

**Joint Source-Channel
and Multiple Description Coding**

NEW ADVANCES IN JOINT SOURCE-CHANNEL
AND MULTIPLE DESCRIPTION CODING

By

XIAOHAN WANG, M.Eng., B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE

DOCTOR OF PHILOSOPHY

McMaster University

© Copyright by Xiaohan Wang, January 2009

DOCTOR OF PHILOSOPHY (2009)
(Electrical and Computer Engineering)

McMaster University
Hamilton, Ontario

TITLE: New Advances in Joint Source-Channel and Multiple
Description Coding

AUTHOR: Xiaohan Wang
M.Eng., B.Eng., (Electrical Engineering)
Xi'an Jiaotong University, Xi'an, China

SUPERVISOR: Dr. Xiaolin Wu, Dr. Sorina Dumitrescu

NUMBER OF PAGES: xv, 114

Dedicated to Zheping

Abstract

This thesis launches some new inquiries and makes significant progress in the active research areas of joint source-channel coding and multiple description coding. Two interesting but previously untreated problems are investigated and partially settled: 1) can index assignment of source codewords be optimized with respect to a given joint source-channel decoding scheme, and if so, how? 2) can joint source-channel coding be optimized with respect to a given multiple description code, and if so, how?

The first problem is formulated as one of quadratic assignment. Although quadratic assignment is NP-hard in general, we are able to develop a near-optimum index assignment algorithm for joint source-channel (JSC) maximum *a posteriori* (MAP) decoding, if the input is a Gaussian Markov sequence of high correlation. For general cases, good heuristic solutions are proposed. Convincing empirical evidence is presented to demonstrate the performance improvement of the index assignments optimized for MAP decoding over those designed for hard-decision decoding.

The second problem is motivated by applications of signal communication and estimation in resource-constrained lossy networks. To keep the encoder complexity at a minimum, a signal is coded by a multiple description quantizer (MDQ) without channel coding. The code diversity of MDQ and the path diversity of the network are exploited by decoders to combat transmission errors. A key design objective is resource scalability: powerful nodes in the network can perform JSC-MD estimation under the criteria of maximum *a posteriori* probability or minimum mean-square error (MMSE), while primitive nodes resort to simpler multiple description (MD) decoding,

all working with the same MDQ code. The application of JSC-MD to distributed estimation of hidden Markov models in a sensor network is demonstrated. The proposed JSC-MD MAP estimator is an algorithm of the longest path in a weighted directed acyclic graph, while the JSC-MD MMSE decoder is an extension of the well-known forward-backward algorithm to multiple descriptions. They outperform the existing hard-decision MDQ decoders by large margins.

Acknowledgement

I would like to take this opportunity to express my sincere gratitude to many people who made the completion of this thesis possible.

First and foremost, I would like to thank my supervisors: Prof. Xiaolin Wu and Prof. Sorina Dumitrescu. I am really fortunate to have the opportunity to be guided by them. I was greatly influenced by Prof. Wu by his endless enthusiasm, vigor, and dedication to research. His active mind, inspiring discussions and helpful criticism initiated this work at the very beginning and has kept improving it ever since. Prof. Dumitrescu has very strong background in mathematics. Several analytical parts of this thesis have benefited directly from her. As a rigorous scholar, she pointed out many mistakes I have made. I thank her for her wit, patience and encouragement.

I would also like to acknowledge Prof. James P. Reilly and Prof. Shahin Sirouspour for serving my supervisory committee. Their insightful suggestions and critical comments throughout the supervision have greatly improved this thesis and are highly appreciated. I would like to thank in particular Prof. Michael Brook and Prof. John Luxat for being on my examination committee and Prof. Khalid Sayood for being my external examiner.

My gratitude also goes to my fellow collaborators at the Multimedia Signal Processing Laboratory Ning, Zhe, Xiangjun, Xiang, Gang, Mingkai, Huazhong, Amin and Reza, for generously sharing their advice, knowledge and happiness. Their help and friendship have made my doctoral study an unforgettable experience. I would also like to thank my friends: Yan, Kai, Lin and Jun, who made my life outside the research world also colorful.

Last but by no means least, I would like to thank my parents, Zhangshun Wang and Fangli Zhou, for their unconditional love and support. I am also deeply indebted to my wife Zheping for her love, sacrifice and patience with me. This thesis is dedicated to her. Also thank my son Julien, who kept distracting me from writing this thesis, for the unparalleled joy he brought us.

最后我要感谢我的父亲王章顺和我的母亲周芳莉，感谢他们对我无条件的支持和爱。他们的养育之恩我没齿难忘。

Contents

Abstract	iv
Acknowledgement	vi
Contents	x
List of Figures	xii
List of Tables	xiii
List of Abbreviations	xiv
1 Introduction	1
1.1 Joint Source-Channel Coding	1
1.2 Motivation	3
1.3 Original Contributions	8
1.4 Organization of Thesis	10
1.5 Related Publications	11
2 Background and Literature Review	12
2.1 Joint Source-Channel MAP decoding	12
2.2 Binary Index Assignment	15
2.3 Multiple Description Coding	20

3	Index Assignment for Joint Source-Channel MAP Decoding	24
3.1	Overview	24
3.2	Problem Formulation	26
3.3	Index Assignment Structures and the Source Memory	32
3.3.1	Flow Matrix for Memoryless Sources	32
3.3.2	Flow Matrix for Gaussian Markov Sources	33
3.3.3	Flow Matrix and Source Memory	34
3.4	Index Assignment for Gaussian Markov Sources	38
3.5	Simulated Annealing Algorithm	41
3.6	Simulation Results	41
3.7	Conclusion	44
4	Joint Source-Channel Decoding of Multiple Descriptions	50
4.1	Overview	50
4.2	Problem Formulation	52
4.3	Joint Source-Channel Multiple Description MAP Decoding	57
4.4	Complexity Reduction of JSC-MD MAP Decoding Algorithm	59
4.5	Distributed Multiple-Description Estimation of Hidden Markov Sequences	61
4.5.1	Problem Setup	61
4.5.2	Distributed MAP Sequence Estimation	63
4.5.3	Resource Scalability	65
4.5.4	Application in Distributed Speech Recognition	66
4.6	Resource-scalable JSC-MD MMSE Decoding	67
4.7	Simulation Results	70
4.8	Conclusions	73
5	Joint Source-Channel MAP Decoding of Variable-Length Coded Multiple Descriptions	76
5.1	Overview	76

