \tag{3.2}$$

3.3 Linear Programming Inference for the Factor Graph without Multi-degree Non-Indicator Function Nodes

Here we consider the case that each non-indicator function in (3.2) is connected only one variable node. We call the functions which are connected more than two variable node *multi-degree* function. Such problem is arising in the decoding problem of binary or non-binary linear codes for single user memoryless channel [8][9].

3.3.1 Transformation to the Linear Programming Problem

When the argument of the non-indicator function is one, $g(\boldsymbol{x})$ can be described as

$$g(\boldsymbol{x}) = \prod_{j \in \mathcal{J}} f_{I_j}(\boldsymbol{x}_{I_j}) \prod_{n \in \mathcal{L}} f_{R_n}(x_n), \tag{3.3}$$

where $\mathcal{L} \subseteq \{1, 2, \cdots, N\}$. Then the problem to find $\boldsymbol{x} \in \mathcal{X}^N$ which maximizes g is translated as

$$\begin{array}{ll} \text{minimize} & \sum_{n \in \mathcal{L}} - \ln f_{R_n}(x_n) \\ \text{subject to} & f_{I_j}(\boldsymbol{x}_{I_j}) = 1, \quad \forall j \in \mathcal{J}. \end{array} \tag{3.4}$$

If there is no $\boldsymbol{x}$ which satisfy the constraints in the above optimization problem, the value of g is 0 for all $\boldsymbol{x} \in \mathcal{X}$.

Then we show that how the above problem is reduce to the LP problem. For each x_n, we define the mapping $\{\phi_{n:\alpha}\}_{\alpha \in \mathcal{X}_n}$ as

$$\phi_{n:\alpha}(x_n) = \begin{cases} 1 & \text{if } x_n = \alpha \\ 0 & \text{otherwise} \end{cases} \quad n = 1, 2, \cdots, N, \forall \alpha \in \mathcal{X}_n. \tag{3.5}$$

We also define the mapping $\phi_n : \mathcal{X}_n \to \{0,1\}^{|\mathcal{X}_n|}$ and $\boldsymbol{\Phi} : \mathcal{X} \to \{0,1\}^d$ as $\phi_n(x_n) = (\phi_{n:\alpha}(x_n))_{\alpha \in \mathcal{X}_n}$ and $\boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_n(x_n))_{n=1,2,\cdots,N}$, where $d = \sum_{n=1}^{N} |\mathcal{X}_n|$. Furthermore we define $(\lambda_{n:\alpha})_{n \in \mathcal{L},\ \alpha \in \mathcal{X}_n}$ as

$$\lambda_{n:\alpha} = -\ln f_{R_n}(\alpha),\ n = 1, 2, \cdots, N,\ \alpha \in \mathcal{X}_n. \tag{3.6}$$

Then the optimization problem in (3.4) is equivalent to the following problem.

$$\begin{array}{ll} \text{minimize} & \sum_{n \in \mathcal{L}} \sum_{\alpha \in \mathcal{X}_n} \lambda_{n:\alpha} \phi_{n:\alpha}(x_n) \\ \text{subject to} & f_{I_j}(\boldsymbol{x}_{I_j}) = 1, \quad j \in \mathcal{J}. \end{array} \tag{3.7}$$

We define the variable $\boldsymbol{\tau}_n$ and $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau}_n = \left(\tau_{n:1}, \tau_{n:2}, \cdots, \tau_{n:|\mathcal{X}_n|}\right) \in \{0,1\}^{|\mathcal{X}_n|} \ \ n = 1, 2, \cdots, N, \tag{3.8}$$

$$\boldsymbol{\tau} = (\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \cdots, \boldsymbol{\tau}_N). \tag{3.9}$$

We also define the set $\mathcal{P}_j, j \in \mathcal{J}$ and $\mathcal{P}$ as

$$\mathcal{P}_j = \left\{ (\phi_n(x_n))_{n \in \mathcal{N}(j)} : f_{I_j}(\boldsymbol{x}_{I_j}) = 1 \right\}, \tag{3.10}$$

$$\mathcal{P} = \left\{ \boldsymbol{\tau} \in \{0,1\}^d : \mathrm{Proj}_j(\boldsymbol{\tau}) \in \mathcal{P}_j,\ \forall j \in \mathcal{J} \right\}, \tag{3.11}$$

where $\mathrm{Proj}_j(\cdot)$ is a projection to the coordinate indexed by $\mathcal{N}_j$, i.e., $\mathrm{Proj}_j(\boldsymbol{\tau}) = (\boldsymbol{\tau}_n)_{n \in \mathcal{N}_j}$. We further define the polytope $\mathcal{Q}$ as

$$\mathcal{Q} = \mathrm{conv}(\mathcal{P}), \tag{3.12}$$

where $\mathrm{conv}(\cdot)$ denotes the operation to take the convex hull of a set. When $\boldsymbol{\tau}$ is a vertex of the polytope $\mathcal{Q}$, the variables $\boldsymbol{x}$ given by $\boldsymbol{\Phi}^{-1}(\boldsymbol{\tau})$ satisfies all constraints in (3.7), where $\boldsymbol{\Phi}^{-1}$ is the inverse mapping of $\boldsymbol{\Phi}$. Then to solve the optimization problem (3.4) is reduced to solve the following optimization problem;

$$\begin{array}{ll} \text{minimize} & \sum_{n \in \mathcal{L}} \sum_{\alpha \in \mathcal{X}_n} \lambda_{n:\alpha} \tau_{n:\alpha} \\ \text{subject to} & \boldsymbol{\tau} \in \mathcal{Q} \end{array} \tag{3.13}$$

The equalities or inequalities which describe the polytope $\mathcal{P}$ are depend on the form of the indicator functions $f_{I_j}, j \in \mathcal{J}$. Let $\boldsymbol{\tau}^*$ be the optimum solution of the problem (3.13), then the optimum solution of the original optimization problem (3.4) is given by $\boldsymbol{\Phi}^{-1}(\boldsymbol{\tau}^*)$.

3.3.2 Derivation of the Relaxed Problem

In general, the number of hyperplanes which express the polytope $\mathcal{Q}$ is the exponentially proportional to N. Therefore it is impractical to solve the LP problem (3.13). Then we define the following polytope,

$$\tilde{\mathcal{Q}} = \left\{ \boldsymbol{\tau} \in [0,1]^d : \mathrm{Proj}_j(\boldsymbol{\tau}) \in \mathrm{conv}(\mathcal{P}_j),\ \forall j \in \mathcal{J} \right\}. \tag{3.14}$$

Then the following theorem holds.

Theorem 3.1

$$\mathcal{Q} \subseteq \tilde{\mathcal{Q}} \tag{3.15}$$

$$\mathcal{Q} \cap \{0,1\}^d = \tilde{\mathcal{Q}} \cap \{0,1\}^d \tag{3.16}$$

Proof: First we prove the first part of the theorem. Let $\boldsymbol{\tau} \in \mathcal{Q}$, then $\boldsymbol{\tau}$ can be described as the convex combination of some vectors $\boldsymbol{\tau}^{(c)} \in \mathcal{P}, c = 1, 2, \cdots, C$, that is,

$$\boldsymbol{\tau} = \omega_1 \boldsymbol{\tau}^{(1)} + \omega_2 \boldsymbol{\tau}^{(2)} + \cdots + \omega_C \boldsymbol{\tau}^{(C)} \tag{3.17}$$

where $\{\omega_c\}_{c=1,2,\cdots,C}$ are coefficients of the convex combination and they satisfy $\sum_{c=1}^{C} \omega_c = 1$ and $0 \le \omega_c \le 1, c = 1, 2, \cdots, C$. From the definition of $\mathcal{P}$, if $\boldsymbol{\tau}^{(c)} \in \mathcal{P}$ then it satisfies $\mathrm{Proj}_j(\boldsymbol{\tau}^{(c)}) \in \mathcal{P}_j$ for all $j \in \mathcal{J}$. Then we have

$$\begin{aligned} \mathrm{Proj}_j(\boldsymbol{\tau}) &= \mathrm{Proj}_j \left(\sum_{c=1}^{C} \omega_c \boldsymbol{\tau}^{(c)} \right) \\ &= \sum_{c=1}^{C} \omega_c \mathrm{Proj}_j(\boldsymbol{\tau}^{(c)}) \in \mathrm{conv}(\mathcal{P}_j). \end{aligned} \tag{3.18}$$

Above relation is satisfied for all $j \in \mathcal{J}$, and hence $\boldsymbol{\tau} \in \tilde{\mathcal{Q}}$.

Next we prove the second part of the theorem. From the first part of the theorem, we have $\mathcal{Q} \bigcap \{0,1\}^d \subseteq \tilde{\mathcal{Q}} \bigcap \{0,1\}^d$, we then prove $\mathcal{Q} \bigcap \{0,1\}^d \supseteq \tilde{\mathcal{Q}} \bigcap \{0,1\}^d$. We assume that $\boldsymbol{\tau} \in \tilde{\mathcal{Q}} \bigcap \{0,1\}^d$. From the definition of $\tilde{\mathcal{Q}}$, it holds $\mathrm{Proj}_j(\boldsymbol{\tau}) \in \mathrm{conv}(\mathcal{P}_j), \forall j \in \mathcal{J}$. However, it satisfies that $\mathrm{conv}(\mathcal{P}_j) \bigcap \{0,1\}^d = \mathcal{P}_j$ for all $j \in \mathcal{J}$, and therefore $\mathrm{Proj}_j(\boldsymbol{\tau}) \in \mathcal{P}_j,\ \forall j \in \mathcal{J}$ and $\boldsymbol{\tau} \in \mathcal{P} \subseteq \mathcal{Q}$. ■

According to the theorem, if the optimum solution $\boldsymbol{\tau}$ of the following problem is an integer, then $\boldsymbol{x}$ obtained from the inverse mapping $\boldsymbol{x} = \boldsymbol{\Phi}^{-1}(\boldsymbol{\tau})$ is also the optimum solution of the optimization problem (3.4).

$$\begin{array}{ll} \text{minimize} & \sum_{n \in \mathcal{L}} \sum_{\alpha \in \mathcal{X}_n} \lambda_{n:\alpha} \tau_{n:\alpha} \\ \text{subject to} & \boldsymbol{\tau} \in \tilde{\mathcal{Q}} \end{array} \tag{3.19}$$

The expression of $\tilde{\mathcal{Q}}$ depends on the functions $f_{I_j}, j \in \mathcal{J}$ as in the case for $\mathcal{Q}$.

Example 3.1 (Decoding of the binary linear codes [8]) Let H be the parity check matrix $\{0,1\}^{M\times N}$ and $\mathcal{C}$ be a code with parity check matrix H, that is,

$$\mathcal{C} = \left\{ \boldsymbol{x} \in \{0,1\}^N : H^T\boldsymbol{x} = \mathbf{0} \bmod 2 \right\}. \tag{3.20}$$

Assume that the codeword $\boldsymbol{c} = (c_1, c_2, \cdots, c_N)$ is transmitted through the channel and the decoder receives $\boldsymbol{y} = (y_1, y_2, \cdots, y_N)$. Furthermore the channel is modeled as $p(\boldsymbol{y}|\boldsymbol{x})$ and it is memoryless, i.e.,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{n=1}^{N} p(y_n|x_n). \tag{3.21}$$

Then the maximum likelihood decoding problem is defined as

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}\in\mathcal{C}} p(\boldsymbol{y}|\boldsymbol{x}) && (3.22)\\ &= \arg\max_{\boldsymbol{x}\in\{0,1\}^N} \prod_{m=1}^{M} f_{I_m}(\boldsymbol{x}_{I_m}) \prod_{n=1}^{N} f_{R_n}(x_n), && (3.23)\end{aligned}$$

where f_{I_m} and f_{R_n} are defined as

$$f_{I_m}(\boldsymbol{x}_{I_m}) = \begin{cases} 1 & \text{if } \sum_{n\in\mathcal{N}_c(m)} x_n = 0 \bmod 2 \\ 0 & \text{otherwise} \end{cases} \qquad m = 1, 2, \cdots, M \tag{3.24}$$

$$f_{R_n}(x_n) = p(y_n|x_n), \quad n = 1, 2, \cdots, N, \tag{3.25}$$

If we apply the LP relaxation (3.19) to this problem, the optimization variable $\boldsymbol{\tau}$ consists of $(\tau_{n:0}, \tau_{n:1})$ for each $n = 1, 2, \cdots, N$. We obtain the following LP:

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} \sum_{\alpha\in\{0,1\}} \lambda_{n:\alpha}\tau_{n:\alpha} \\ \text{subject to} & \boldsymbol{\tau} \in \tilde{\mathcal{Q}} \end{array}, \tag{3.26}$$

We then derive the explicit form of the relaxed polytope $\tilde{\mathcal{Q}}$. The relax polytope is defined as

$$\tilde{\mathcal{Q}} = \left\{ \boldsymbol{\tau} \in [0,1]^{2N} : \text{Proj}_m(\boldsymbol{\tau}) \in \text{conv}(\mathcal{P}_m),\ \forall m = 1, 2, \cdots, M \right\} \tag{3.27}$$

$$\mathcal{P}_m = \left\{ (\phi_n(x_n))_{n\in\mathcal{N}(m)} : f_{I_m}(\boldsymbol{x}_{I_m}) = 1 \right\} \tag{3.28}$$

Let $\boldsymbol{\beta}$ denotes the elements in $\mathcal{P}_m$, where $\boldsymbol{\beta} = (\boldsymbol{\beta}_n)_{n\in\mathcal{N}(m)}$ and $\boldsymbol{\beta}_n = (\beta_{n:0}, \beta_{n:1})$. The polytope conv$(\mathcal{P}_m)$ is the set of the convex combinations of all $\boldsymbol{\beta} \in \mathcal{P}_m$. Let $\omega_{m,\boldsymbol{\beta}}$ be the convex combination coefficient for $\boldsymbol{\beta}$, then the condition $\text{Proj}_m(\boldsymbol{\tau}) \in \text{conv}(\mathcal{P}_m)$ is equivalent to the condition that $\text{Proj}_m(\boldsymbol{\tau})$ satisfies followings.

$$\begin{cases} \tau_{n:1} = \sum_{\boldsymbol{\beta}\in\mathcal{P}_m:\beta_{n:1}=1} \omega_{m,\boldsymbol{\beta}} \\ \tau_{n:0} = \sum_{\boldsymbol{\beta}\in\mathcal{P}_m:\beta_{n:0}=1} \omega_{m,\boldsymbol{\beta}} \end{cases} \qquad \forall n \in \mathcal{N}(m), \tag{3.29}$$

$$\sum_{\boldsymbol{\beta}\in\mathcal{P}_m} \omega_{m,\boldsymbol{\beta}} = 1, \tag{3.30}$$

$$0 \le \omega_{m,\boldsymbol{\beta}} \le 1 \quad \forall \boldsymbol{\beta} \in \mathcal{P}_m. \tag{3.31}$$

We can reduce the number of variables by susbstituting the constraints $\tau_{n:0} = 1 - \tau_{n:1}$. It is also possible to remove the variables $\omega_{m,\boldsymbol{\beta}}$ by Fourier-Motzkin elimination [10], so as to obtain an equivalent LP that is described only in terms of the vectors $(\tau_{1:1}, \tau_{2:1}, \cdots, \tau_{N:1})$. As the result of such operations, we obtain the following LP:

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} (\lambda_{n:1} - \lambda_{n:0})\, \tau_{n:1} \\ \text{subject to} & m = 1, 2, \cdots, M,\ \forall S \text{ s.t. } S \subseteq \mathcal{N}(m),\ |S| \text{ is odd.} \\ & \sum_{n \in S} \tau_{n:1} - \sum_{n \in \mathcal{N}(m) \setminus S} \tau_{n:1} \leq |S| - 1 \\ & 0 \leq \tau_{n:1} \leq 1, \quad n = 1, 2, \cdots, N. \end{array} \tag{3.32}$$

Given that $\tau_{n:1} = 1$ if $x_n = 1$ and $\tau_{n:1} = 0$ if $x_n = 0$, we can replace $\tau_{n:1}$ by x_n. Then the LP is described as,

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} (\lambda_{n:1} - \lambda_{n:0})\, x_n \\ \text{subject to} & m = 1, 2, \cdots, M,\ \forall S \text{ s.t. } S \subseteq \mathcal{N}(m),\ |S| \text{ is odd.} \\ & \sum_{n \in S} x_n - \sum_{n \in \mathcal{N}(m) \setminus S} x_n \leq |S| - 1 \\ & 0 \leq x_n \leq 1, \quad n = 1, 2, \cdots, N \end{array} . \tag{3.33}$$

This LP is equivalent to the formulation which is derived in [8]. (Which is also described in chapter 2 of the thesis.)