List of Figures

1.1	The system diagram of this thesis.	3
1.2	Heterogeneity in consumer-oriented wireless networks.	4
2.1	The Hamming distance matrix and adjacency matrix of NBC ($N = 16$).	16
2.2	The central and side quantizers of the example MDQ.	22
2.3	The index assignment for two two-description scalar quantizer proposed by Vaishampayan, where d denotes the number of diagonals occupied by the central quantizer indices.	23
3.1	System diagram.	25
3.2	The flow matrix of the optimal index assignment problem ($N = 32$) for five Gaussian Markov Sources: (a) $\rho = 0.9$, Hamming distortion. (b) $\rho = 0.5$, Hamming distortion. (c) $\rho = 0.9$, MSE. (d) $\rho = 0.5$, MSE. (e) $\rho = 0$, MSE and (f) a memoryless uniform source, MSE.	36
3.3	The adjacency matrix of various index assignments ($N = 16$): (a) Gray code, (b) Natural binary code, (c) Folded binary code, (d) HarperA code, (e) HarperB code, and (f) code found by simulated annealing algorithm for Gaussian Markov source ($\rho = 0.9$) and MSE, with decimal form (12 9 4 3 2 10 14 15 13 5 1 0 8 6 11 7).	37
3.4	Evolution of $C(\pi)$ in the process of simulated annealing algorithm for a non-Gaussian Markov source ($N = 8$).	45

4.1	Block diagram of a MDQ based communication system with a JSC-MD decoder.	53
4.2	Graph G constructed for the JSC-MD MAP decoding ($L = 5$).	58
4.3	JSC-MD sequence estimation in sensor network.	62
4.4	Symbol error rates of JSC-MD distributed MAP decoder and MDQ hard-decision decoder with $\rho = 0, 0.5, 0.9$	71
4.5	SNR performances of different MDQ decoders ($\rho = 0$).	73
4.6	SNR performances of different MDQ decoders ($\rho = 0.5$).	74
4.7	SNR performances of different MDQ decoders ($\rho = 0.9$).	75
5.1	Block diagram of a MDSQ based communication system with a MAP decoder.	77
5.2	Graph G constructed for the joint source-channel MDQ decoding ($K = 2$, $L = 6$, $\mathbb{C}_1 = \{0, 10, 11\}$, $\mathbb{C}_2 = \{0, 10, 110, 111\}$, $N_1 = 7$ and $N_2 = 9$).	82
5.3	SNR performances of different MDQ decoders ($\rho = 0.1$), where d is the number of diagonals in the 2DSQ index assignment matrix.	84
5.4	SNR performances of different MDQ decoders ($\rho = 0.5$), where d is the number of diagonals in the 2DSQ index assignment matrix.	85
5.5	SNR performances of different MDQ decoders ($\rho = 0.9$), where d is the number of diagonals in the 2DSQ index assignment matrix.	86

List of Tables

2.1	Various index assignments in both decimal (in bold) and binary form ($N = 16$).	18
3.1	MAP decoding with index assignment π_1	28
3.2	MAP decoding with index assignment π_2	29
3.3	The best and worst codes found by exhaustive simulation for SER and MSE ($N = 8, p_c = 0.001$), and the code found by the simulated annealing algorithm for MSE ($N = 8$)	42
3.4	SER of different index assignments for Gaussian Markov source of correlation coefficient $\rho = 0.9$ ($N = 8, 16, 32, 64, 128, 256$)	46
3.5	Index assignments found by the simulated annealing algorithm for Gaussian Markov sources with correlation coefficient $\rho = 0.9$ and for MSE ($N = 16, 32, 64, 128, 256$).	47
3.6	MSE of different index assignments for Gaussian Markov source of correlation coefficient $\rho = 0.9$ ($N = 8, 16, 32, 64, 128, 256$)	48
3.7	Symbol error rate of different index assignments for a non-Gaussian Markov source ($N = 8$)	49
6.1	MEPL of various VLC index assignments for Gaussian Markov source of correlation coefficient $\rho = 0.9$ when $p_c = 0.01$ ($N = 5, 6, 7, 8, 9$) . . .	90

List of Abbreviations

2DSQ	Two Description Scalar Quantization
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase-Shift Keying
BSC	Binary Symmetric Channel
CPU	Central Processing Unit
DPCM	Differential (or Delta) Pulse-Code Modulation
EEC	Error-and-Erasure Channel
FBC	Folded Binary Code
FEC	Forward Error Protection
FLC	Fixed-Length Code
GC	Gray Code
HMM	Hidden Markov Model
JSC	Joint Source-Channel
JSCC	Joint Source-Channel Coding
MAP	Maximum <i>A Posteriori</i>
MD	Multiple Description
MDC	Multiple Description Coding
MDLVQ	Multiple Description Lattice Vector Quantizer
MDQ	Multiple Description Quantization

MDSQ	Multiple Description Scalar Quantization
MDVQ	Multiple Description Vector Quantization
MEPL	Mean Error Propagation Length
MMSE	Minimum Mean Squared Error
MPEG	Motion Picture Experts Group
MSE	Mean Squared Error
NBC	Natural Binary Code
PDA	Personal Data Assistance
QAP	Quadratic Assignment Problem
SA	Simulated Annealing
SER	Symbol Error Rate
SNR	Signal-to-Noise Ratio
VLC	Variable-Length Code
VLCMD	Variable-Length Coded Multiple Description
WDAG	Weighted Directed Acyclic Graph

Chapter 1

Introduction

1.1 Joint Source-Channel Coding

Shannon's separation theorem of source and channel coding for point-to-point communication states that *source coding* (data compression) and *channel coding* (error protection) can be performed separately without sacrificing the optimality of the whole system [1]. The purpose of the source coding is to reduce the number of bits needed to represent a signal by removing the deterministic component (redundancy) from the signal. The channel coding provides reliable transmission of data in the presence of noise by introducing structured redundancy. The importance of the separation theorem is that the source code and the channel code can be optimized independently, which greatly simplifies the design of a communication system.

This theorem, however, does not hold for all communication systems. It does not hold for multiuser communication. Moreover, for point-to-point communication, the theorem assumes that the source and the channel coder are both optimal. This assumption can only be achieved with unlimited complexity and delay at the encoder, which is unrealistic for any practical applications. Furthermore, the channel code may break down completely if the channel condition degrades to a certain level, which makes the system unsuitable for varying channel conditions. In these situations it is advantageous to design the source and channel codes jointly, as witnessed by a large

body of literature on *joint source channel coding* (JSCC) [2, 3, 4, 5]. In an article by Sayood et al. [4], JSCC was classified into four broad categories (for a full list of related references please see the paper):

Joint source channel coders: the source and channel coding operations are truly integrated.

Concatenated source/channel coders: the encoder cascades known source coders and known channel coders, and allocates the fixed bit rate between the source coder and the channel coder to maximize the system performance.

Unequal error protection: the output of the source encoder is afforded unequal protection based on the effect of errors on the reconstructed sequence.

Constrained joint source/channel coding: the source coder and/or receiver are modified to account for the presence of a given noisy channel. This class includes two subclasses:

1. When the channel code is not perfect, the source coder needs to be optimized subject to a noisy channel constraint, e.g. the *binary index assignment* optimization.
2. When the source code is not perfect, there might be some redundancy left in the output of the source coder. The joint source-channel (JSC) decoder tries to use some knowledge of the source or source coder properties to detect channel errors and compensate for their effects. For example, the JSC maximum *a posteriori* (MAP) decoding tries to find the input sequence that maximizes the *a posteriori* probability given the received sequence.

Similar classification of JSCC approaches can also be found in the introductory part of [5]. For a detailed review of various techniques encompassed by the concept of JSCC, see [2].

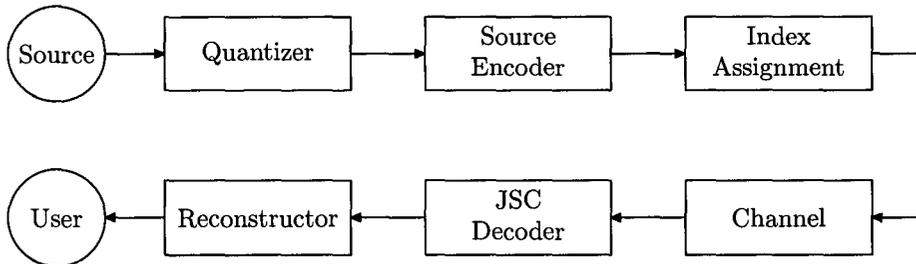


Figure 1.1: The system diagram of this thesis.

In this thesis, we focus on the last category of JSCC techniques and their utilization in resource-aware networked signal communication as outlined in the next section.

1.2 Motivation

The work flow of a signal communication system that employs JSCC techniques but without forward error correction (FEC) coding is shown in Figure 1.1. We are interested in lossy communication systems that have resource-deprived transmitters and receivers of varied capabilities. Such scenarios are common in sensor networks and wireless networks. For instance, a large number of inexpensive sensors with no or low maintenance are deployed to monitor, assess, and react to a large environment. On one hand these sensors have to conserve energy to ensure a long lifespan, and on the other hand they need to communicate with processing centers and possibly also among themselves in volatile and adverse network conditions. The energy budget and equipment level of the receivers vary greatly, ranging from powerful processing centers to deprived sensors themselves. Heterogeneity is also the norm in consumer-oriented wireless networks. A familiar and popular application is multimedia streaming with mobile devices such as handsets, personal data assistance (PDA), and notebook computers, as depicted in Figure 1.2. Here power consumption is again a primary constraint for all mobile data transmitters, while its criticality varies for receivers, depending on whether the receivers are cell phones, notebooks, or base

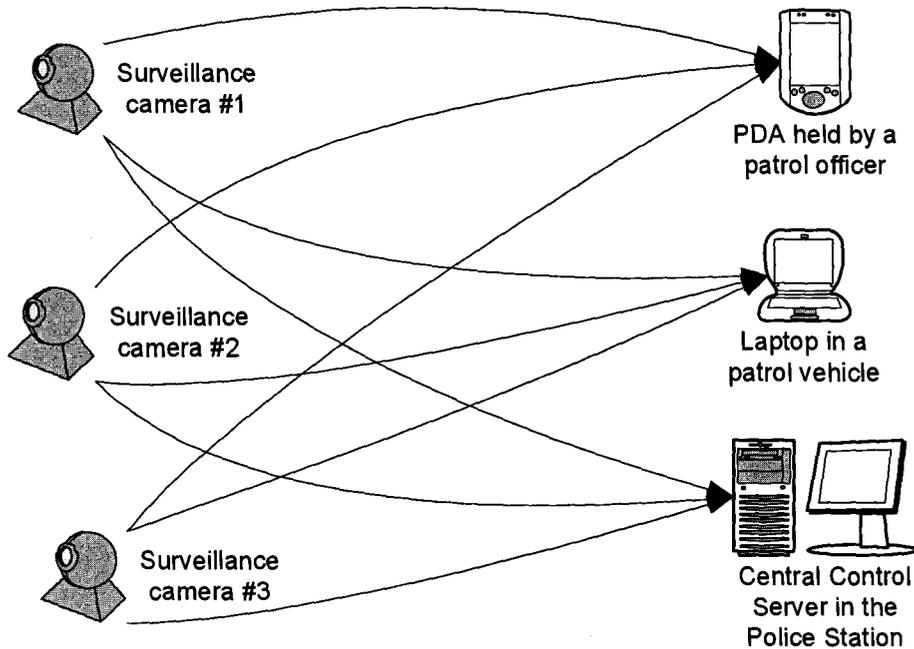


Figure 1.2: Heterogeneity in consumer-oriented wireless networks.

stations, etc.

Conventional source and channel coding techniques may not be good choices for networks of resource-constrained nodes, because they make coding gains proportional to computational complexity (hence energy consumption). The needs for power-aware signal compression techniques have generated renewed interests in the theory of Slepian-Wolf and Wyner-Ziv coding, which was developed more than thirty years ago [6, 7]. The key insight of these works is that statistically dependent random sources can be encoded independently without loss of rate-distortion performance, if the decoder has the knowledge or side information about such dependencies. Although originally intended for distributed source coding, the approach of Slepian-Wolf and Wyner-Ziv coding is of significance to resource-constrained compression in two aspects:

1. communication or coordination between the encoders of the different sources is not necessary to achieve optimal compression, even if the sources are statistically dependent, saving the energy to communicate between the encoders;

2. it is possible to shift heavy computation burdens of rate-distortion optimal coding of dependent sources from encoders to decoders.