3.4 Linear Programming Inference for the Factor Graph with Multi-degree Non-indicator Function Nodes

Here we develop the LP-based inference algorithm for the factor graph which has the multi-degree non-indicator function nodes. The basic idea is to reduce the problem to solve the maximization problem corresponds to the factor graph without multi-degree non-indicator function nodes.

3.4.1 Derivation of the Linear Programming Problem

For each $\boldsymbol{x}_{R_l}, l \in \mathcal{L}$, we define the set $\mathcal{W}_l = \mathcal{X}_{l_1} \times \mathcal{X}_{l_2} \times \cdots \times \mathcal{X}_{l_{|\mathcal{N}(l)|}}$ and variable $\boldsymbol{w}_l$ which takes value in $\mathcal{W}_l$. We also define the following function for $\boldsymbol{w}_l$ and $\boldsymbol{x}_{R_l}$,

$$f_{I'_l}(\boldsymbol{w}_l, \boldsymbol{x}_{R_l}) = \begin{cases} 1 & \text{if } \boldsymbol{w}_l = \boldsymbol{x}_{R_l} \\ 0 & \text{otherwise} \end{cases} \quad l \in \mathcal{L}. \tag{3.34}$$

Furthermore we also define $\boldsymbol{w} = (\boldsymbol{w}_l)_{l \in \mathcal{L}} \in \mathcal{W}$, where $\mathcal{W} = \times_{l \in \mathcal{L}} \mathcal{W}_l$, and the function of $\boldsymbol{x}$ and $\boldsymbol{w}$ as

$$g'(\boldsymbol{x}, \boldsymbol{w}) = \prod_{j \in \mathcal{J}} f_{I_j}(\boldsymbol{x}_{I_j}) \prod_{l \in \mathcal{L}} f_{I'_l}(\boldsymbol{w}_l, \boldsymbol{x}_{R_l}) \prod_{l \in \mathcal{L}} f_{R_l}(\boldsymbol{w}_l). \tag{3.35}$$

Then there is a relation between g and g'.

Theorem 3.2 Let $(\boldsymbol{x}^*, \boldsymbol{w}^*)$ maximizes the function g', then $\boldsymbol{x}^*$ maximizes the function g.

Proof: Assume that there exists $\boldsymbol{x}'$ such that $\boldsymbol{x}' \neq \boldsymbol{x}^*$ and $g(\boldsymbol{x}') > g(\boldsymbol{x}^*)$. We set $\boldsymbol{w}'_l$ as

$$\boldsymbol{w}'_l = (x'_{l_1}, x'_{l_2}, \cdots, x'_{l_{|\mathcal{N}(l)|}}). \tag{3.36}$$

Then it holds that $f'(\boldsymbol{x}_{R_l}, \boldsymbol{w}_l) = 1$ for all $l \in \mathcal{L}$. Then it is satisfied that

$$g'(\boldsymbol{x}', \boldsymbol{w}') = g(\boldsymbol{x}') > g(\boldsymbol{x}^*) = g'(\boldsymbol{x}^*, \boldsymbol{w}^*), \tag{3.37}$$

however, it contradicts to the assumption that $(\boldsymbol{x}^*, \boldsymbol{w}^*)$ maximizes g'. ■

According to the above theorem, if we want to solve the maximization problem for g, it is sufficient to solve the maximization problem for g'. Furthermore the non-indicator functions in g' are only $\{f_{R_l}\}_{l\in\mathcal{L}}$ and we can treat them as a functions connected to one variable node $\boldsymbol{w}_l$. Therefore we can derive the LP in the similar manner of the previous section. The optimization problem that we should solve is described as

$$\begin{array}{ll} \text{minimize} & -\sum_{l\in\mathcal{L}} \ln f_{R_l}(\boldsymbol{w}_l) \\ \text{subject to} & f_{I_j}(\boldsymbol{x}_{I_j}) = 1,\ \forall j \in \mathcal{J} \\ & f_{I'_l}(\boldsymbol{w}_l, \boldsymbol{x}_{R_l}) = 1,\ \forall l \in \mathcal{L}. \end{array} \tag{3.38}$$

Here, we derive the LP formulation of the problem as in the previous section. For each x_n we define the mapping ϕ_n as defined in the previous section. For each $\boldsymbol{w}_l$, we also define the mapping $\{\phi_{l:\boldsymbol{\beta}}\}_{\boldsymbol{\beta}}$ as

$$\phi_{l:\boldsymbol{\beta}}(\boldsymbol{w}_l) = \begin{cases} 1 & \text{if } \boldsymbol{w}_l = \boldsymbol{\beta} \\ 0 & \text{otherwise} \end{cases} \qquad l \in \mathcal{L}, \boldsymbol{\beta} \in \mathcal{W}_l \tag{3.39}$$

We define $\phi_l(\boldsymbol{w}_l) = (\phi_{l:\boldsymbol{\beta}}(\boldsymbol{\beta}))_{\boldsymbol{\beta}\in\mathcal{W}_l}$ and $\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{w}) = \Big((\phi_n(x_n))_{n=1,2,\cdots,N}, (\phi_l(\boldsymbol{w}_l))_{l\in\mathcal{L}}\Big)$. We also define $(\lambda_{l:\boldsymbol{\beta}})_{\boldsymbol{\beta}\in\mathcal{W}_l}$ as

$$\lambda_{l:\boldsymbol{\beta}} = -\ln f_{R_l}(\boldsymbol{\beta}), \quad \boldsymbol{\beta} \in \mathcal{W}_l \tag{3.40}$$

The optimization problem in (3.38) is reduced to the following problem.

$$\begin{array}{ll} \text{minimize} & \sum_{l\in\mathcal{L}} \sum_{\boldsymbol{\beta}} \lambda_{l:\boldsymbol{\beta}} \phi_{l:\boldsymbol{\beta}}(\boldsymbol{w}_l) \\ \text{subject to} & f_{I_j}(\boldsymbol{x}_{I_j}) = 1,\ \forall j \in \mathcal{J} \\ & f_{I'_l}(\boldsymbol{w}_l, \boldsymbol{x}_{R_l}) = 1,\ \forall l \in \mathcal{L} \end{array} \tag{3.41}$$

We define the variables $\boldsymbol{\tau}_l$ and $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau}_l = (\tau_{l:1}, \tau_{l:2}, \cdots, \tau_{l:|\mathcal{W}_l|}) \in \{0,1\}^{|\mathcal{W}_l|},\ l \in \mathcal{L} \tag{3.42}$$

$$\boldsymbol{\tau} = \Big((\boldsymbol{\tau}_n)_{n=1,2,\cdots,N}, (\boldsymbol{\tau}_l)_{l\in\mathcal{L}}\Big) \tag{3.43}$$

The length of variable $\boldsymbol{\tau}$ is $d = \sum_{n=1}^{N} |\mathcal{X}_n| + \sum_{l\in\mathcal{L}} |\mathcal{W}_l|$. For all $j \in \mathcal{J}, l \in \mathcal{L}$, we define $\mathcal{P}_j$ and $\mathcal{P}_l$ as

$$\mathcal{P}_j = \Big\{(\phi_n(x_n))_{n\in\mathcal{N}(j)} : f_{I_j}(\boldsymbol{x}_{I_j}) = 1\Big\} \tag{3.44}$$

$$\mathcal{P}_l = \Big\{\Big(\phi_l(\boldsymbol{w}_l), (\phi_n(x_n))_{n\in\mathcal{N}(l)}\Big) : f_{I_l}(\boldsymbol{w}_l, \boldsymbol{x}_{R_l}) = 1\Big\} \tag{3.45}$$

We also define the polytope $\mathcal{P}'$ as

$$\mathcal{P}' = \left\{\boldsymbol{\tau} \in \{0,1\}^d : \mathrm{Proj}_j(\boldsymbol{\tau}) \in \mathcal{P}_j,\ \forall j \in \mathcal{J},\ \mathrm{Proj}_l(\boldsymbol{\tau}) \in \mathcal{P}_l,\ \forall l \in \mathcal{L}\right\} \tag{3.46}$$

where $\mathrm{Proj}_l(\boldsymbol{\tau}) = \left(\boldsymbol{\tau}_l, (\boldsymbol{\tau}_n)_{n\in\mathcal{N}(l)}\right)$. Then the linear programming problem corresponds to the maximization problem in (3.38) is described as

$$\begin{array}{ll} \text{minimize} & \sum_{l\in\mathcal{L}}\sum_{\boldsymbol{\beta}\in\mathcal{W}_l} \lambda_{l:\boldsymbol{\beta}}\tau_{l:\boldsymbol{\beta}} \\ \text{subject to} & \boldsymbol{\tau} \in \mathcal{Q}', \end{array} \tag{3.47}$$

where $\mathcal{Q}' = \mathrm{conv}(\mathcal{P}')$.

The relaxed polytope is defined as

$$\tilde{\mathcal{Q}}' = \left\{\boldsymbol{\tau} \in [0,1]^d : \mathrm{Proj}_j(\boldsymbol{\tau}) \in \mathrm{conv}(\mathcal{P}_j),\ \forall j \in \mathcal{J},\ \mathrm{Proj}_l(\boldsymbol{\tau}) \in \mathrm{conv}(\mathcal{P}_l),\ \forall l \in \mathcal{L}\right\} \tag{3.48}$$

and the relaxed problem is

$$\begin{array}{ll} \text{minimize} & \sum_{l\in\mathcal{L}}\sum_{\boldsymbol{\beta}\in\mathcal{W}_l} \lambda_{l:\boldsymbol{\beta}}\tau_{l:\boldsymbol{\beta}} \\ \text{subject to} & \boldsymbol{\tau} \in \tilde{\mathcal{Q}}'. \end{array} \tag{3.49}$$

As the consequence of the theorem 3.1, the following lemma is holds.

Lemma 3.1

$$\mathcal{Q}' \subseteq \tilde{\mathcal{Q}}' \tag{3.50}$$

$$\mathcal{Q}' \cap \{0,1\}^{d'} = \tilde{\mathcal{Q}}' \cap \{0,1\}^{d'} \tag{3.51}$$

Proof Direct consequence of the theorem 3.1. ■

According to the lemma, the optimum solution $(\boldsymbol{x}, \boldsymbol{w})$ which maximizes the function g' is obtained from the inverse mapping $\boldsymbol{\Phi}^{-1}$ of $\boldsymbol{\tau}$ when $\boldsymbol{\tau}$ is an integer.

3.5 Application to the Decoding Problem of Binary Linear Codes over Multiple-Access Channel

3.5.1 Multiple-Access Channel Model and Decoding Problem

Then the ML decoding problem is defined as

$$(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_U) = \arg\max_{\boldsymbol{x}_1\in\mathcal{C}_1, \boldsymbol{x}_2\in\mathcal{C}_2, \cdots, \boldsymbol{x}_K\in\mathcal{C}_U} p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U). \tag{3.52}$$

If the prior distributions of the codewords are uniform, the ML decoding rule is optimal with respect to the following decoding error probability

$$P_e = \Pr\left\{(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_U) \neq (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U)\right\}. \tag{3.53}$$

For the U of binary N-length vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U$, we define the binary vector $\boldsymbol{x}$ with length UN as $\boldsymbol{x} = (\boldsymbol{x}_1 \; \boldsymbol{x}_2 \; \cdots \; \boldsymbol{x}_U)$. Then the ML decoding problem defined in (3.52) is equivalent to

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \bar{\mathcal{C}}} p(\boldsymbol{y}|\boldsymbol{x}). \tag{3.54}$$

where,

$$\mathcal{C} = \{\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) \; : \; \boldsymbol{x}_1 \in \mathcal{C}_1, \boldsymbol{x}_2 \in \mathcal{C}_2, \cdots, \boldsymbol{x}_U \in \mathcal{C}_U\}. \tag{3.55}$$

We define the $\boldsymbol{x}_{R_n}$ as $\boldsymbol{x}_{R_n} = (x_{1,n}, x_{2,n}, \cdots, x_{U,n}), \; n = 1, 2, \cdots, N$. This is the set of codeword symbols at time n when the $\boldsymbol{x}$ is the set of transmitted codewords. According to the memoryless property of the channel, it satisfies that

$$p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_{R_n}). \tag{3.56}$$

We define the function $f_{R_n}(\boldsymbol{x}_{R_n})$ as

$$f_{R_n}(\boldsymbol{x}_{R_n}) = p(y_n|\boldsymbol{x}_{R_n}). \tag{3.57}$$

Then we can describe the ML decoding problem as

$$\begin{aligned} \hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x} \in \bar{\mathcal{C}}} p(\boldsymbol{y}|\boldsymbol{x}) && (3.58) \\ &= \arg \max_{\boldsymbol{x} \in \{0,1\}^{UN}} \prod_{n=1}^{N} f_{R_n}(\boldsymbol{x}_{R_n}) \prod_{u=1}^{U} \prod_{m=1}^{M_u} f_{I_{u,m}}(\boldsymbol{x}_{I_{u,m}}). && (3.59) \end{aligned}$$

where

$$f_{I_{u,m}}(\boldsymbol{x}_{I_{u,m}}) = \begin{cases} 1 & \text{if } \sum_{n \in \mathcal{N}_u(m)} x_{u,n} = 0 \bmod 2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.60}$$

and $\mathcal{N}_u(m) = \left\{n : H^u_{m,n} \neq 0\right\}$, $\boldsymbol{x}_{I_{u,m}} = (x_{u,n})_{n \in \mathcal{N}_u(m)}$.

3.5.2 Derivation of the Linear Programming based Decoder

According to the derivation in the previous section, we should solve the maximization problem of the function g', which is defined as

$$g'(\boldsymbol{x}, \boldsymbol{w}) = \prod_{n=1}^{N} f_{R_n}(\boldsymbol{w}_n) \prod_{u=1}^{U} \prod_{m=1}^{M_u} f_{I_{u,m}}(\boldsymbol{x}_{I_{u,m}}) \prod_{n=1}^{N} f_{I'_n}(\boldsymbol{w}_n, \boldsymbol{x}_{R_n}), \tag{3.61}$$

where $\boldsymbol{w}_n$ are variables which takes those values in $\mathcal{W}_n = \{0,1\}^U$ and the functions $f_{I'_n}$ are defined as

$$f_{I'_n}(\boldsymbol{w}_n, \boldsymbol{x}_{R_n}) = \begin{cases} 1 & \text{if } \boldsymbol{w}_n = \boldsymbol{x}_{R_n} \\ 0 & \text{otherwise} \end{cases} \qquad n = 1, 2, \cdots, N. \tag{3.62}$$

We define the variables to obtain the LP. Let $\boldsymbol{\tau}_{u,n} = (\tau_{u,n:\alpha})_{\alpha\in\{0,1\}}$ and $\boldsymbol{\tau}_n = (\tau_{n:\boldsymbol{\beta}})_{\boldsymbol{\beta}\in\{0,1\}^U}$ and $\boldsymbol{\tau} = \left((\boldsymbol{\tau}_{u,n})_{u=1,2,\cdots,U,\ n=1,2,\cdots,N}, (\boldsymbol{\tau}_n)_{n=1,2,\cdots,N}\right)$. The length of $\boldsymbol{\tau}$ is $d = 2UN + 2^U N$. Then we obtain the following LP problem:

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} \sum_{\boldsymbol{\beta}\in\{0,1\}^U} \lambda_{n:\boldsymbol{\beta}} \tau_{n:\boldsymbol{\beta}} \\ \text{subject to} & \boldsymbol{\tau} \in \mathcal{Q} \end{array} \tag{3.63}$$

where $\lambda_{n:\boldsymbol{\beta}}$ and $\mathcal{Q}$ are defined as

$$\lambda_{n:\boldsymbol{\beta}} = -\ln f_{R_n}(\boldsymbol{\beta}), \tag{3.64}$$

$$\mathcal{Q} = \text{conv}(\mathcal{P}), \tag{3.65}$$

$$\mathcal{P} = \Big\{\boldsymbol{\tau} \in \{0,1\}^d : \text{Proj}_{u,m}(\boldsymbol{\tau}) \in \mathcal{P}_{I_{u,m}}, u = 1,2,\cdots,U,\ m = 1,2,\cdots,M_u, \\ \text{Proj}_n(\boldsymbol{\tau}) \in \mathcal{P}_{I'_n}, n = 1,2,\cdots,N\Big\} \tag{3.66}$$

$$\mathcal{P}_{I_{u,m}} = \Big\{(\phi_n(x_n))_{n\in\mathcal{N}_u(m)} : f_{I_{u,m}}(\boldsymbol{x}_{I_{u,m}}) = 1\Big\} \tag{3.67}$$

$$\mathcal{P}_{I'_n} = \Big\{\Big(\phi_n(\boldsymbol{w}_n), (\phi_n(x_n))_{n=1,2,\cdots,N}\Big) : f_{I'_n}(\boldsymbol{w}_n, \boldsymbol{x}_{R_n}) = 1\Big\}, \tag{3.68}$$

where $\text{Proj}_{u,m}(\boldsymbol{\tau}) = (\boldsymbol{\tau}_n)_{n\in\mathcal{N}_u(m)}$ and $\text{Proj}_n(\boldsymbol{\tau}) = \Big(\boldsymbol{\tau}_n, (\boldsymbol{\tau}_{u,n})_{u=1,2,\cdots,U}\Big)$.