Such an asymmetric codec design provides an attractive signal compression solution in situations where a large number of resource-deprived and autonomous encoders need to communicate multiple statistically dependent sources to one or more capable decoders, as is the case for some hierarchical sensor networks [8].

Recently, many researchers have been enthusiastically investigating practical Wyner-Ziv video coding schemes [9, 10], seeking for energy-conserving solutions of video streaming on mobile devices. The motive is to perform video compression without computationally expensive motion compensation at the encoder, departing from the prevailing MPEG (motion picture experts group) practice. Instead, the decoder is responsible to exploit the interframe correlations to achieve coding efficiency.

While Wyner-Ziv coding can shift computational complexity of signal compression from encoders to decoders, it does not address another characteristic of modern communication networks: varied resource levels at different nodes. As mentioned earlier, decoders can differ greatly in power supply, bandwidth, computing capability, response time, and other constraints. What can be done if a decoder has to operate under severe resource constraints as well? Despite the information theoretical promise of Wyner-Ziv coding, the rate-distortion performance of distributed compression is operationally bounded by the intrinsic complexity of the problem, or equivalently by the energy budget. It is well known that optimal rate-distortion compression in centralized form is NP-hard [11]. We have no reason to believe that approaching the Wyner-Ziv limit is computationally any easier.

Given the conflict between energy conservation and coding performance, it is desirable to have a versatile signal coding and estimation approach whose performance can be scaled to available energy. This resource scalability is a key notion of this thesis. Our design principle is to keep the complexity of the encoders (often synonymously sensors in sensor networks) at minimum, while allowing a wide range of trade-offs between the complexity and rate-distortion performance at decoders.

In this thesis the source signal is modeled as a Markov process. To keep its complexity at a minimum, a resource-deprived encoder quantizes the signal, applies fixed-length or a simple variable-length code (e.g., Huffman code) to quantizer output, and then transmits without forward error protection. The quantizer can be either single description or multiple descriptions of simple structure (see Section 2.3 for a short introduction to multiple description quantizers). Before transmission an index assignment is given to the codewords of the fixed-length or variable-length code. The optimization of the index assignment with respect to the channel statistics and the chosen JSC decoder is a main topic of this thesis.

The expediency of the inexpensive encoder furnishes the decoder with rich forms of statistical redundancy:

- the memory of the Markov sequence that is unexploited due to suboptimal quantization;
- the residual source redundancy for lack or for the suboptimality of the entropy coding;
- in case of multiple description quantization the redundancy introduced among different descriptions.

These residual redundancies left in the output of the encoder, together with the prior knowledge of the channel, can be exploited by a joint source-channel decoder to mitigate channel errors.

Depending on the availability of energy, bandwidth, central processing unit (CPU) power, and other resources, different receivers should be able to estimate *the same coded signal(s)* on a best effort basis. If a JSC decoder is not confined by computation power or energy budget, it can perform full fledged maximum *a posteriori* sequence estimation. Using both the memory of the Markov sequence and the knowledge of the channel statistics, the JSC MAP decoder infers the input sequence as the one of maximal *a posteriori* probability. Another powerful JSC decoding technique is

the minimum mean-squared error sequence estimation, which minimizes the mean-squared error on every symbol rather than maximizing a sequence probability. On the other hand, resource-deprived decoders can resort to various approximation solutions to reduce the computational and/or space complexity. The reduction is achieved by using only a part of the redundancies available, or by sacrificing the optimality in the decoding process. To the extreme, decoders of minimum amount of resources can simply accept whatever are received at the moment and perform symbol-by-symbol hard-decision decoding.

In our system design, JSC decoders of different performance-cost tradeoffs all operate on the same set of code streams. Ideally, a resource-scalable code should not deny a decoder without resource constraint the possibility of approaching the Wyner-Ziv performance limit, and at the same time it should allow even the least capable decoder in the network to reconstruct the signal, unless there is a complete transmission failure. By not producing different codes of a source or a set of sources according to different decoder specifications, the encoders also reduce energy consumption. Furthermore, the proposed approach simplifies and modularizes the encoder design, lowering the manufacturing cost.

In the above described asymmetric communication systems, the JSC decoder is not an isolated component from a rate-distortion perspective. Its design needs to be optimized with respect to the techniques adopted by other components of the system. In fact, the aforementioned classification of JSCC techniques is by no means a rigid boundary. Different JSCC techniques can be used in combination in a system, and their interactions can affect the overall system performance. Even within the fourth class itself the techniques in the two subclasses can have intricate interplays when used in conjunction. For instance, optimal index assignment varies not only with source and channel statistics, but also with the chosen decoding algorithms. Another example is that the JSC MAP decoder has very different designs depending on whether the source is coded into single or multiple descriptions.

The main thrust of this thesis is to investigate and solve the aforementioned type of

problems related to JSCC optimization across different system components. This type of problem was studied for two relatively simple cases. By far the most thoroughly researched case is optimal index assignment for hard-decision symbol-by-symbol decoding. There are few classical results for some sources and distortion metrics. More recently, quite few authors also studied the problem of JSC MAP decoding of fixed-rate or variable-rate coded Markov sequences. For a comprehensive review of these published techniques, please refer to Chapter 2. However, few more complex and difficult cases of the above type of problems still remain largely untreated, although they occur in many important application scenarios as identified earlier. In this thesis we will investigate these unsolved problems and propose a family of algorithmic solutions of various performance-cost tradeoffs.

1.3 Original Contributions

One of the contributions of this thesis is the above briefed resource-scalable asymmetric communication strategy. In the technical aspect, this thesis is the first study of its kind on the interplays between JSC soft-decision sequence estimation techniques and network-aware source coding techniques. It advances the state of the art in joint source-channel coding by answering the following three open questions.

1. How to optimize index assignment of source codewords for JSC MAP decoding?
2. How to perform JSC MAP decoding of multiple description coded sources?
3. How to perform JSC MMSE decoding of multiple description coded sources?

Despite its practical significance the first question has largely been neglected by researchers in the field of JSCC. This is quite a surprise considering that channel-optimized quantizer index assignment and MAP decoding have been extensively studied in their own right as error-resilient communication techniques. In this thesis we formulate the first problem as one of quadratic assignment, and develop solutions for various sources and distortion metrics. The developed solutions vary in complexity

depending on the algorithm generality ranging from the most general to some special cases. Experimental results demonstrate that the index assignments optimized for JSC MAP decoding improves the performance considerably over those designed for hard-decision decoding.

For a Gaussian Markov source of high correlation and the Hamming distortion, the optimal index assignment problem is shown to be equivalent to the antibandwidth problem of the hypercube. By investigating the recursive structure of the so-called Hales numbering, we derive the exact value of the hypercube antibandwidth, which is a new interesting result in its own right. Our study of the recursive structure of the Hales numbering also leads to a very simple proof of the hypercube bandwidth formula. The formula has been known and used for more than four decades in discrete mathematics and computer science, but its formal proof was thought to be “surprisingly difficult”.

To answer the second and third questions we extend the scope of JSC MAP decoding from single description to multiple descriptions. We introduce a novel joint source-channel multiple description (JSC-MD) framework for signal estimation and communication. The new JSC-MD framework offers an effective means of achieving resource scalability: powerful nodes in the network can perform JSC-MD estimation under the criteria of MAP or MMSE, while primitive nodes resort to simpler multiple description (MD) decoding, all working with the same MD code. We also discuss the application of JSC-MD to distributed estimation of hidden Markov models in a sensor network.

The proposed JSC-MD MAP estimator is an algorithm of the longest path in a weighted directed acyclic graph (WDAG), while the JSC-MD MMSE decoder is an extension of the well-known forward-backward algorithm [12] to multiple descriptions. Both algorithms simultaneously exploit the source memory, the redundancy of the fixed-rate multiple description quantization (MDQ) and the inter-description correlations. They outperform the existing hard-decision MDQ decoders by large margins (up to 8dB). Either algorithm has a computational complexity of $O(L^2NK)$,

where L is the codebook size of the central quantizer, N is the input sequence length in symbols and K is the number of MDQ descriptions. For Gaussian Markov sources, the complexity of JSC-MD distributed MAP sequence estimation can be made as low as that of typical single description Viterbi-type algorithms [13].

The proposed JSC-MD framework can also be applied to variable-rate MDQ. The JSC-MD MAP decoder exploits both the source memory and the correlation between different MDQ descriptions to combat the channel noise. It can also be modeled and solved as the longest path problem in a WDAG. The difference between this graph and the one constructed for fixed-rate MDQ is that 1) the former is bit-based while the latter is symbol-based, and 2) the former is a $K + 1$ dimensional hyper-trellis while the latter is always a two dimensional trellis. The computational complexity of the proposed algorithm is $O(L(L + K) \prod_{k=1}^K N_k)$, where $N_k, 1 \leq k \leq K$, is the number of bits in the received sequence from the k^{th} channel. If the source sequence is Gaussian Markov, the decoder complexity can be reduced to $O((L + K) \prod_{k=1}^K N_k)$. For variable-rate MDQ compressed Markov sequences impaired by both bit errors and erasure errors, the proposed joint source-channel MAP decoder can achieve 5dB higher signal-to-noise ratio (SNR) than the conventional hard-decision decoder.

Furthermore, the new JSC-MD framework enjoys an operational advantage over the existing MDQ decoders. It unifies the treatments of different subsets of the K descriptions available at the decoder, circumventing the thorny issue of requiring up to $2^K - 1$ MDQ side decoders, which has been a major obstacle for the practical use of MDQ.

1.4 Organization of Thesis

This thesis consists of six chapters. The current chapter serves as a brief introduction, stating the motivation, contribution and organization of this thesis. The rest of the thesis is organized as follows. **Chapter 2** presents a review and background information on technical topics related to this thesis, including joint source-channel MAP

decoding, binary index assignment and multiple description coding. In **Chapter 3**, we investigate the problem of optimal index assignment design for joint source-channel MAP decoding. In **Chapter 4**, the JSC-MD framework is proposed to estimate the Markov sequence coded by fixed-rate MDQ and transmitted over noisy channels. **Chapter 5** extends the work of Chapter 4 to the variable-rate MDQ. It can also be regarded as an extension of the traditional JSC MAP decoding of variable-length coded Markov sequences to the multiple description realm. **Chapter 6** concludes this thesis and outlines some future research topics.

In Chapter 3 we introduce the index assignments that solve the bandwidth and anti-bandwidth problems of the hypercube. We present the proof of the hypercube bandwidth and anti-bandwidth formulae in **Appendix A**. The proof procedure also reveals the intrinsic structures of these index assignments.

1.5 Related Publications

The contents of this thesis are also contained in five conference and three journal papers.

The contents of Chapter 3 have been presented in part in [14, 15] and was submitted to *IEEE Transactions on Communications* [16]. The work in Chapter 4 was first published in [17] and then extended in [18]. It is summarized in [19], which will appear in *IEEE Transactions on Signal Processing*. The JSC MAP decoding of variable-length coded multiple descriptions presented in Chapter 5 was previously published in [20]. The proof of the hypercube bandwidth and antibandwidth formulae that is given in Appendix A has been revised and resubmitted to the *Journal of Discrete Applied Mathematics* [21].

Chapter 2

Background and Literature Review

This section briefly introduces three key components in the framework of this thesis: joint source-channel MAP decoding, binary index assignment and multiple description coding. Previous works on these subjects are reviewed.

2.1 Joint Source-Channel MAP decoding

Joint source-channel decoding capitalizes on any redundancy left in the source code to repair channel impairments. Due to various operational reasons, such as the complexity and/or energy budget of the encoder, a practical source code is far from being rate-distortion optimal, leaving residual redundancies intentionally or involuntarily. For example, it was shown in [22] that the signal obtained by encoding an image with the differential (or delta) pulse-code modulation (DPCM) is not independent and identically distributed (i.i.d.). The resulting compressed image can be modeled as a Markov process because of the residual source redundancy. In addition, the source code may be unprotected by forward error correction code due to lack of computing resources or/and energy at the encoder side. In such scenarios, the decoder can exploit the residual source redundancy to the fullest to minimize the damages of channel errors. This is the idea of joint source-channel soft-decision sequence decoding (estimation) as opposed to the traditional hard-decision symbol by symbol decoding.