The relaxed problem is described as

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} \sum_{\boldsymbol{\beta}\in\{0,1\}^U} \lambda_{n:\boldsymbol{\beta}} \tau_{n:\boldsymbol{\beta}} \\ \text{subject to} & \boldsymbol{\tau} \in \tilde{\mathcal{Q}} \end{array} \tag{3.69}$$

where $\tilde{\mathcal{Q}}$ is

$$\tilde{\mathcal{Q}} = \Big\{\boldsymbol{\tau} \in \{0,1\}^d : \text{Proj}_{u,m}(\boldsymbol{\tau}) \in \text{conv}(\mathcal{P}_{I_{u,m}}),\ u = 1,2,\cdots,U,\ m = 1,2,\cdots,M_u \\ \text{Proj}_n(\boldsymbol{\tau}) \in \text{conv}(\mathcal{P}_{I'_n}), n = 1,2,\cdots,N\Big\} \tag{3.70}$$

According to the results for single-user channels, conv $\left(\mathcal{P}_{I_{u,m}}\right)$ is described as

$$\sum_{n\in S} \tau_{u,n:1} - \sum_{n\in\mathcal{N}_u(m)\setminus S} \tau_{u,n:1} \le |S| - 1, \quad \forall S \subseteq \mathcal{N}_u(m) \text{ s.t. } |S| \text{ is odd} \tag{3.71}$$

$$0 \le \tau_{u,n:1} \le 1, \quad \forall n \in \mathcal{N}_u(m) \tag{3.72}$$

The polytope conv $\left(\mathcal{P}_{I'_n}\right)$ is described as

$$\begin{cases} \tau_{u,n:1} = \sum_{\boldsymbol{\beta}:\beta_u=1} \tau_{n:\boldsymbol{\beta}} \\ \tau_{u,n:0} = \sum_{\boldsymbol{\beta}:\beta_u=0} \tau_{n:\boldsymbol{\beta}} \end{cases} \quad u = 1,2,\cdots,U \tag{3.73}$$

$$0 \le \tau_{n:\boldsymbol{\beta}} \le 1, \quad \forall \beta \in \{0,1\}^U \tag{3.74}$$

$$\sum_{\boldsymbol{\beta}\in\{0,1\}^U} \tau_{n:\boldsymbol{\beta}} = 1 \tag{3.75}$$

We can eliminate the variables $\tau_{u,n:0}$ by substituting the constraints $\tau_{u,n:0} = 1 - \tau_{u,n:1}$. Summarizing the results, the relaxed LP problem for the ML decoding over multiple-access channels is described as

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} \sum_{\boldsymbol{\beta} \in \{0,1\}^U} \lambda_{n:\boldsymbol{\beta}} \tau_{n:\boldsymbol{\beta}} \\ \text{subject to} & \tau_{u,n:1} = \sum_{\boldsymbol{\beta}:\beta_u=1} \tau_{n:\boldsymbol{\beta}},\ \forall u = 1,2,\cdots,U,\ n = 1,2,\cdots,N \\ & 0 \le \tau_{n:\boldsymbol{\beta}} \le 1,\ \forall n = 1,2,\cdots,N,\ \boldsymbol{\beta} \in \{0,1\}^U \\ & \sum_{\boldsymbol{\beta}} \tau_{n:\boldsymbol{\beta}} = 1,\ \forall n = 1,2,\cdots,N, \\ & \forall u = 1,2,\cdots,U,\ m = 1,2,\cdots,M_u,\ S \subseteq \mathcal{N}_u(m) \text{ s.t. } |S| \text{ is odd} \\ & \sum_{n \in S} \tau_{u,n:1} - \sum_{n \in \mathcal{N}_u(m) \setminus S} \tau_{u,n:1} \le |S| - 1 \end{array} \tag{3.76}$$

In this formulation, the number of variables is $O(2^U N)$ and the number of constraints is $O(2^U N + \sum_u M_u 2^{d_{max}^{(u)}})$ where $d_{max}^{(u)}$ is the maximum row Hamming weight of parity check matrix $H^{(u)}$. We can solve the problem by LP solver such as the simplex method or interior-point method. The decoding algorithm works as follows. The decoder solves the LP problem (3.76). If $\tau_{u,n:1} \in \{0,1\}$ for all u and n, output $\tau_{u,n:1}$ as the ML estimates of $x_{u,n}$. If there are any non-integer elements in $\boldsymbol{\tau}$, the decoder declare a decoding failure.

3.6 Concluding Remarks

In this chapter, we construct the linear programming (LP) -based inference algorithms for general factor graphs. The LP-based inference algorithm for the factor graph which does not include the multi-degree non-indicator function nodes, which are the nodes correspond to the non-indicator function that connected to more than one variable node, had been proposed in the past research. However, general probabilistic inference problem possibly has any multi-degree non-indicator functions. We extended that we can apply the LP based inference algorithm for such problems. The proposed algorithm has the ML certificate property as in the case for the past research.

As an application, we proposed the LP based decoding algorithm for binary linear codes over memoryless multiple-access channels. We deal with the problem again in Chapter 4. We will show that we can possibly reduce the computational complexity of the inference algorithm for certain class of multiple-access channels. It would be interesting to improve the inference algorithm by using the combinatorial optimization algorithms. We will deal with such algorithms in Chapter 5.

Chapter 4

Transformation of the Factor Graph and its Application

4.1 Introduction

In this chapter, we show that we can reduce the computational complexity of the inference algorithms by changing the factorization structure of the posterior distribution. In general, the computational complexity of the inference algorithms based on the Sum-Product (SP) and the Linear Programming (LP) are exponential order in the degree of the factor node, where degree is a number of nodes which are connected to the factor node. We investigate the factor graph structure for the decoding problems for the Direct Sequence Code Division Multiple Access Channel (DS-CDMA) and the Multiple Access Channel (MAC) with binary linear codes. This enables us to implement the SP algorithm for the decoding problem for the DS-CDMA, it was believed that it is impossible to apply the SP algorithm to the problem. We also show that the outputs of the LP for the two different factorizations for the MAC with binary linear codes are equivalent.

4.2 Application to DS-CDMA Channel Model

In a direct sequence code division multiple access (DS-CDMA) wireless communication environment, multiuser detection is an important demodulation technique.

The Maximum Posterior Marginal (MPM) detector is optimum with respect to the bit error rate [11], but the computational complexity of the detector increases exponentially with the number of users. In this section, we illustrates the application of the SP algorithm for approximate MPM detector.

Since the detection problem can be regarded as a probabilistic inference problem, it is natural to attempt to apply the SP algorithm to the problem. Previously, the idea to apply the SP algorithm to the multiuser detection problem was already suggested in [3]. But it has been reported that hence the computational complexity of the resulting algorithm increases expo-

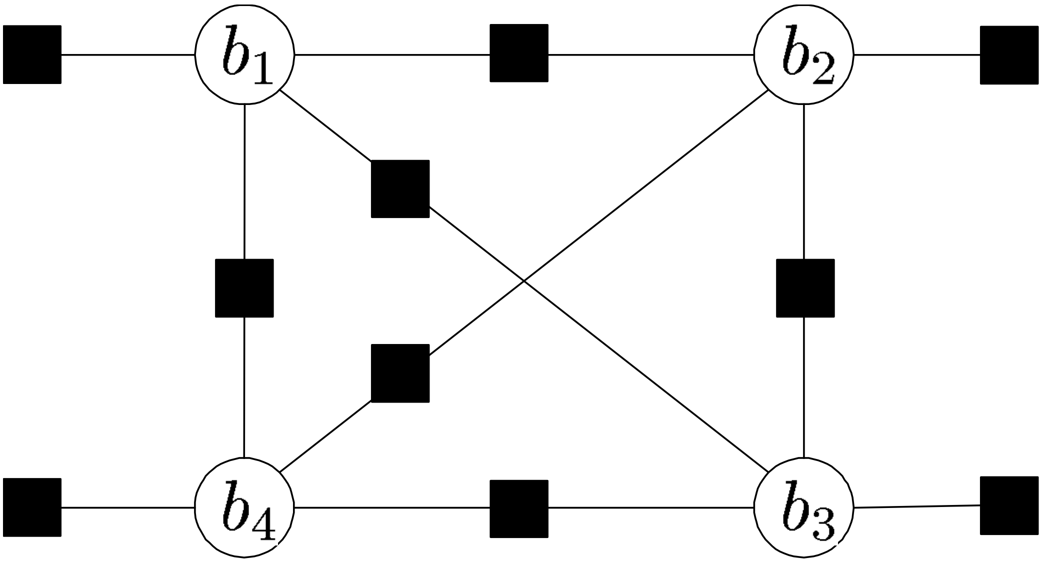

Figure 4.1: An example of the factor graph for the detection problem factorizing with outputs of matched filter and cross correlations ($U = 4$).

nentially with the number of users, it is impossible to apply the SP algorithm to the detection problem. This is also described in chapter 2 of the thesis. Thus main interest in [3] is the parallel interference canceller (PIC) rather than the SP itself. That paper indicates that PIC is derived as an approximation of SP.

In this chapter, we suggest that the SP algorithm can be applied to the detection problem by converting the factor graph structure. The form of the proposed factor graph is depend on the signature sequences and in some cases the graph has only few cycles. In such cases, the proposed algorithm is expected to perform a attractive behavior.

4.2.1 Sum-Product Algorithm based Detector

We suggest that we can change the structure of the factor graph using the outputs of matched filter and cross correlation functions, and as a result, SP algorithm can be applied.

In general, there are various ways to factorize the function and the structure of the factor graph is not unique. The posterior function (2.44) can be factorize into the following form

$$p(\boldsymbol{b}|\boldsymbol{r}) = \frac{1}{Z}\prod_{u=1}^{U} \exp\left(\frac{b_u h_u}{\sigma^2}\right) \prod_{u<u'} \exp\left(-\frac{b_u b_{u'} W_{uu'}}{\sigma^2}\right), \tag{4.1}$$

where h_u and $W_{uu'}$ are defined in (2.61). According to above factorization, factor graph is expressed as Fig. 4.1.

In the proposed factor graph, factor node corresponds to a function $f_u(b_u) = \exp(b_u h_u/\sigma^2)$ is only connected to variable node b_u for each u, and factor node corresponds to a function $f_{uu'}(b_u, b_{u'}) = \exp(-b_u b_{u'} W_{uu'}/\sigma^2)$ is connected between two different variable nodes b_u and $b_{u'}$.

The proposed factor graph seemingly has many cycles, however, the number of cycles will descend for some cases. If $W_{uu'} = 0$, then the value of the function $f_{uu'}$ does not depend on its argument and takes value 1. Consequently we can eliminate the corresponding factor node.

Applying SP algorithm for the proposed factor graph, the algorithm is described as follows. Messages from factor node $f_{uu'}$ to variable nodes $b_u, b_{u'}$ are updated as (for all $u' > u$)

$$m^{(t)}_{uu'\to u}(b_u) = \sum_{b_{u'}} f_{uu'}(b_u, b_{u'}) \cdot m^{(t-1)}_{u'\to uu'}(b_{u'}), \tag{4.2}$$

$$m^{(t)}_{uu'\to u'}(b_{u'}) = \sum_{b_u} f_{uu'}(b_u, b_{u'}) \cdot m^{(t-1)}_{u\to uu'}(b_u). \tag{4.3}$$

On the other hand, messages from variable node $b_u, b_{u'}$ to factor node $f_{uu'}$ are updated as (for all $u' > u$)

$$m^{(t-1)}_{u\to uu'}(b_u) = f_u(b_u) \prod_{v=1}^{u-1} m^{(t-1)}_{vu\to u} \prod_{v=u+1}^{U} m^{(t-1)}_{uv'\to u'}, \tag{4.4}$$

$$m^{(t-1)}_{u'\to uu'}(b_{u'}) = f_{u'}(b_{u'}) \prod_{v=1}^{u'-1} m^{(t-1)}_{vu'\to u'} \prod_{v=u'+1}^{U} m^{(t-1)}_{uv\to u}. \tag{4.5}$$

At last, beliefs are computed as

$$q^{(t)}_u(b_u) = \alpha_u f_u(b_u) \prod_{u'<u} m^{(t)}_{u'u\to u}(b_u) \prod_{u'>u} m^{(t)}_{uu'\to u}(b_u). \tag{4.6}$$

Since computational complexity of the above algorithm is $O(U^3)$, the algorithm is practical.

4.2.2 Simulations

Simulations Conditions

We compare MPM detector, PIC and SP detector with the bit error rate. Monte Carlo simulations of 10 million runs are used to approximate the detectors' bit error rate. Three kind of the signature sequences are used.

In the first example, we simulated a 7-user system with

$$\boldsymbol{W} = \frac{1}{8} \begin{pmatrix} 8 & 4 & 0 & 0 & -2 & 0 & 0 \\ 4 & 8 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 8 & 4 & -2 & 4 & 0 \\ 0 & 0 & 4 & 8 & -2 & 0 & -4 \\ -2 & -2 & -2 & -2 & 8 & 2 & 2 \\ 0 & 0 & 4 & 0 & 2 & 8 & 0 \\ 0 & 0 & 0 & -4 & 2 & 0 & 8 \end{pmatrix}, \tag{4.7}$$

where $\boldsymbol{W}$ is the sample cross correlation matrix $\{W_{uu'}\}$. In this example, the length of the signature sequences equals 8. There are many 0 elements in the matrix, the number of cycles in the proposed factor graph is relatively small.

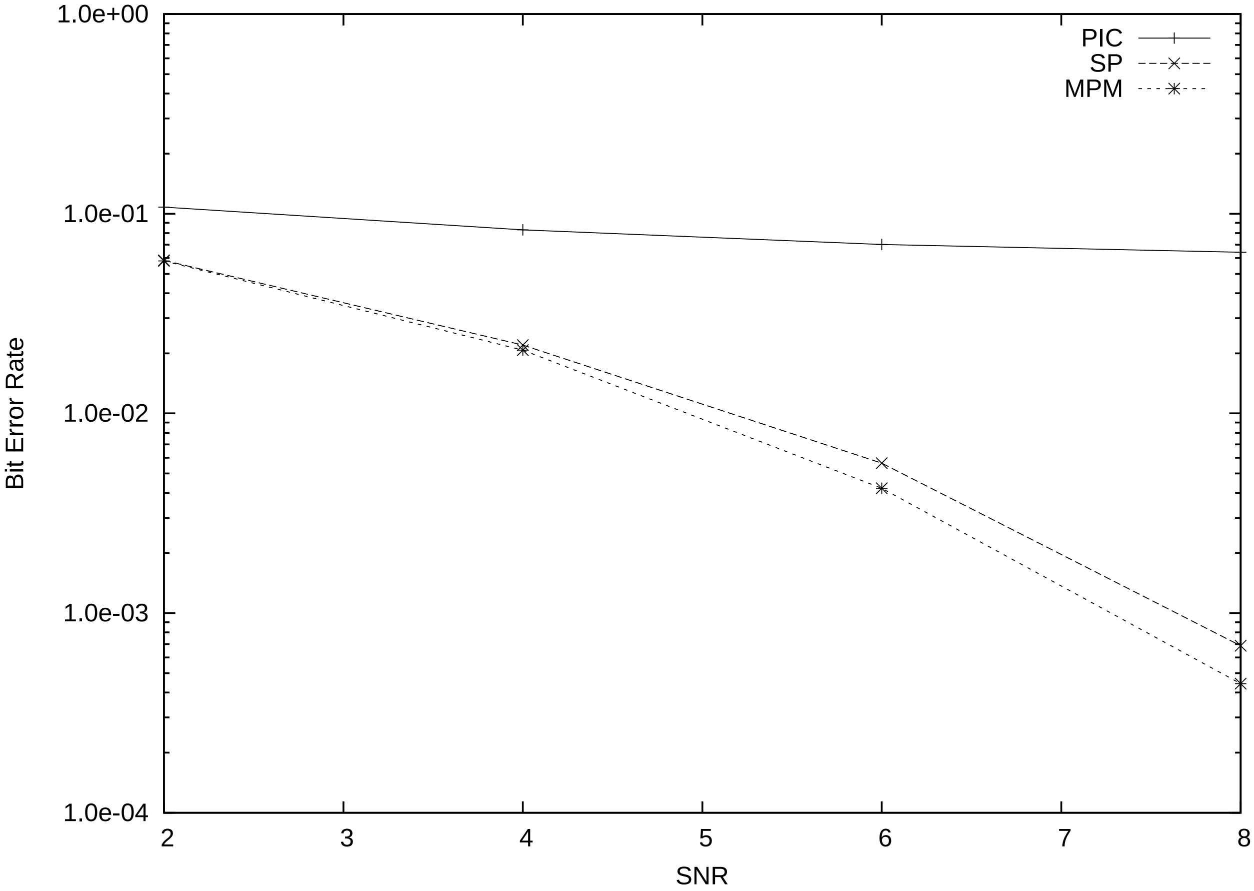

Figure 4.2: The Detector Performance for the 7 user System with the cross correlation matrix is given in (4.7).

In the second example, we simulated a 7-user system with

$$\boldsymbol{W} = \frac{1}{8}\begin{pmatrix} 8 & 0 & -4 & -2 & -2 & 0 & -2 \\ 0 & 8 & 0 & -2 & 2 & 0 & -2 \\ -4 & 0 & 8 & 2 & -2 & 4 & 6 \\ -2 & -2 & 2 & 8 & 0 & 2 & 4 \\ -2 & 2 & -2 & 0 & 8 & 2 & -4 \\ 0 & 0 & 4 & 2 & 2 & 8 & 2 \\ -2 & -2 & 6 & 4 & -4 & 2 & 8 \end{pmatrix}, \tag{4.8}$$

In this example, the length of the signature sequences equals 8.

In the third example, we simulated a 7-user system with Shift-M sequences which length equals 7. The sequences has the property that $W_{uu'} = -1/N, \quad \forall u \neq u'$ [12].

Results and Discussions

Figures 4.2, 4.3 and 4.4 shows the bit error rates of the detectors versus the signal-to-noise ratio (SNR).

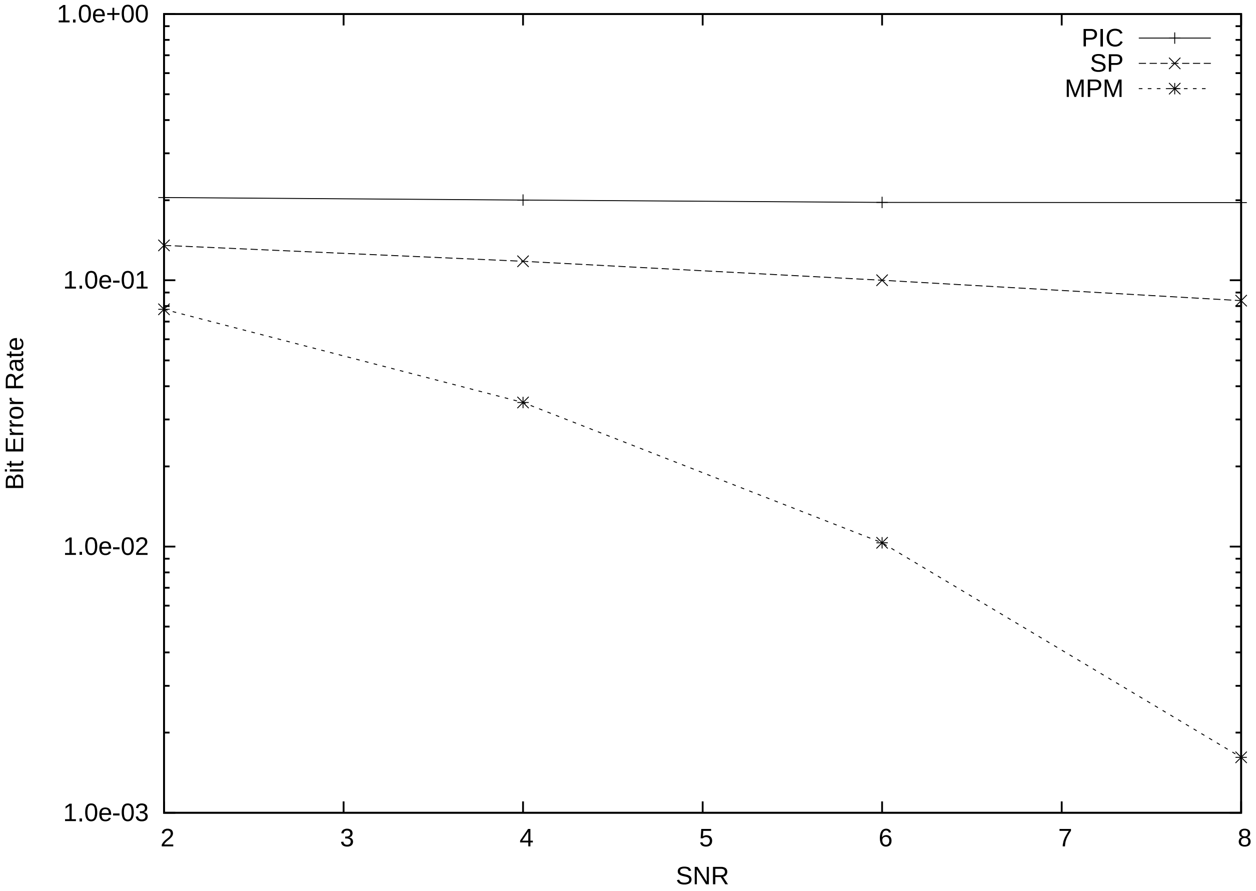

Figure 4.3: The Detector Performance for the 7 user System with the cross correlation matrix is given in (4.8).

In the first example, performance of the SP detector is near to that of the optimum MPM detector. From the result, we can say that not only for the decoding problem of the LDPC codes, the SP algorithm also performs a desirable behavior for the multiuser detection problem when the factor graph has only few cycles.

In the all results, the SP detector's performance is improved compared to the PIC. We can give two possibility for the result. One is that the PIC uses the hard bits instead of the soft bits. In generally, using hard bits instead of the soft bits causes performance degradation.

Second possibility is the form of the factor graph. According to the research about decoding problem for LDPC codes [2], the performance of the SP algorithm for the graph which has many cycles may not be good, especially when there are cycles of the length 4. If we regard the PIC as an approximation to the SP, the number of cycles of the proposed factor graph is less than that of the PIC. Moreover, there are no cycles of the length 4 in the factor graph for the SP detector while there are many in that of the PIC.

Although the proposed factor graph has many cycles in the third example, we can see that the performance of the SP detector is near to the performance of the MPM detector. On the other hand, the performance of the SP detector is bad when the factor graph has many cycles

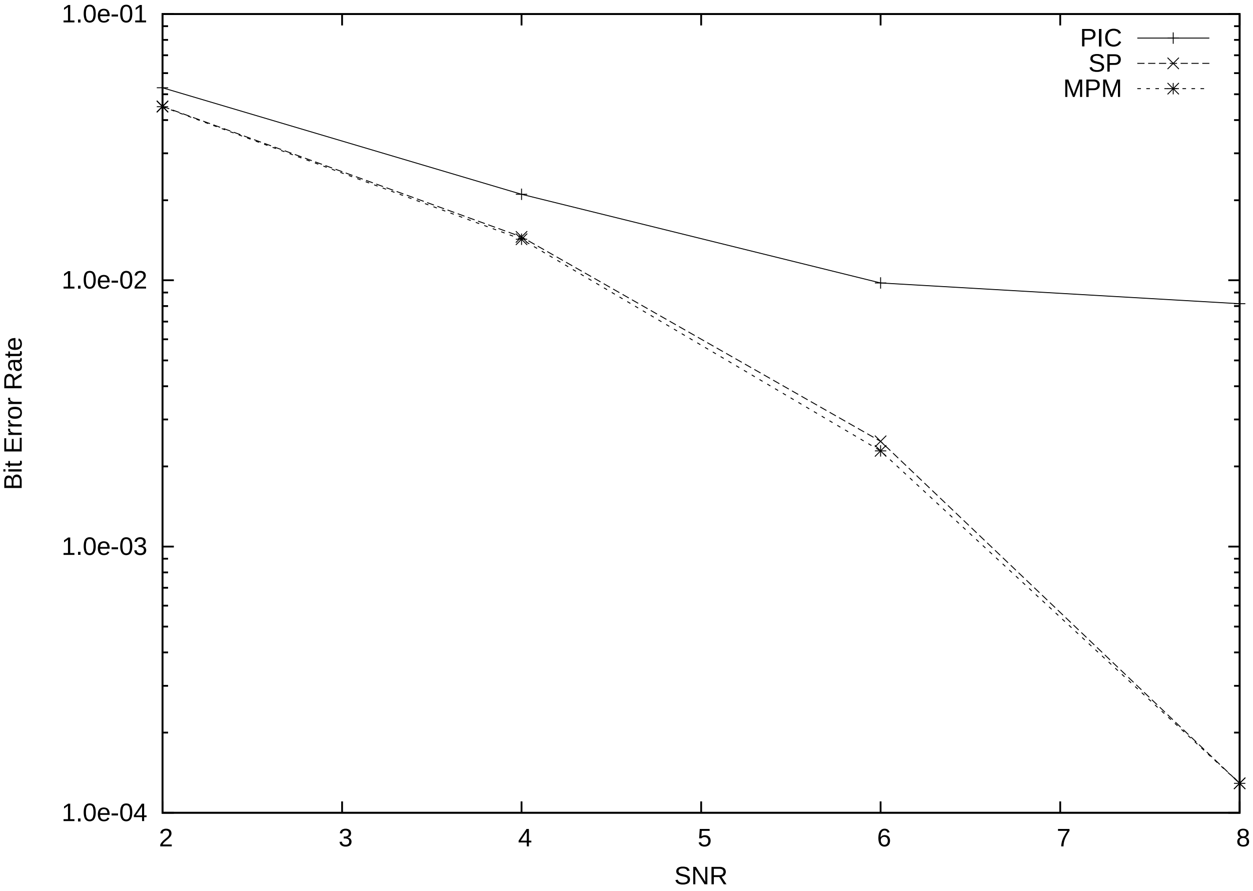

Figure 4.4: The Detector Performance for the 7 user System using the Shift M Sequences.

and the absolute values of the cross correlations are large like the second example. This shows that even when the factor graph has many cycles, if the absolute value of the cross correlations are small, the SP detector would give a good performance.

4.3 Application to Gaussian Multiple-Access Channel Model

Gaussian MAC model is a particular class of the MAC model which is described in section 3.5. According to the formulation 3.5, the computational complexity of the inference based on the LP is $O(2^U)$. We show that we can reduce the computational complexity to $O(U^2)$ by changing the factorization structure of the posterior distribution.

4.3.1 Gaussian Multiple-Access Channel Model

Gaussian MAC is modeled as [1]

$$\boldsymbol{y} = \sum_{u=1}^{U} a_u \tilde{\boldsymbol{x}}_u + \boldsymbol{\epsilon}, \tag{4.9}$$

where a_u is the amplitude of the user u and $\boldsymbol{\epsilon}$ is a Gaussian random variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 I)$. We used a notation $\tilde{\boldsymbol{x}}$ to represent the bipolar version of a binary vector $\boldsymbol{x}$ and it is given by $\tilde{\boldsymbol{x}} = \boldsymbol{1} - 2\boldsymbol{x}$. The likelihood function $p(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_U)$ is given by[2]

$$p(\boldsymbol{y}|\boldsymbol{x}_{R_n}) = \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_{R_n}), \tag{4.10}$$

$$p(y_n|\boldsymbol{x}_{R_n}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{\left(y_n - \sum_{u=1}^{U} a_u \tilde{x}_{u,n}\right)^2}{2\sigma^2} \right\}. \tag{4.11}$$

4.3.2 Factor Graph for Gaussian Multiple-Access Channel

We assume that the prior distribution $p(\boldsymbol{x})$ is uniform. Then the ML decoding problem is described as

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{C}} \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_{R_n}), \tag{4.12}$$

$$= \arg \max_{\boldsymbol{x}\in\{0,1\}^{UN}} \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_{R_n}) \prod_{u=1}^{U} \prod_{m=1}^{M_u} f_{I_{u,m}}(\boldsymbol{x}_{u,m}). \tag{4.13}$$

Then the factor graph corresponds to the above formulation, the degree of the function node corresponds to the function $p(y_n|\boldsymbol{x}_{R_n})$ is U, therefore the computational complexity of the LP based inference algorithm for the above problem is $O(2^U)$.

However, we can reduce the computational complexity of the problem by changing the factorization. The function $p(y_n|\boldsymbol{x}_{R_n})$ is expanded as

$$p(y_n|\boldsymbol{x}_{R_n}) = \frac{1}{Z} \exp \frac{\left(y_n - \sum_{u=1}^{U} a_u \tilde{x}_{u,n}\right)^2}{\sigma^2} \tag{4.14}$$

$$= \frac{1}{Z} \prod_{u=1}^{U} \exp\left(-\frac{y_n a_u \tilde{x}_{u,n}}{\sigma^2}\right) \prod_{u'>u} \exp\left(\frac{a_u a_{u'} \tilde{x}_{u,n} \tilde{x}_{u',n}}{\sigma^2}\right) \tag{4.15}$$

We define the function $f_{R_{u,n}}$ and $f_{R_{(u,u'),n}}$ as

$$f_{R_{u,n}}(x_{u,n}) = \exp\left(-\frac{y_n a_u \tilde{x}_{u,n}}{\sigma^2}\right), \quad u = 1, 2, \cdots, U, \ n = 1, 2, \cdots, N \tag{4.16}$$

$$f_{R_{(u,u'),n}}(x_{u,n}, x_{u'n}) = \exp\left(\frac{a_u a_{u'} \tilde{x}_{u,n} \tilde{x}_{u',n}}{\sigma^2}\right), \quad u' > u, \ n = 1, 2, \cdots, N. \tag{4.17}$$

[1] We use the same notation as in section 3.5

[2] Remember that $\boldsymbol{x}_{R_n} = (x_{1,n}, x_{2,n}, \cdots x_{U,n})$.

Then the ML decoding problem for Gaussian multiple-access channels is described as,

$$\hat{\boldsymbol{x}} = \arg \max_{\boldsymbol{x} \in \{0,1\}^{UN}} \prod_{n=1}^{N} \left(\prod_{u=1}^{U} f_{R_{u,n}}(x_{u,n}) \prod_{u'>u} f_{R_{(u,u'),n}}(x_{u,n}, x_{u',n}) \right) \prod_{u=1}^{U} \prod_{m=1}^{M_u} f_{I_{u,m}}(\boldsymbol{x}_{u,m}). \tag{4.18}$$

The degree of the function node $f_{R_{u,n}}$ is 1 for all u and n, and the degree of the function node $f_{R_{(u,u'),n}}$ is 2. Given that the number of the function $f_{R_{u,n}}$ is UN and of the function $f_{R_{(u,u'),n}}$ is $(N(U^2-U)/2)$, the computational complexity of the LP based decoding algorithm is reduced to $O(U^2)$.