Obviously, the former approach is computationally far more expensive than the latter, but it, on the other hand, shifts the system complexity from the encoder to the decoder. This asymmetric system design in favor of inexpensive encoders is desirable for many modern communication applications, in which the decoder operates at a much higher resource level than the encoder. Such examples can be found in deep space communications where satellite ground stations are far more richly equipped than the satellites themselves. Similar trade-off between encoder and decoder complexities also makes sense in wireless networks (inexpensive small mobile devices vs. base stations) and in sensor networks (inexpensive tiny sensors vs. central control nodes/receivers). In fact, the potential and usefulness of joint source-channel soft-decision decoding was foreseen by Shannon in his celebrated 1948 paper: "... any redundancy in the source will usually help if it is utilized at the receiving point. In particular, if the source already has redundancy and no attempt is made to eliminate it in matching to the channel, this redundancy will help combat noise" [1].

A popular and effective joint source-channel decoding technique is the maximum *a posteriori* sequence estimation. Using both the memory of the Markov sequence and the prior knowledge of the channel, the MAP decoder infers the input sequence as the one of maximal *a posteriori* probability. In other words, if the input sequence to the channel is \mathbf{X} and the receiver receives \mathbf{Y} , the joint source-channel MAP decoder emits

$$\hat{\mathbf{X}} = \underset{\mathbf{x}}{\operatorname{argmax}} P(\mathbf{X}|\mathbf{Y}). \quad (2.1)$$

Early works on the estimation of a Markov Source over a noisy channel include [23] and [24]. In [25], the problem of MAP estimation of fixed-length coded Markov sequences was addressed. Although their simulation results showed a significant amount of error correction without forward error correction, the metric used to solve the problem was not strictly optimal. The work of Sayood and Borkenhagen was later extended by Phamdo and Farvardin, who showed that the MAP sequence detection can be posed as a shortest path problem through a *symbol-based* trellis, which can be solved exactly and efficiently by dynamic programming [26]. This dynamic

programming algorithm is very similar to the Viterbi algorithm used to decode the convolutional code. Actually, Forney, Jr. had pointed out that the dynamic programming algorithm used to solve the joint source-channel MAP decoding problem is a generalized version of the Viterbi algorithm [13].

When the source sequence is encoded by variable-length code (VLC), the problem of joint source-channel MAP decoding becomes much more complicated. This is due to the fact that the candidate sequences can have different phases and even different lengths if the source sequence is encoded by VLC, which is not an issue for FLC. As a result, in the case of VLC the joint source-channel MAP decoder has to consider all possible legal parsings of the received sequence. Nevertheless, the VLC version of MAP decoding can still be solved by dynamic programming technique on an irregular *bit-level* trellis. The structure of the bit-level trellis is given by the codeword lengths of the corresponding VLC codebook.

The existing algorithms for MAP decoding of VLC-coded Markov sequences fall into two types: without and with the prior knowledge about the number of symbols transmitted (the length constraint). Subbalakshmi and Vaisey proposed MAP decoding algorithms without length constraint for the binary symmetric channel (BSC) [27] and the additive Markov channel [28], respectively. These algorithms were extended by Wu *et al.* to correct substitution error and erasure error as well [29]. In [29], the bit-level trellis of [27] was simplified which led to a more efficient solution to the problem. It was also shown that the solution can be made even faster by matrix search strategy if the source is Gaussian Markov.

The number of transmitted symbols is a useful side information and can improve the error correction capability. However, making use of this information in VLC MAP decoding is not simple. In [30], a computationally complex exact MAP decoding method and an efficient approximation were both suggested. These methods save in every bit-level node of the trellis not only the probability information (weight) but also the number of symbols parsed so far. Another similar approximation method was proposed in [4]. These algorithms were compared in performance in [31]. In [32], the

MAP decoding algorithm without length constraint was used to solve the problem with length constraint by inserting a free parameter into the graph edge weights. This technique saved the expensive operations of parsing a VLC bit stream. In [33] a maximum *a posteriori* probability (MAP) based, joint source-channel decoder for variable length coded intra and inter macroblocks in a MPEG-4 encoded video stream is presented.

The joint source-channel MAP decoding problem was also investigated for other types of source coding and in the presence of channel coding. Joint source-channel coding of arithmetic codes using a special symbol was proposed by Pettijohn et al. in [34]. In [35] a MAP decoding technique for arithmetic code with a forbidden symbol was also proposed. The joint source-channel decoding using higher order Markov models is studied by Lahouti and Khandani [36]. The joint MAP decoding of source code and channel code, e.g., convolutional code and turbo code, was extensively studied [37, 4, 38, 39, 40] as well.

2.2 Binary Index Assignment

Index assignment refers to the labeling of source symbols by binary integer numbers. An example that the readers can immediately think of is the so-called natural binary code (NBC), which is the binary encoding of the decimal indices (starting from 0) of the quantizer codewords. Different index assignments do not change the distortion of the quantizer, but they do affect the overall distortion of a communication system in case of channel errors. Therefore, index assignment can be optimized with respect to channel statistics to mitigate the impact of channel errors. In this context index assignment can be considered as a joint source-channel coding technique. The distortion performance of an index assignment can be fully characterized by its *Hamming distance matrix*, which is an $N \times N$ matrix whose elements are the Hamming distances between corresponding codeword pairs. Two index assignments are called *equivalent* if their Hamming distance matrices are the same. Otherwise they are *distinct*. It is

0	1	1	2	1	2	2	3	1	2	2	3	2	3	3	4
1	0	2	1	2	1	3	2	2	1	3	2	3	2	4	3
1	2	0	1	2	3	1	2	2	3	1	2	3	4	2	3
2	1	1	0	3	2	2	1	3	2	2	1	4	3	3	2
1	2	2	3	0	1	1	2	2	3	3	4	1	2	2	3
2	1	3	2	1	0	2	1	3	2	4	3	2	1	3	2
2	3	1	2	1	2	0	1	3	4	2	3	2	3	1	2
3	2	2	1	2	1	1	0	4	3	3	2	3	2	2	1
1	2	2	3	2	3	3	4	0	1	1	2	1	2	2	3
2	1	3	2	3	2	4	3	1	0	2	1	2	1	3	2
2	3	1	2	3	4	2	3	1	2	0	1	2	3	1	2
3	2	2	1	4	3	3	2	2	1	1	0	3	2	2	1
2	3	3	4	1	2	2	3	1	2	2	3	0	1	1	2
3	2	4	3	2	1	3	2	2	1	3	2	1	0	2	1
3	4	2	3	2	3	1	2	2	3	1	2	1	2	0	1
4	3	3	2	3	2	2	1	3	2	2	1	2	1	1	0

(a) Hamming distance matrix

0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0
1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0
0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0
0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1
0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1
0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0

(b) adjacency matrix

Figure 2.1: The Hamming distance matrix and adjacency matrix of NBC ($N = 16$).

obvious that equivalent index assignments have the same distortion performance.