4.3.3 Linear Programming Decoding for Gaussian Multiple-Access Channel

Here, we derive the relaxed LP formulation for the Gaussian multiple-access channels. The LP formulation for Gaussian multiple-access channels is described as,

$$\begin{array}{ll}
\text{minimize} & \sum_{n=1}^{N} \sum_{u=1}^{U} 2a_u y_n \tau_{u,n:1} - \\
& \sum_{n=1}^{N} \sum_{u'>u} a_u a_{u'} \left\{ \left(\tau_{(u,u'),n:00} + \tau_{(u,u'),n:11} \right) - \left(\tau_{(u,u'),n:01} + \tau_{(u,u'),n:10} \right) \right\} \\
\text{subject to} & \forall (u,u') \in \mathcal{U},\ n = 1,2,\cdots,N, \\
& \tau_{u,n:1} = \tau_{(u,u'),n:10} + \tau_{(u,u'),n:11},\ \tau_{u',n:1} = \tau_{(u,u'),n:01} + \tau_{(u,u'),n:11} \\
& \sum_{\boldsymbol{\beta} \in \{0,1\}^2} \tau_{(u,u'),n:\boldsymbol{\beta}} = 1, \\
& 0 \le \tau_{(u,u'),n:\boldsymbol{\beta}} \le 1, \quad \forall \boldsymbol{\beta} \in \{0,1\}^2 \\
& u = 1,2,\cdots,U,\ m = 1,2,\cdots,M_u,\ \forall S\ s.t.\ S \subseteq \mathcal{N}_u(m) \text{ and } |S| \text{ is odd}, \\
& \sum_{n \in S} \tau_{u,n:1} - \sum_{n \in \mathcal{N}_u(m) \setminus S} \tau_{u,n:1} \le |S| - 1.
\end{array} \tag{4.19}$$

The number of variables in the problem (4.19) is $O(U^2N)$ and the number of constraints is $O(U^2N + \sum_u M_u 2^{d_{max}^{(u)}})$. Therefore the LP problem (4.19) can be solved efficiently than the LP problem (3.76).

4.3.4 Simulations

In this section, we present an example of a simulation. The simulation conditions are as follows:

- The channel model is the Gaussian multiple-access channel described in (4.9)
- $U = 2$
- The amplitudes are set to $a_1 = 1.0$, $a_2 = 1.5$
- The two user codes C_1, C_2 are the $(60,30)$ and $(100,50)$ LDPC codes that satisfy $C_1 \cap C_2 = \{\boldsymbol{0}\}$

Fig. 4.5 and 4.6 show the decoding error probabilities of the LP and SP decoders, respectively. In these simulations, we measure the Eb/N0 by the ratio of the average power $\sum_u a_u^2/U$ to the power of the noise σ^2. In general, if the factor-graph has any cycles, the SP algorithm

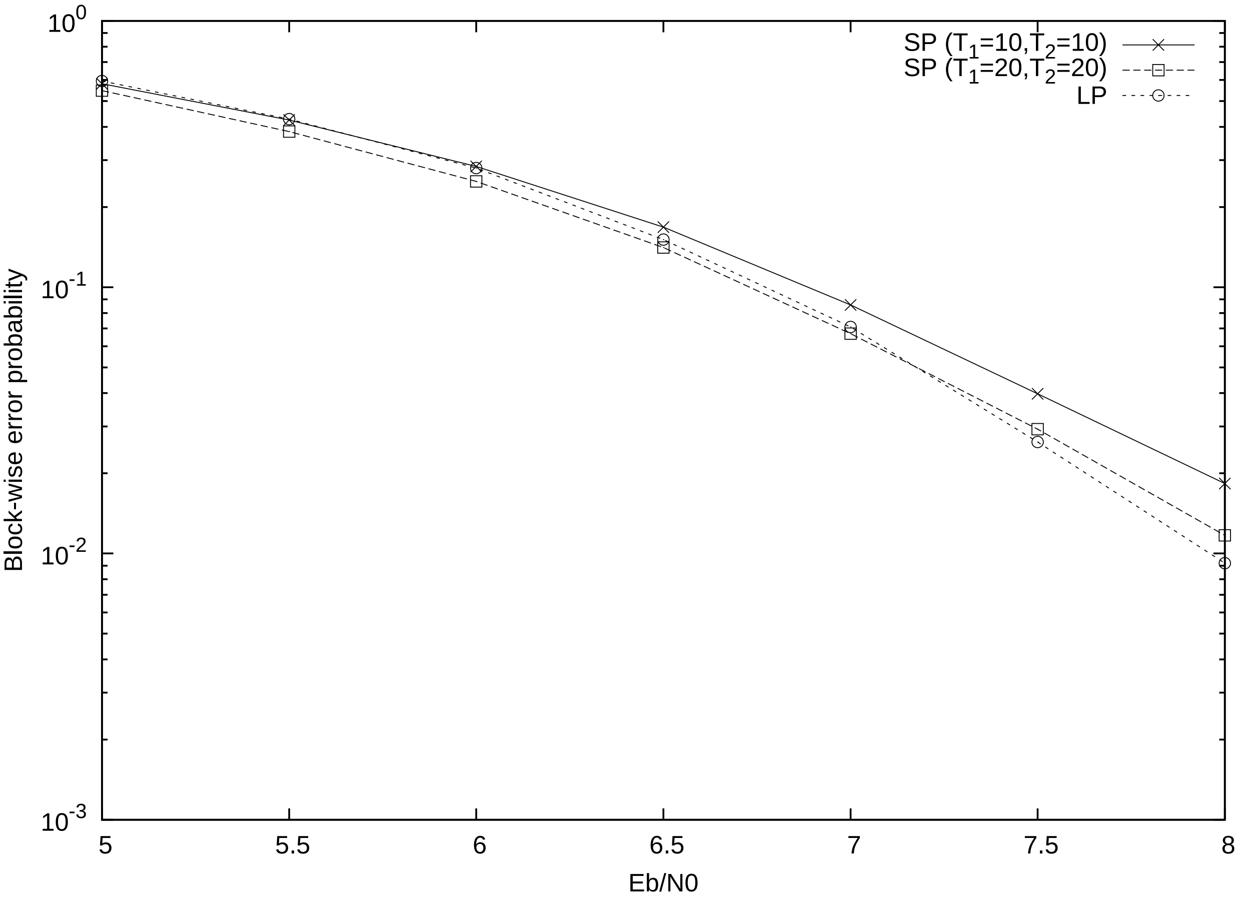

Figure 4.5: Decoding error probability of LP and SP decoders for (60, 30) LDPC codes.

performance is sensitive to the number of iterations and the order in which the computations are carried out through the nodes. This problem is known as the scheduling problem. The algorithm detail is described in —-.

The iteration number is set to $(T_1, T_2) = (10, 10)$ and $(T_1, T_2) = (20, 20)$ for the (60, 30) codes and $(T_1, T_2) = (20, 20)$ and $(T_1, T_2) = (30, 30)$ for the (100, 50) codes. Table 4.1 and 4.2 show the average decoding times of the SP and LP decoders, respectively. We see that the performance of the LP decoder is almost the same as that of the SP decoder. This result is similar to the case of single-user channels. We expect that the performance of the LP decoder will be improved by employing the cutting-plane method [13] or the mixed-integer programming method [14], as in the case for single-user channels.

4.4 Concluding Remarks

This chapter we show that we can reduce the computational complexity of the inference algorithms by changing the factorization structure of the posterior distribution. First, we investigated the factor graph structure for the decoding problem for the DS-CDMA channels. The decoding problem for the DS-CDMA channels is called multiuser detection. The idea to apply

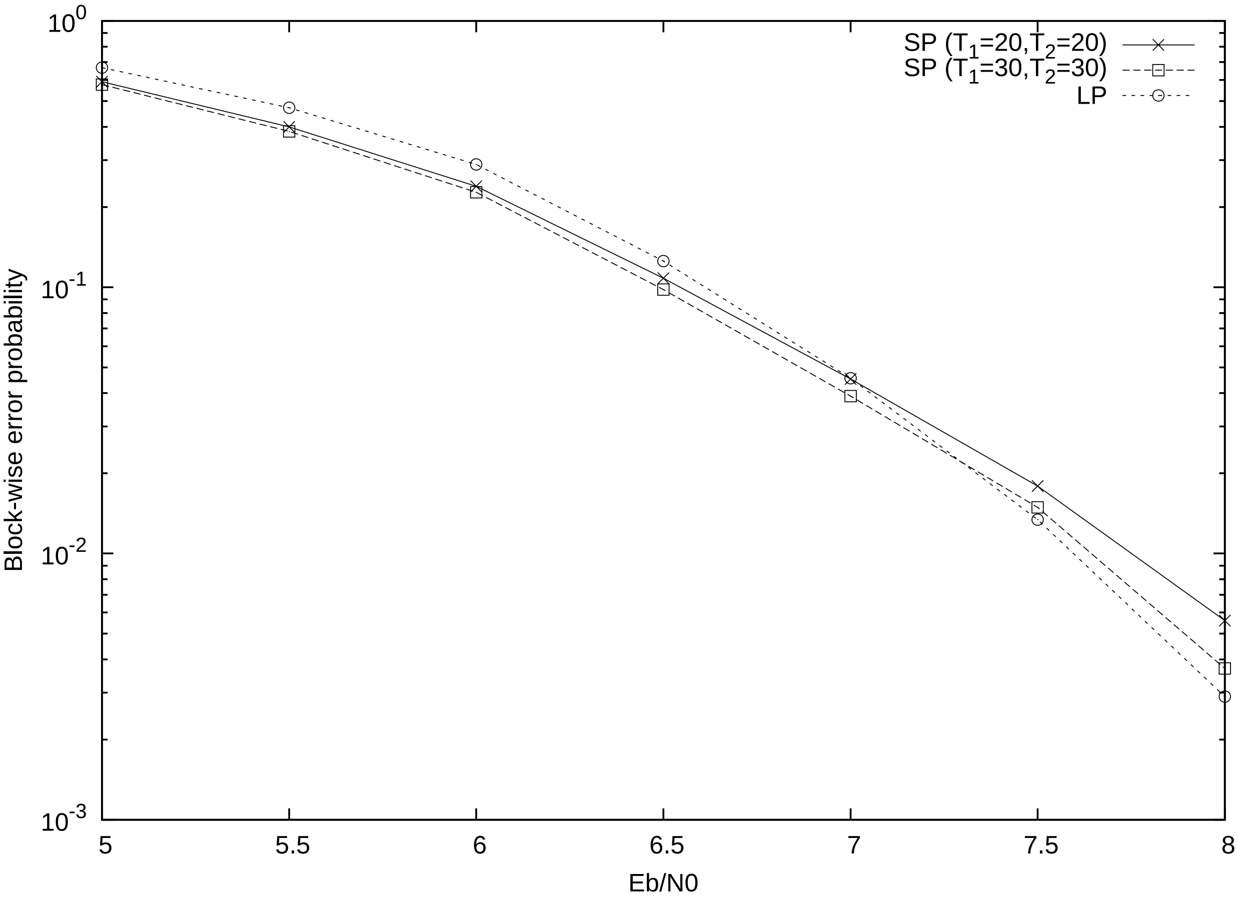

Figure 4.6: Decoding error probability of LP and SP decoders for (100, 50) LDPC codes.

BP algorithm to the multiuser detection problem was already suggested, but it had been believed that it is impossible to apply BP algorithm directly to the detection problem. We showed that we can reduce the computational complexity of BP algorithm for the detection problem by converting the graph structure and as a result, BP algorithm can be applied directly to the detection problem. Simulation results showed that BP detector provides better bit error rate than PIC, especially when the factor graph of the detection problem has no loop or has only a few cycles. There are several research problems that have not been addressed in this thesis. Firstly, we assume the synchronous CDMA model. In general, it is difficult to fully synchronize. It is desirable to extend BP algorithm to the asynchronous case. Secondly, the form of the proposed factor graph is depend on the signature sequences. It is desirable to derive a construction algorithm of the signature sequences such that corresponding factor graph has no cycles or has a few cycles.

We also investigated the factor graph structure for the decoding problem for the binary linear codes over Gaussian multiple-access channels. Based on the fact, we can reduce the computational complexity of the LP decoder for Gaussian multiple-access channels. The construction of codes that are suitable for LP decoding will be an aim of future work. It would be interesting to further reduce the time complexity and/or improve the error correcting performance of the

Table 4.1: Average decoding times (in ms) of SP and LP decoders for (60, 30) LDPC codes with Eb/N0 = 6.0 dB.

SP ($T_1 = 10, T_2 = 10$)	SP ($T_1 = 20, T_2 = 20$)	LP
0.0520	0.1762	0.1403

Table 4.2: Average decoding times (in ms) of SP and LP decoders for (100, 50) LDPC codes with Eb/N0 = 6.0 dB.

SP ($T_1 = 20, T_2 = 20$)	SP ($T_1 = 30, T_2 = 30$)	LP
0.2818	0.6259	0.4235

decoder.

Chapter 5

Application of Combinatorial Optimization Algorithm for the Inference Problem

5.1 Introduction

In this chapter, we consider to apply the combinatorial optimization algorithm for the probabilistic inference problem. We deal with the Maximum likelihood (ML) decoding problem of linear block codes over the memoryless channels. Since it is an NP-hard problem in general, there are many researches about the algorithms to approximately solve the problem. ML decoding problem can be described as an integer linear programming (ILP). LP decoder relaxes the integer constraints of the ILP and solve a linear programming over the relaxed codeword polytope.

Branch-and-bound method is one of the work to improve the performance of LP relaxation. It adaptively adds integer constraints to the original LP and solve them. Although the computational complexity of the branch-and-bound method grows exponentially with the number of enforced integer constraints, the output of the algorithm is guaranteed to be the integer. The decoding algorithms based on the branch-and-bound method are developed in [15] [14].

Another way to improve the performance of relaxed LP is to use cutting-plane methods which iteratively refine a feasible set by means of linear inequalities, termed cuts. LP decoding algorithms based on cutting-plane methods have been discussed in [16][13][17].

In this chapter, we consider the branch-and-cut based ML decoding algorithm. Branch-and-cut method is a hybrid of cutting-plane and branch-and-bound methods [18]. It starts by solving a linear programming and implements cutting-plane method to the current solution. The branch-and-cut method is widely used to solve ILP. It is important to consider how to select the technical components in the branch-and-bound method and branch-and-cut method. We show that the performance of the decoder based on the branch-and-bound method or branch-and-cut method depends on the selection of those technical components.

5.2 Cutting Plane Method and Branch-and-Bound Method

5.2.1 Cutting Plane Method

If an inequality $\boldsymbol{a}'^T\boldsymbol{x} \le b'$ is satisfied for all $\boldsymbol{x} \in \mathcal{C}$, the inequality is called "valid inequality". A valid inequality becomes a "cut" for $\hat{\boldsymbol{x}}$ if the inequality is violated at $\hat{\boldsymbol{x}}$. Cutting-plane methods iterate the process that search any cuts and add them to the problem and re-solve the updated problem. There are various way to find a cut for $\hat{\boldsymbol{x}}$. In [13, 17, 16], they proposed to construct a cut for the ML decoding problem based on the redundant parity checks, which are the sum of two original parity checks. The ALP decoding algorithm presented in [16] is an LP solver based on the cutting plane method. It starts from the simple initial problem that the optimization variables x_i are only constrained $0 \le x_i \le 1$, and iteratively adds cuts which are contained in the set of constraints. An efficient algorithm to find cuts from the set of constraints is presented in [16]. Taghavi et al. also presented an algorithm to construct cuts from redundant parity checks. The algorithm is described as follows [16]:
Algorithm:Redundant Parity Check Search Algorithm

Step.1 Having a solution $\boldsymbol{x}$, prune the Tanner graph by removing all the variable nodes with integer values.

Step.2 Starting from an arbitrary check node, randomly walk through the pruned graph until a cycle is found.

Step.3 Create an RPC by combining the rows of the parity-check matrix corresponding to the check nodes in the cycle.

Step.4 If this RPC does not introduce a cut, go to Step.2.

5.2.2 Branch-and-Bound Method

Branch-and-bound method adds integer constraints into the LP. The processes of the algorithm can be expressed in tree search. Added integer constraints are expressed as paths in the tree and different linear programming problem is allocated to each node. The top node is called the root node, which is at level 0. The original LP problem is allocated to the root node. Each node has two branches, labeled by 0 and 1, respectively, and nodes at level $n-1$ have no branches. Each branch corresponds to an integer constraint $x_n = l$ where $n \in \{1, \cdots, N\}$, $l \in \{0, 1\}$. The child problem, which is a problem allocated to a child node of a node, is obtained by adding an integer constraint corresponding to a branch from its parent node. The example is illustrated in Fig. 5.1.

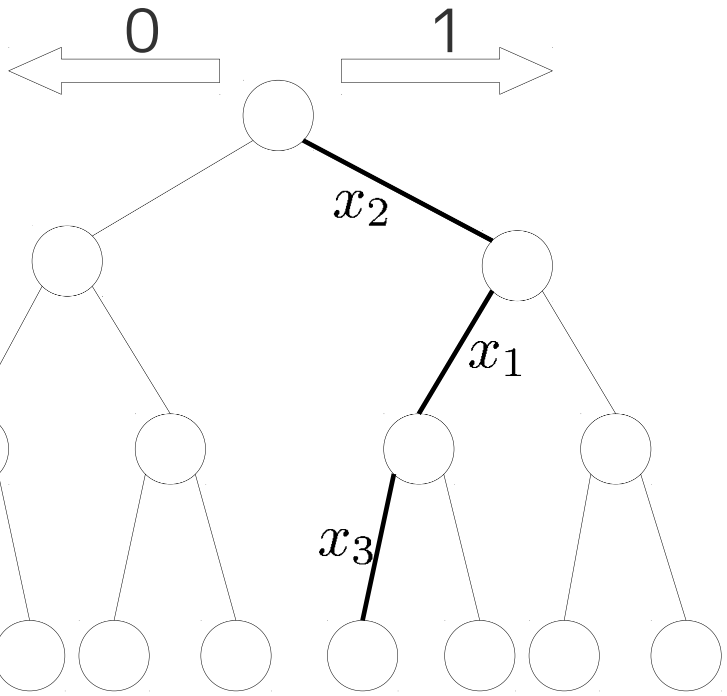

Figure 5.1: Integer constraints are expressed in a tree. One branch corresponds to one integer constraint. For example, the bold path corresponds to the constraints $x_2 = 1, x_1 = 0, x_3 = 0$

5.3 Branch-and-Cut Method

The branch-and-cut method is a hybrid of cutting-plane and branch-and-bound method [18]. The algorithm is described as follows:

Algorithm 2

Step.1 Set a root node in the node list $\mathcal{L}$ and initialize $\boldsymbol{x}^* = \mathbf{0}$, $z^* = 0$.

Step.2 Get and remove a node from the node list $\mathcal{L}$. If $\mathcal{L} = \{\phi\}$, output $\boldsymbol{x}^*$ as a optimum solution.

Step.3 Let $\mathcal{K}$ be the indices of the non-integer variables in the solution of LP correspond to the node selected in Step.2. Select a branching variable $x_n, n \in \mathcal{K}$. Create two child nodes and allocate LPs, which are obtained by adding a constraint $x_n = 0$ or $x_n = 1$, respectively.

Step.4 Solve two child problems created in Step.3. Let $\tilde{\boldsymbol{x}}$ be the solution of a problem and $\tilde{z}$ be the objective value.

- If $\tilde{\boldsymbol{x}}$ is non-integer and $\tilde{z} < z^*$. Implement the cutting-plane method and add cuts found in the cutting-plane method to the problem. Let $\boldsymbol{x}'$ and z' be the solution and objective value of the problem after the cutting plane method is implemented, respectively. If $\boldsymbol{x}'$ is non-integer and $z' < z^*$, add the node to $\mathcal{L}$. Else if $\boldsymbol{x}'$ is integer and $z' < z^*$, update $\boldsymbol{x}^* := \boldsymbol{x}'$, $z^* := z'$. Remove nodes in $\mathcal{L}$ whose objective value is greater than $\boldsymbol{z}^*$.
- If $\tilde{\boldsymbol{x}}$ is integer and $\tilde{z} < z^*$, update $\boldsymbol{x}^* := \tilde{\boldsymbol{x}}$, $z^* := \tilde{z}$. Remove nodes in $\mathcal{L}$ whose objective value is greater than $\boldsymbol{z}^*$.

The difference between the branch-and-bound method and branch-and-cut method is that the branch-and-cut method implements the cutting-plane method in Step.4. In Step.2, there are some choices to select a node from the node list. Breadth-first search, Depth-first search and Priority-first search are well-known strategy. In [14, 15], the authors proposed the branch-and-bound method based decoder and they used only the breadth-first search or the depth-first search. However, the priority-first search should be a powerful searching strategy. If we use the priority-first search strategy, the node on the top of the list, which has the smallest objective value of the corresponding LP problem, is selected in Step.2.

The number of nodes visited in the algorithm also changes according to the choice of the branching variable in the generating process of child problem in Step.3. In the algorithm proposed in [14, 15], the choice of the branching variable is determined according to the value of optimization variables. In [14], the branching variable is determined by

$$n^{'} = \arg\min_{n} |x'_n - 0.5| \tag{5.1}$$

where x'_n is nth element of $\boldsymbol{x}'$, which is the optimum solution of the problem allocated to the parent node.

5.4 Simulation Results

In this section we compare some decoding algorithms through numerical simulations. Simulation results are based on three regular LDPC codes which details are shown in Table 5.1 (denoted by "Code 1-3"). w_c and w_r denote a column weight and a row weight of a parity-check matrix, respectively. We assume the additive white Gaussian noise (AWGN) channel. The results are the average of 10000 experiments. ALP algorithm presented in [16] is used to solve LP problems. ALP algorithm solves many LPs and therefore it requires other LP solver. We used GLPK [19] as LP solver. Simulations were run on 2.4GHz Intel Core 2 duo processors.

5.4.1 Searching Strategy

First, we see the difference of the average numbers of the visited nodes which is caused by the choice of searching strategy in Step.2. We compare the three algorithms, which are based on

Table 5.1: Details of the codes

	n	Rate	(w_c, w_r)
Code 1	60	0.5	$(3, 6)$
Code 2	100	0.5	$(3, 6)$
Code 3	100	0.7	$(3, 10)$

branch-and-cut decoder and the choice of branching variable is determined according to (5.1). The cutting-plane method is implemented based on the *algorithm* 1 proposed in [16] and we fix the number of iterations in *algorithm* 1 to 10. The difference among the algorithms is only searching strategy in Step.2 and they are breadth-first search, depth-first search and priority-first search. The average number of visited nodes of the algorithms are shown in Fig.5.2-5.4. For high SNR, there isn't so much difference in the number of visited nodes among the algorithms, however, for low SNR, the algorithm employing the priority-first search is superior to the other algorithms.

Since the operations other than search are also needed for the algorithms, it is not fair to compare the complexity only in the number of visited nodes. For the proposed priority-first employing algorithm, the algorithm must search the node whose objective value is the smallest in the node list $\mathcal{L}$. The time complexity of this operation is the linear order of the number of nodes in the node list $\mathcal{L}$ and it is very small compared to the time complexity of solving the LPs. In Fig. 5.5, we provide the decoding time of the algorithms versus SNR for the code 1. We can see that the algorithm employing the priority-first search is still superior to the other algorithms.

5.4.2 Effect of Cutting-Plane Method

Here, we compare the branch-and-bound decoder and branch-and-cut decoder. Both algorithm employ the priority-first-search and the choices of branching variables are determined based on (5.1). The cutting-plane method is implemented based on the *algorithm* 1 proposed in [16] and we fix the number of iterations in *algorithm* 1 to C_{max}. We compare the decoding times for different value of C_{max}. Fig. 5.6-5.8, provides the decoding times of the algorithms versus SNR. We can see that the average decoding time of the branch-and-cut decoder is superior to branch-and-bound decoder. We can also see that cutting-plane method produce a large effect in low SNR setting.

5.4.3 Performance Comparison to LP Decoding

Here we compare the word-error rate (WER) of ALP decoding to ML decoding. The performance of ML decoding is obtained by our branch-and-cut algorithm. The estimates of the ALP WER and the ML WER is plotted in Fig.5.9. We can see that there is a large gap between the ALP

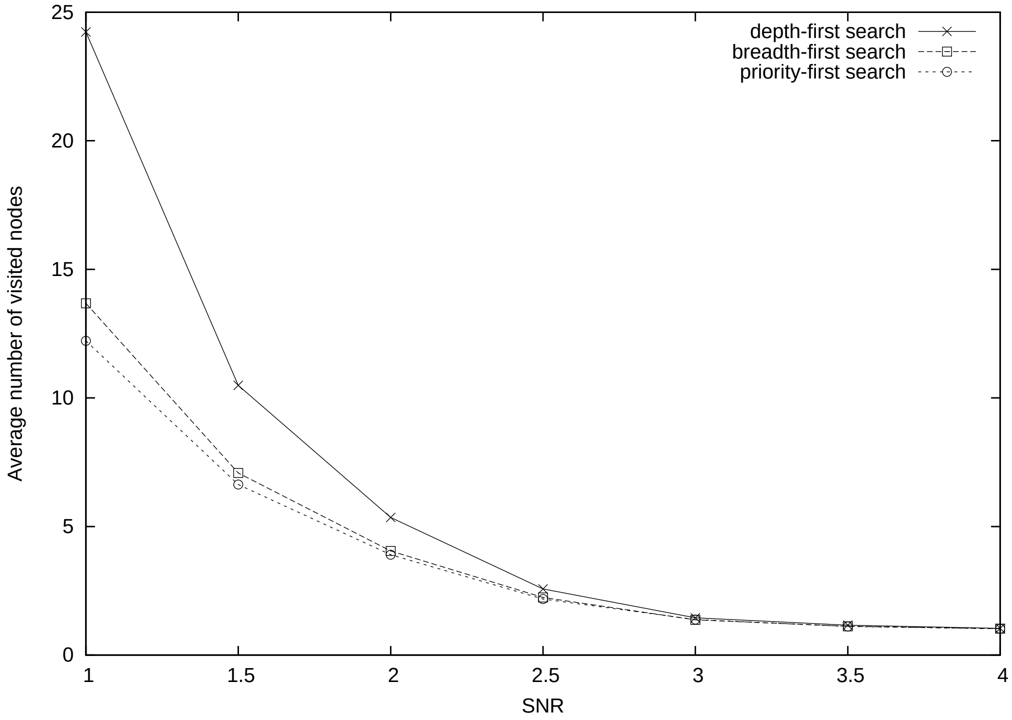

Figure 5.2: The average number of visited nodes for the code 1 as a function of SNR.

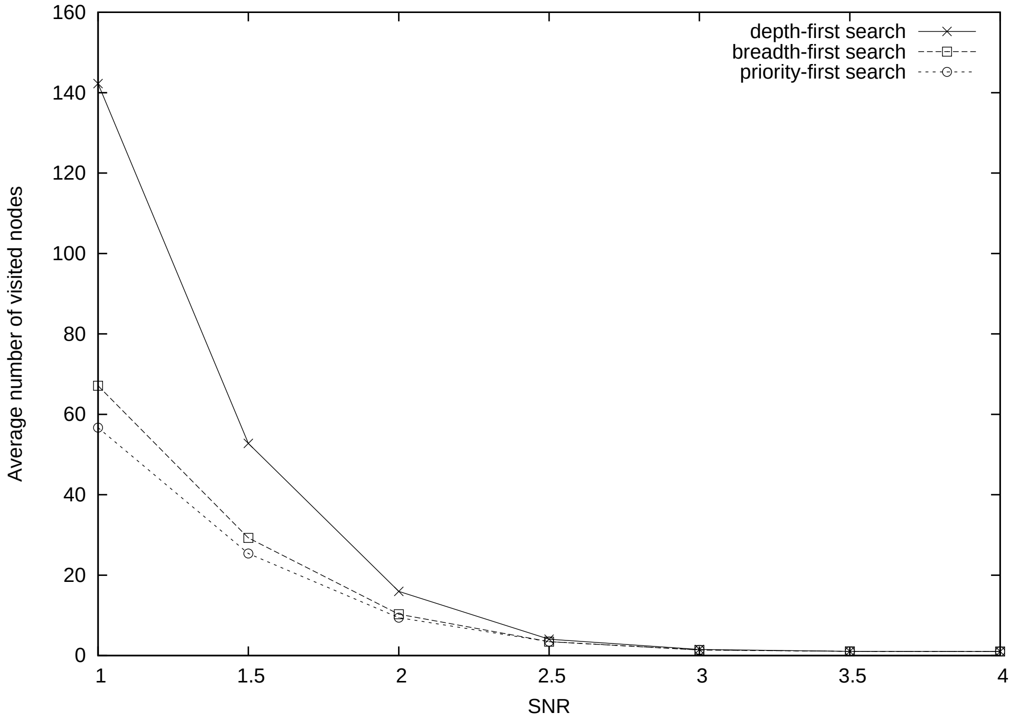

Figure 5.3: The average number of visited nodes for the code 2 as a function of SNR.

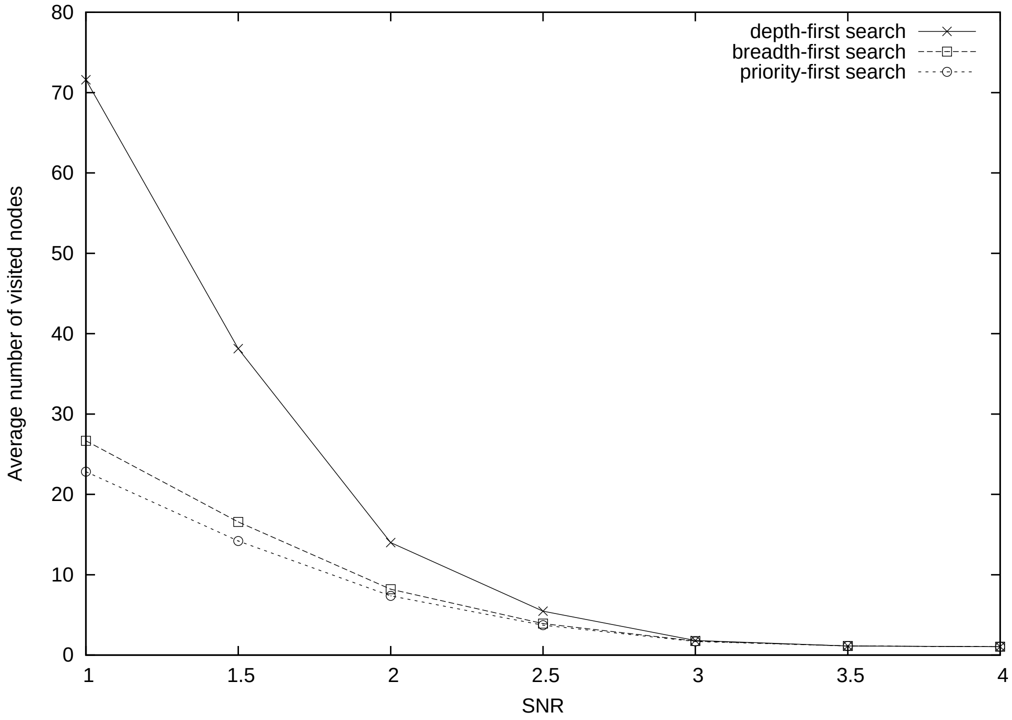

Figure 5.4: The average number of visited nodes for the code 3 as a function of SNR.