Now we count the number of distinct index assignments. If the source is encoded by fixed integer rate n , there are $2^n!$ possibilities to order $N = 2^n$ codewords. Two index assignments are equivalent if one can be obtained by flipping some bit positions of the other, or by permutation of some bit indices. Therefore the number of distinct index assignments is [2]

$$\frac{2^n!}{2^n \times n!} = \frac{(2^n - 1)!}{n!}. \quad (2.2)$$

In many applications, for instance, the transmission over BSC of very small crossover probability, it is a good approximation to consider isolated single bit errors in transmission. In this approximation, the Hamming distance matrix is simplified to a special 0-1 form: those elements corresponding to a pair of codewords of mutual Hamming distance 1 are 1; all other elements are 0. The resulting 0-1 matrix is called the *adjacency matrix* of an index assignment. The Hamming distance matrix and the adjacency matrix of NBC of 16 codewords are given in Figure 2.1(a) and Figure 2.1(b) respectively.

First let us review how index assignment should be optimized for symbol-by-

symbol hard-decision decoding. An important observation is that even a single error could cause a large distortion given an index assignment. For example, when the NBC of 4 bits is used, a single bit error could alter 0 (0000) into 8 (1000). Therefore, it is natural to index source codewords that are mutually far apart in code space by binary numbers of mutually large Hamming distance. An early example of index assignment is the Gray code (GC) [41], which labels two consecutive scalar source codewords by two binary numbers of Hamming distance 1. Another code is called folded binary code (FBC) or signed binary code [42], where the first and most significant bit indicates the sign of signal and the remaining bits represent the magnitude. Errors in the most significant bit will cause a distortion twice the magnitude. For many sources, such as the prediction error in image and video coding, the signal has very high probability to be close to zero, resulting in a small expected distortion. Note that the error in the first bit of NBC will always cause a distortion of half the amplitude range. The NBC, GC and FBC for $N = 16$ are listed in Table 2.1.

A great deal of efforts have been made in the communication community since the 1960's to determine the performances of various index assignments and to find the optimal code. In [43, 44], formulae for the MSE of uniform scalar quantizers and uniform sources for NBC, Gray code and any index assignment on a BSC are derived. The optimality of NBC is also asserted without a published proof. Later in [45], Crimmins et al. proved NBC is optimal for uniform scalar quantization for scalar source in the sense of MSE. A general approach of analyzing digital errors in linear and nonlinear PCM systems with arbitrary channel is proposed by Rydbeck and Sundberg in [46].

The effects of channel errors on vector quantizer performance were first discussed by Bingöl and Sankur in [47]. In [48], Zeger and Gersho proposed the pseudo-Gray coding that can effectively reduce the average distortion of a communication system using vector quantization. In [49], the problem of scalar and vector quantized signal transmitted via the BSC was also considered by McLaughlin et al.

The index assignment problem was also studied for other distortion measurements.

	NBC	GC	FBC	HarperB
0	0 0000	0 0000	14 1110	0 0000
1	1 0001	8 1000	12 1100	1 0001
2	2 0010	12 1100	10 1010	2 0010
3	3 0011	4 0100	8 1000	4 0100
4	4 0100	6 0110	6 0110	8 1000
5	5 0101	14 1110	4 0100	3 0011
6	6 0110	10 1010	2 0010	5 0101
7	7 0111	2 0010	0 0000	9 1001
8	8 1000	3 0011	1 0001	6 0110
9	9 1001	11 1011	3 0011	10 1010
10	10 1010	15 1111	5 0101	12 1100
11	11 1011	7 0111	7 0111	7 0111
12	12 1100	5 0101	9 1001	11 1011
13	13 1101	13 1101	11 1011	13 1101
14	14 1110	9 1001	13 1101	14 1110
15	15 1111	1 0001	15 1111	15 1111

Table 2.1: Various index assignments in both decimal (in bold) and binary form ($N = 16$).

In [50, 51], a class of index assignments that minimizes the mean magnitude error was presented for signals (drawn from an equally probable source) transmitted over a BSC of small but realistic crossover probability. It was shown that the NBC and some unit distance codes, including the GC are among this class of optimal index assignments. In [52], Harper found the optimal code that minimizes the maximum absolute distortion for uniform source quantized by uniform quantizer and contaminated by a single bit error. This optimal code, which is referred to as the HarperB code in this thesis (see Table 2.1 for the HarperB code of $N = 16$) in Chapter 3, can be easily constructed by a so-called Hales numbering.

Besides being optimized for the hard-decision decoder, index assignment can also be optimized for other types of decoders. For example, a channel optimized decoder

finds a weighted centroid of all possible source symbols as follows,

$$y(j) = \sum_{i=0}^{2^n-1} c(i)q(j|i),$$

where $c(i)$ is the centroid of a code cell i and $q(j|i)$ is the probability of sending i and receiving j . For quantizers with a uniform encoder and the channel optimized decoder, it is shown in [53] that NBC is the optimal index assignment.

For decoders that perform sequence estimation instead of symbol by symbol decoding, the index assignment design shares a similar spirit with the channel coding theory. In [50], Harper pointed out that, “A variation on this idea (*minimization of the average absolute error*) shows how to maximize the average. Such a result may be useful for detecting errors in information where consecutive readings are correlated, e.g., temperature readings from a space probe. A single error in a word would then cause a grievous error in the reading which could be eliminated by comparison with preceding and following readings [54]”. This idea was extended in the work [37], where a joint source-channel decoder was combined with a convolutional decoder. Sayood et al. noted that “it would be useful to make assignments which increase the “distance” between likely sequences” and “our strategy is therefore to try to maximize the Hamming distance between codewords that are likely to be mistaken for one another”. It was shown that a heuristic code designed according to this rule does give good performance, which is however, not optimal. In Chapter 3, we build upon this idea and develop a systematic way to find the optimal index assignment for joint source-channel MAP decoding.