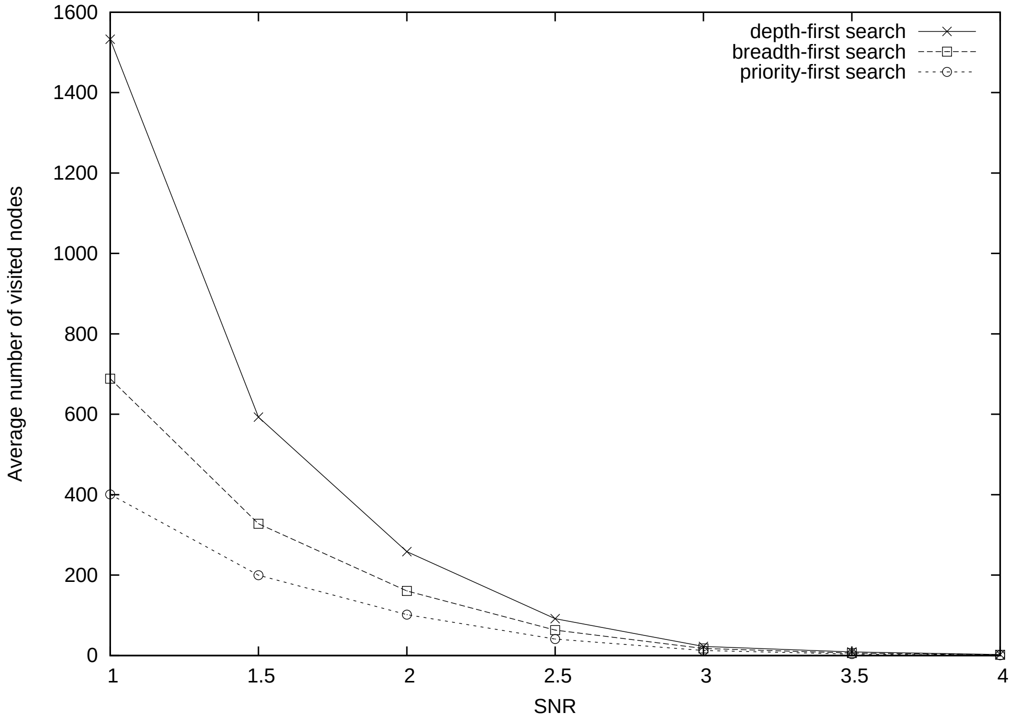

Figure 5.5: The average decoding time for the code 1 as a function of SNR.

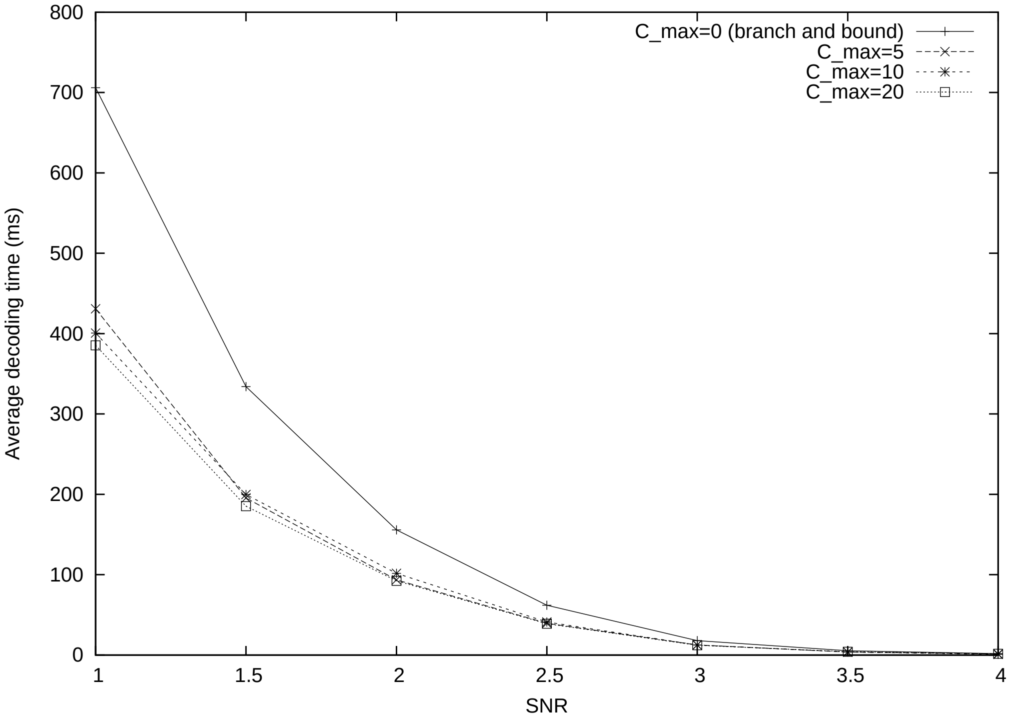

Figure 5.6: The efficiency of cutting-plane method (result for the the code 1).

SNR	1.0	1.5	2.0	2.5	3.0
average decoding time of ALP (ms)	6.3	4.6	3.5	2.4	1.7
average decoding time of branch-and-cut (ms)	385.5	185.3	92.4	39.0	12.2

Table 5.2: Average decoding time of ALP and Branch-and-cut algorithm for the code 1.

WER and the ML WER.

We are also interest in the computation requirement of branch-and-cut decoding. In Table 5.2 we provide the average decoding times of ALP and branch-and-cut decoding. In the low SNR setting, the branch-and-cut decoding requires dozen of times time.

5.5 Concluding Remakrs

In this chapter we present a branch-and-cut decoding of linear block codes. Since the branch-and-cut method is an extension of the branch-and-bound method and cutting-plane method, the proposed decoding algorithm is an extension of the decoding algorithm based on those methods. We also show that we can reduce the time complexity of the ML decoding by tune-up the algorithm. However, the algorithm is still time consuming especially for low SNR setting. There

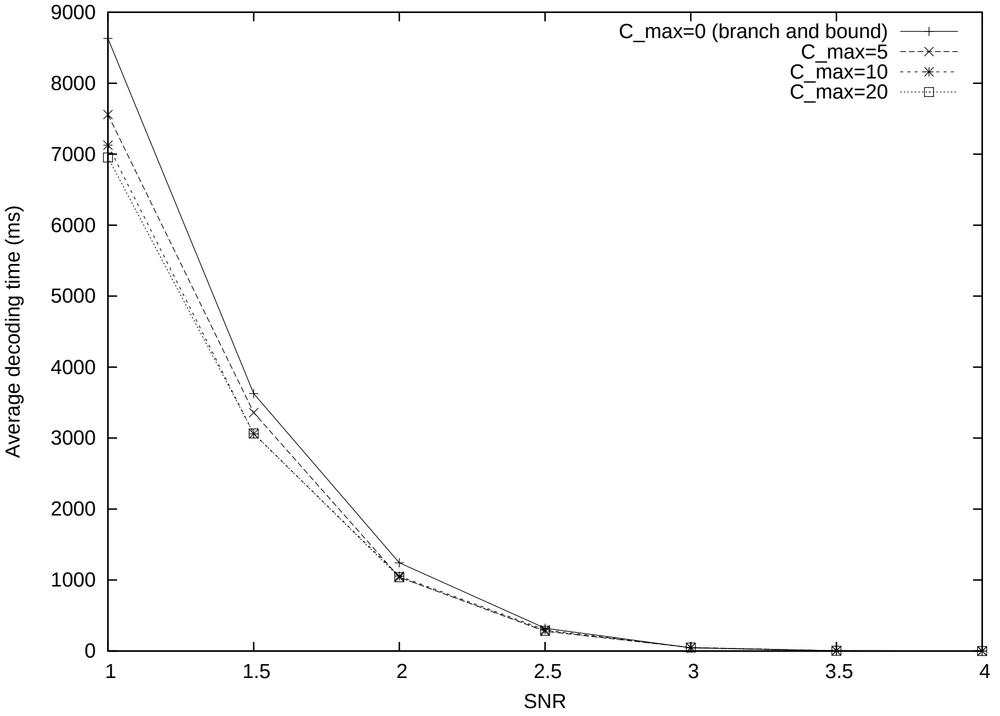

Figure 5.7: The efficiency of cutting-plane method (result for the the code 2).

is a possibility to reduce the time complexity further by making efforts, for example, use other LP solver, implement other cutting plane method, change the choice of branching variable, mix the searching strategy, and so on. It is a future work.

The algorithms used in this chapter are applicable for other probabilistic inference problems. We expect that these algorithms work well for such problems. The application for other problems are also a future work.

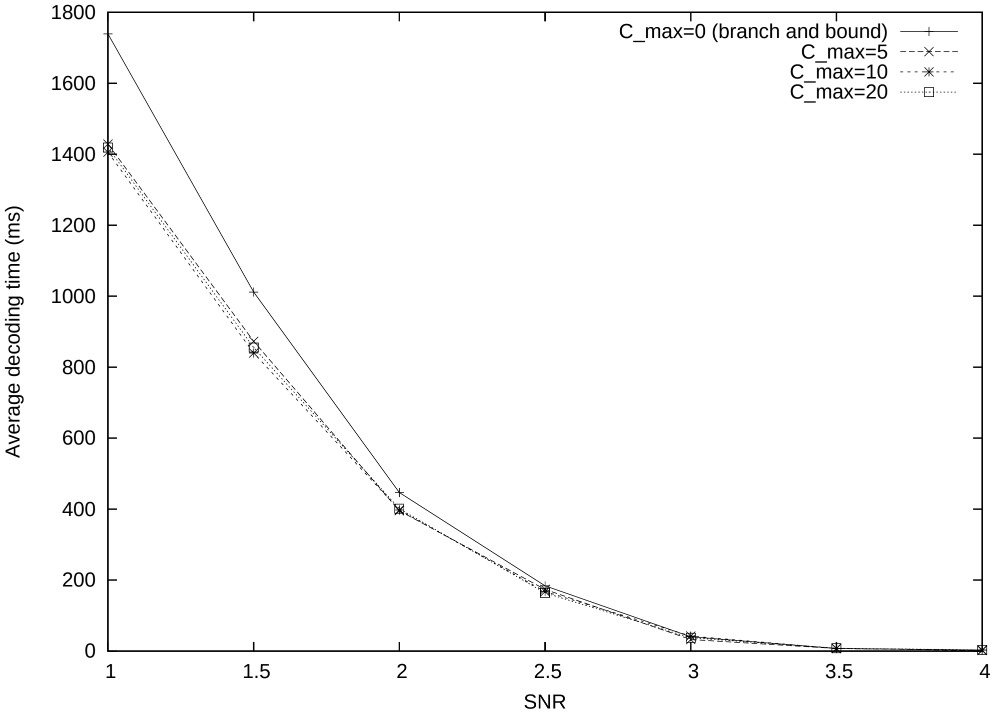

Figure 5.8: The efficiency of cutting-plane method (result for the the code 3).

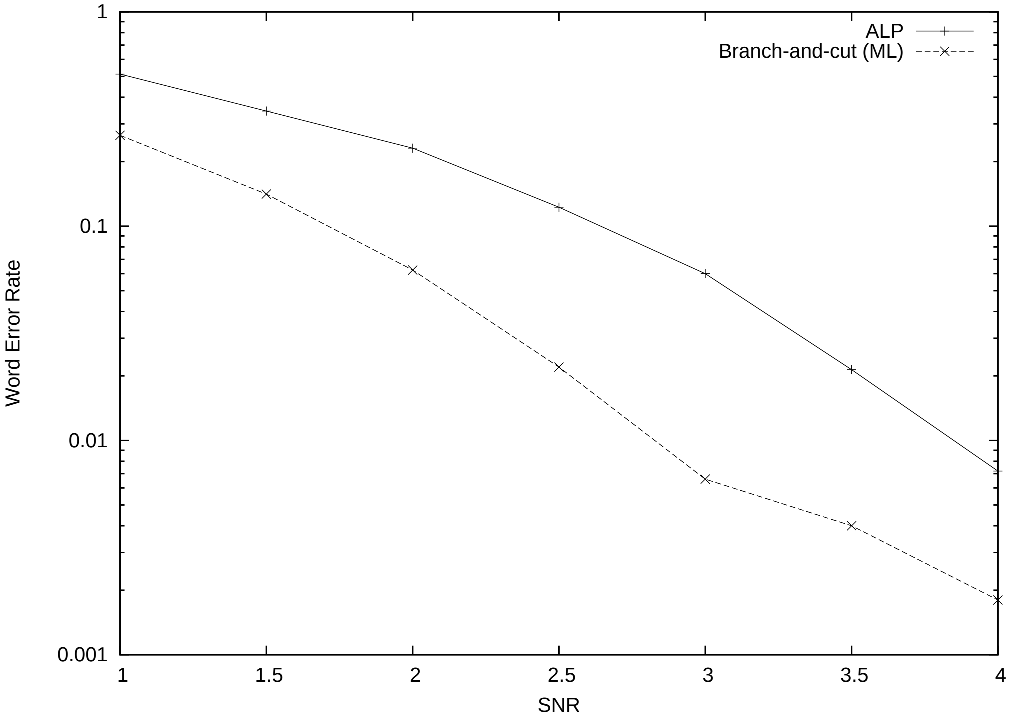

Figure 5.9: ALP and ML decoding word-error rates (WERs) for the code 1 as a function of SNR.

Chapter 6

Concluding Remarks and Future Works

6.1 Concluding Remarks

In this thesis, we consider the probabilistic inference problems for graphical models. These problems are very important since they arise in many applications. Finding the maximum a posterior probability (MAP) estimator and/or maximum a posterior marginal (MPM) estimator is often the main step in solving probabilistic inference problems. In general, the problem of obtaining the MAP and/or MPM estimator is computationally difficult unless the problem has a special structure. Therefore, several studies have been conducted on statistical models that have a high expression ability and a structure with which the MAP and/or MPM estimator can be found in a practical manner. Graphical models are widely used to capture the complex dependencies among random variables. It is also important to develop the algorithms that calculate the MAP and/or MPM estimator efficiently.

The contribution of this thesis to resolving the probabilistic inference problems are as follows:

(a) Development of the linear programming based inference algorithms for general probabilistic inference problems (Chapter 3)

In chapter 3, we extended the linear programming decoding algorithm, which is originally proposed as the decoding algorithm for binary linear codes over the single-user memoryless channel, to general probabilistic inference problems. If we regard the decoding problem of binary linear codes for the memoryless channel as an example of the probabilistic inference problem, the factor graph corresponding to the problem does not have any non-indicator functions connected to more than one variable node. On the other hand, the factor graph corresponding to the general probabilistic inference problem possibly has some such non-indicator functions. We showed that even if the factor graph has some non-indicator functions connected to more than one variable node, the LP-based inference algorithm can be applied by altering the graph structure. Based on this proposition, we applied the LP-based inference algorithm to various problems. As an example of the problem where the factor graph has some non-indicator functions connected to more than one variable node, we considered the decoding problem of the binary linear code over multiple-access channel.

(b) Reduction of the computational complexity of the inference algorithms (Chapter 4)

In chapter 4, we showed that we can reduce the computational complexity of the inference algorithms by changing the factorization structure of the posterior distribution. First, we proposed the new factorization structure for the decoding problem for the DS-CDMA model. It has been believed that it is impossible to implement the SP algorithm for the decoding problem for the DS-CDMA model, however, the new factorization enabled us to implement the SP algorithm for the problem. Second, we construct the factor graph for the Gaussian multiple-access channel. Since the Gaussian multiple-access channel is one of the particular model of the multiple-access channels, we can apply the method described in chapter 3, however, its computational complexity is the exponential order of the number of users. We show that we can construct the another factor graph for the decoding problem for Gaussian multiple-access channel and obtain the another linear programming formulation. The computational complexity to solve the obtained linear programming is the polynomial order of the number of users.

(c) The application of the combinatorial optimization algorithms for the probabilistic inference problems (Chapter 5)

In chapter 5, we studied the application of the combinatorial optimization algorithms for probabilistic inference problems. We dealt with the decoding problem of binary linear codes for single-user memoryless channel. In previous studies, application of the branch-and-bound and cutting plane methods had been proposed for solving the decoding problem. We proposed a decoding algorithm based on the branch-and-cut method, which is a hybrid of the branch-and-bound method and the cutting plane method. The numerical simulation results show that the performance of the proposed algorithm is better than that of previous algorithms.

6.2 Future Works

Here, we discuss some applications not considered or achieved in this study and which need to be addressed in future works.

(a) Application to various probabilistic inference problems

Originally, LP-based inference algorithm is proposed as an approximation algorithm for the maximum likelihood (ML) decoding of binary linear codes over memoryless channels. Based on the results of this thesis, we can apply the algorithm for other problems. However, it is not obvious whether the LP-based inference algorithm is always efficient. It has been shown in previous studies that the performance of the LP-based inference algorithm for the ML decoding of the linear codes over memoryless channels is almost the same as that of the SP algorithm. This thesis confirms that both algorithms have similar performance. According to these results, we expect that the LP-based inference algorithm works well for the problems that can be effectively handled by the SP algorithm. The SP algorithm is used for many applications such as image processing and resolving computer vision problems [20] [21] [22]. In the future, we plan to use the LP-based inference algorithm in these applications.

(b) Analysis of the performance of the inference algorithm

Recently, some analytical studies have been conducted on the performance of the LP-based inference algorithm for solving the decoding problem of the linear codes over memoryless channels [23] [24]. Contrary to the case of the SP algorithm, the performance of which was analytically studied only for the case when the code length is infinite [25], the performance of the LP-based algorithm was analytically studied for the case when the code length is finite. The finite code length analysis is one of the advantages of the LP-based algorithm over the SP algorithm. We need to study the performance of the LP based inference algorithm for many cases other than the decoding problem of the linear codes over the memoryless channels. For example, we need to guarantee the probability of the decoding error of the LP based decoding algorithm for linear codes over multiple-access channels, which is dealt with in chapters 3 and 4 of this thesis.

(c) Development of a sophisticated algorithm

In this thesis, we dealt with the LP relaxation for probabilistic inference problems. LP relaxation can be considered as one of the techniques to approximately solve the integer programming. There are many algorithms for approximately solving the integer programming, one of which is semi-definite programming (SDP) relaxation. The SDP relaxation based inference algorithm may possibly be effective in solving probabilistic inference problems. This algorithm is effective in solving the MAP decoding problem in the DS-CDMA model [26]. Hence, study on the inference algorithms based on the SDP seems a worthwhile endeavor.

Although we are not concerned with the LP solver, we consider it is important to develop a practical LP solver. In recent years, some efficient LP solvers for the decoding problem of linear codes over single-user memoryless channels have been proposed [16] [27]. These solvers utilize the structure of the problem. We aim to develop an efficient LP solver that is suitable for other problems.

Appendix A

Linear Programming Inference for Pairwise Markov Random Field

Wainwright et al. had proposed the LP-based inference algorithm for a certain class of the graphical model named Pairwise Markov Random Field (MRF) [28]. We can express the pairwise MRF by factor graph. We can say that the pairwise MRF is the model that corresponding factor graph has the property that degrees of all function nodes are at most 2. The objective function of the inference problem for the pairwise MRF is described as

$$g(\boldsymbol{x}) = \prod_{s\in V} f_s(x_s) \prod_{(s,t)\in E} f_{st}(x_s, x_t) \tag{A.1}$$

where $V = \{1, 2, \cdots, N\}$ and $E \subseteq V \times V$. As in the chapter 3 of the thesis, they define the functions $\phi_{s:\alpha},\ \phi_{st:\boldsymbol{\beta}}$ and the variables $\lambda_{s:\alpha},\ \lambda_{st:\boldsymbol{\beta}}$ as

$$\phi_{s:\alpha}(x_s) = \begin{cases} 1 & \text{if } x_s = \alpha \\ 0 & \text{otherwise} \end{cases} \quad \forall s \in V,\ \forall \alpha \in \mathcal{X}_s, \tag{A.2}$$

$$\phi_{st:\boldsymbol{\beta}}(x_s, x_t)(x_s, x_t) = \begin{cases} 1 & \text{if } (x_s, x_t) = \boldsymbol{\beta} \\ 0 & \text{otherwise} \end{cases} \quad \forall (s,t) \in E,\ \forall \boldsymbol{\beta} \in \mathcal{X}_s \times \mathcal{X}_t, \tag{A.3}$$

$$\lambda_{s:\alpha} = -\ln f_s(\alpha) \quad \forall s \in V,\ \forall \alpha \in \mathcal{X}_s, \tag{A.4}$$

$$\lambda_{st:\boldsymbol{\beta}} = -\ln f_{st}(\boldsymbol{\beta}), \quad \forall (s,t) \in E,\ \forall \boldsymbol{\beta} \in \mathcal{X}_s \times \mathcal{X}_t, \tag{A.5}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2)$. Then it satisfies

$$-\ln g(\boldsymbol{x}) = \sum_{s\in V} \sum_{\alpha\in\mathcal{X}_s} \lambda_{s:\alpha}\phi_{s:\alpha}(x_s) + \sum_{(s,t)\in E} \sum_{\boldsymbol{\beta}\in\mathcal{X}_s\times\mathcal{X}_t} \lambda_{st:\boldsymbol{\beta}}\phi_{st:\boldsymbol{\beta}}(x_s, x_t). \tag{A.6}$$

Therefore the problem to find $\boldsymbol{x}$ that maximizes $g(\boldsymbol{x})$ is equivalent to the problem to find $\boldsymbol{x}$ that minimizes (A.6). Wainwright et al. presented the relaxed LP problem defined as

$$\text{minimize} \sum_{s \in V} \sum_{\alpha \in \mathcal{X}_s} \lambda_{s:\alpha} \tau_{s:\alpha} + \sum_{(s,t) \in E} \sum_{\boldsymbol{\beta} \in \mathcal{X}_s \times \mathcal{X}_t} \lambda_{st:\boldsymbol{\beta}} \lambda_{st:\boldsymbol{\beta}} \tau_{st:\boldsymbol{\beta}} \tag{A.7}$$

$$\text{subject to} \sum_{\alpha \in \mathcal{X}_s} \tau_{s:\alpha} = 1, \quad \forall s \in V \tag{A.8}$$

$$\sum_{\beta_1 \in \mathcal{X}_s} \tau_{st:\boldsymbol{\beta}} = \tau_{t:\beta_2} \quad \forall (s,t) \in E,\ \forall \beta_2 \in \mathcal{X}_t \tag{A.9}$$

$$\sum_{\beta_2 \in \mathcal{X}_t} \tau_{st:\boldsymbol{\beta}} = \tau_{s:\beta_1} \quad \forall (s,t) \in E,\ \forall \beta_1 \in \mathcal{X}_s \tag{A.10}$$

They showed that if the solution of the LP is an integer, it is guaranteed to be optimal.

They also show that any factor graph with discrete random variables can be converted to an equivalent pairwise MRF. The idea used in their derivation is very similar to the way we took in chapter 3 of the thesis. It suffices to show that how to convert a function f_{123} defined on a triplet $\{x_1, x_2, x_3\}$ into a pairwise form. They introduced an auxiliary node A, and associate with it random variable $\boldsymbol{z}$ that takes value in the Cartesian product space $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$. Each configuration of $\boldsymbol{z}$ can be identified with a triplet (z_1, z_2, z_3). For each $s \in \{1, 2, 3\}$, they define a pairwise function ψ_{As} as $\psi_{As}(\boldsymbol{z}, x_s) = f_{123}(z_1, z_2, z_3)^{1/3} \mathbb{I}[z_s = x_s]$ (the function $\mathbb{I}[z_s = x_s]$ takes value 1 if and only if $z_s = x_s$). Then it satisfies

$$f_{123}(x_1, x_2, x_3) = \sum_{\boldsymbol{z}} \prod_{s=1}^{3} \psi_{As}(\boldsymbol{z}, x_s). \tag{A.11}$$

References

[1] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory*, 47(2):498–519, February 2001.

[2] D.J.C. MacKay. Good error-correcting codes based on very sparse matrices. *Information Theory, IEEE Transactions on*, 45(2):399–431, 2002.

[3] T. Tanaka and M. Okada. Approximate belief propagation, density evolution, and statistical neurodynamics for cdma multiuser detection. *IEEE Trans. Inf. Theory*, 51(2):700–706, 2005.

[4] M. K. Varanasi and B. Aazhang. Multistage detection in asynchronous code-division multiple-access communications. *IEEE Trans. Commun.*, 38(4):509–519, April 1990.

[5] A. Amraoui, S. Dusad, and R. Urbanke. Achieving general points in the 2-user Gaussian MAC without time-sharing or rate-splitting by means of iterative coding. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, page 334. IEEE, 2005.

[6] A. de Baynast and D. Declercq. Gallager codes for multiple user applications. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, page 335. IEEE, 2005.

[7] A. Roumy, D. Declercq, and E. Fabre. Low complexity code design for the 2-user gaussian multiple access channel. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 483. IEEE, 2005.

[8] J. Feldman, M. J. Wainwright, and D. Karger. Using linear programming binary linear codes. *IEEE Trans. Inf. Theory*, 51(3):954–972, March 2005.

[9] M. F. Flanagan, V. Skachek, E. Byrne, and M. Geferath. Polytope representations for linear-programming decoding of non-binary linear codes. In *In Proc. of Int. Symp. Inf. Thoery (ISIT2008)*, pages 1508–1512, July 2008.

[10] D. Bertsimas and J.N. Tsitsiklis. Introduction to linear optimization. 1997.

[11] S. Verdú. Minimum probability of error for asynchronous gaussian multiple-access channels. *IEEE Trans. Inf. Theory*, 32(1):85–96, January 1986.

[12] K. W. Yip and T. S. Ng. Code phase assignment - a technique for high capacity indoor mobile ds-cdma communications. In *In Proc. IEEE Vehicular Technology Conf. (VTC'94)*, pages 1586–1590, June 1994.

[13] M. Miwa, T. Wadayama, and I. Takumi. A cutting-plane method based on redundant rows for improving fractional distance. *IEEE Journal on Selected Areas in Commun.*, 27(6):1005–1012, August 2009.

[14] S. C. Draper, J. S. Yedidia, and Y. Wang. Ml decoding via mixed-integer adaptive linear programming. In *In Proc. of Int. Symp. Inf. Theory*, pages 1656–1660, Nice, France, June 2007.

[15] K. Yang, J. Feldman, and X. Wang. Nonlinear programming approaches to decoding low-density parity-check codes. *IEEE Journal on Selected Areas in Commun.*, 24(8):1603–1613, August 2006.

[16] M. H. Taghavi and P. H. Siegel. Adaptive linear programming decoding. In *In Proc. of Int. Symp. Inf. Thoery*, pages 1374–1378, Seatle, USA, July 2006.

[17] A. Tanatmis, S. Ruzika, H. Hamacher, M. Punekar, F. Kienle, and N. Wehn. Valid inequalities for binary linear codes. In *In Proc. of Int. Symp. Inf. Theory*, pages 2216–2220, Seoul, July 2009.

[18] A. Martin. General mixed integer programming : Computational issues for branch-and-cut algorithms. *Computational Combinatorial Optimization, Lecture Notes in Computer Science*, 2241:1–25, 2001.

[19] *GNU Linear Programming Kit.*

[20] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000.

[21] B.J. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In *NIPS*, pages 737–743, 2001.

[22] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1068–1080, 2007.

[23] N. Halabi and G. Even. LP decoding of regular LDPC codes in memoryless channels. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 744–748. IEEE, 2010.

[24] W. Meng, Weiyu X., and Ao T. The limits of error correction with lp decoding. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 749–753. IEEE, 2010.

[25] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *Information Theory, IEEE Transactions on*, 47(2):599–618, 2002.

[26] W.K. Ma, TN Davidson, KM Wong, ZQ Luo, and PC Ching. Quasi-Maximum-Likelihood Multiuser Detection using Semi-Definite Relaxation with Application to Synchronous CDMA. 2001.

[27] P. O. Vontobel. Interior-point algorithms for linear-programming decoding. In *In Proc. of. Inf. Theory and its Applications Workshop*, pages 433–437, UC San Diego, LA Jolla, Janurary 2008.

[28] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Publications

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on the Branch-and-Cut Approach to Decoding Linear Block Codes,” IEICE Trans. Fundamentals, Vol.E93-A, No.11, pp.1912-1917 Nov. 2010.

Tota Suko, Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”Bayes Universal Source Coding Scheme for Correlated Sources,” Proc. of the 1st IEEE African Winter School on Information Theory and Communications 2010, p.27, Mopani Camp, Kruger National Park, South Africa, June 2010.

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on the Branch-and-Cut Approach to Decoding Linear Block Codes,” Proc. of 2010 International Workshop on Nonlinear Circuits, Communication and Signal Processing (NCSP'10), pp.321-324, Hawaii, USA, March 2010.

Shunsuke Horii, Tota Suko, Toshiyasu Matsushima, Shigeichi Hirasawa, ”Multiuser Detection Algorithm for CDMA based on the Belief Propagation Algorithm,” Proc. of 2008 IEEE 10th International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'08), pp.194-199, Bologna, Italy, Aug. 2008.

Shunsuke Horii, Tota Suko, Toshiyasu Matsushima, Shigeichi Hirasawa, ”Multiuser Detection Algorithms for CDMA based on the Message Passing Algorithms,” (unrefereed) IEICE Tech Rep. substituted for Proc. of 2006 Hawaii, IEICE and SITA Joint Conference on Information Theory (HISC'06), pp.17-22, Nara, Japan, May 2006.

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on the Inference Algorithm on the Factor Graph based on the Linear Programming,” IEICE Tech. Rep., vol.IT2010-63, pp.55-60, Nara, Japan (2011-1).

Shunsuke Horii, Tota Suko, Toshiyasu Matsushima, Shigeichi Hirasawa, ”Maximum Likelihood Detection for DS-CDMA using Gröbner Bases,” Proc. of 2010 Symposium on Information Theory and its Applications (SITA2010), pp.489-493, Nagano (2010-12).

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”Linear Programming Decoding of Binary Linear Codes for Multiple-Access Channel,” IEICE Tech. Rep., vol.IT2010-39, pp.31-36, Miyagi, Japan (2010-9).

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on the Branch-and-Cut Approach to Decoding Linear Block Codes,” Proc. of 2009 Symposium on Information Theory and its Applications (SITA2009), pp.569-573, Yamaguchi (2009-12).

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on the Iterative Interference Cancellation and Decoding for Coded CDMA,” Proc. of 2008 Symposium on Information Theory and its Applications (SITA2008), pp.66-70, Tochigi (2008-10).

Shunsuke Horii, Tota Suko, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on Multiuser Detection Algorithms for CDMA based on the Belief Propagation Algorithm,” IEICE Tech. Rep., vol.IT2007-26, pp.7-12, Tokyo, (2008-1).

Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”A Note on Resilient network coding in the presence of eavesdropping,” Proc. of 2007 Symposium on Information Theory and its Applications (SITA2007), pp.742-745, Mie (2007-11).

Shunsuke Horii, Tota Suko, Toshiyasu Matsushima, ”Bayes Optimal Multi User Detection for DS/CDMA Systems with Time-Varying Group of Active Users,” Proc. of 2005 Symposium on Information Theory and its Applications (SITA2005), pp.781-784, Okinawa, (2005-11).

Tota Suko, Shunsuke Horii, Toshiyasu Matsushima, Shigeichi Hirasawa, ”Bayes Coding for Multiple Correlated Sources,” Proc. of 2010 Symposium on Information Theory and its Applications (SITA2010), pp.759-763, Nagano (2010-12).

Daiki Koizumi, Shunsuke Horii, Toshiyasu Matsushima, ”On the Web Traffic Modeling under the Openin and Closing Services by the Non-stationary Poisson Process,” IEICE Tech. Rep., vol.IN2009-201, pp.392-396, Miyazaki, (2010-3).

Shinya Nishimura, Shunsuke Horii, Toshiyasu Matsushima, ”State-Based Modeling and Lossless Coding of Images,” 2007 Symposium on Information Theory and its Applications (SITA2007), pp.392-396, Mie, (2007-11).

Koichi Suzuki, Shunsuke Horii, Toshiyasu Matsushima, ”A Note on Nonstationary Poisson Model with the Consideration of Change Factors,” IEICE Tech. Rep., vol.IN2006-176, pp.83-88, Aichi, (2007-2).