In general, it is impossible to design optimal index assignment explicitly. Exhaustive search is out of question, even if the cost function is very fast to compute, because the factorial growth rate of the number of distinct index assignments in code size. To see this, let us review equation (2.2). For $n = 3$, there are totally 840 distinct index assignments. For $n = 4$, this number soars to 54486432000. Therefore, researchers resort to some heuristic algorithms to find good codes. In [55], simulated annealing

algorithm [56, 57, 58] was used to help VQ index assignment design. In [59], Knagenhjelm and Agrell used Hadamard transform to find good index assignments. In this technique, the set of all index assignments were divided into classes. Starting from an arbitrary assignment, the best index assignment within a class can be easily found by a sorting procedure.

The reader is also referred to [60, p. 2372] for a detailed review on algorithmic techniques for designing good index assignments for particular sources and channels.

2.3 Multiple Description Coding

Multiple description coding (MDC) is a technique to combat channel impairments by mapping a single source sequence into multiple descriptions. Any description can be independently used to reconstruct a coarse version of the source signal. Multiple descriptions received could refine each other to yield a reconstruction of higher fidelity. The quality of the reconstruction improves with more descriptions received. An MDC of K descriptions has $2^K - 1$ decoders, corresponding to all combinations of received descriptions.

Traditional communication system protected by forward error correction is only good for known and stable channel condition. However, in many applications such as the Internet, wireless communication and sensor networks, the channel condition is volatile. The channel code may fail completely when the channel condition deteriorates below a certain threshold, which is called the *cliff effect* or *threshold effect*. Systems protected by automatic repeat request (ARQ) work well only when the channel condition is good. Otherwise, the retransmission could also fail and cause congestion. Besides, it requires more resources, e.g., a feedback channel and a larger receiver buffer. Therefore, ARQ technique is not suitable for real time applications such as teleconferencing and video streaming.

For applications with poor channel conditions and stringent delay constraints, MDC is a good coding and transmission strategy. All data received will help to

reconstruct the original signal and the quality of the reconstructed signal will improve gradually as more descriptions are received. As such, there is no cliff effect. The MDC decoder works on a best-effort basis and makes full use of the received packets (descriptions) to recover the source sequence, requiring no retransmission of missing descriptions. This makes MDC very suitable for real time communication via lossy networks such as multimedia streaming over the Internet.

The simplest implementation of MDC is to divide a bit stream into two: one sequence consisting of only odd bits and the other of only even bits. When both descriptions are received, there is no loss of information. When only one description is received, the missing bits can be estimated using resampling techniques such as linear or cubic interpolation. Obviously this approach works only for sources with high correlation between adjacent samples. It is not suitable for memoryless sources because when only one description is received, the missing samples cannot be estimated from the received descriptions.

MDC can also be realized using scalar quantization, which can be used for sources with or without memory. This technique is called multiple description quantization. The concept of MDQ can be explained by a simple example. Two uniform quantizers, called *side quantizers*, have interlaced code cells and can refine each other by intersection operation. When both descriptions are available, they constitute a *central quantizer* with higher resolution. A simple example of this approach is given in Figure 2.2 for the quantization of a source that is uniformly distributed in the interval $[0, 1]$. The two side quantizers (Figure 2.2(b) and Figure 2.2(c)) have 8 quantization cells each. But the quantization boundaries are shifted by half the step size. The resulting central quantizer has 15 quantization cells (Figure 2.2(a)). In this design, each sample is coded into two quantization indices, each of 3 bits. Note that the MDQ introduces a level of redundancy to the central quantizer when there is no loss, i.e., the rate of the central quantizer is fewer than 4 ($\log_2 15$ actually) rather than 6 bits per sample.

In general, if each side description needs B bits per sample to encode, then the

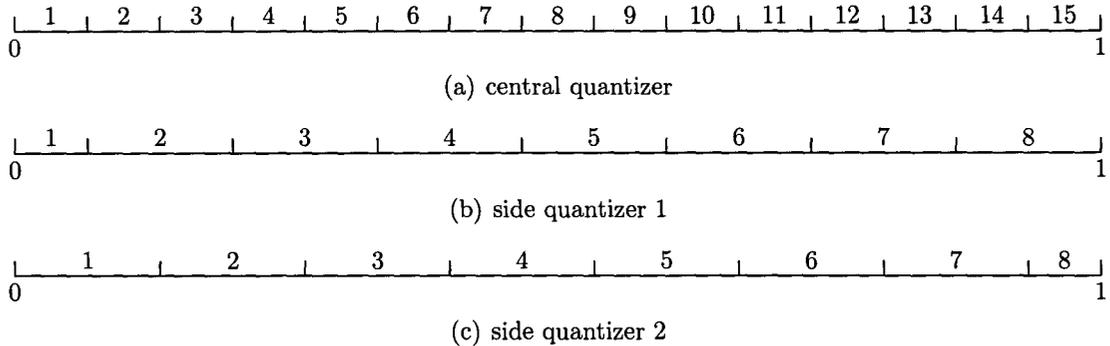


Figure 2.2: The central and side quantizers of the example MDQ.

above MDQ scheme will require a total rate of $2B$ bits per sample. And the central quantizer could only achieve a resolution of no more than $B + 1$ bits. This reveals the redundancy introduced between the side descriptions. The redundancy level, however, could be controlled. In [61], Vaishampayan proposed a systematic way to design multiple description scalar quantization schemes similar to the one given in Figure 2.2. A source sample is first quantized by a central quantizer. Each central codeword x is then mapped to a pair $(\lambda_1(x), \lambda_2(x))$, where λ_i , $i = 1, 2$ is the index assignment function, which can be represented by an index assignment¹ matrix. For instance, the index assignment matrix of the MDQ given in Figure 2.2 is shown in Figure 2.3(a). An index assignment matrix of an MDQ with less redundancy is given in Figure 2.3(b). This method can be easily extended to more than two descriptions.

A simple way to quantify the redundancy level of a two-description quantization is to count the number of diagonals occupied by the central quantizer indices. The more diagonals occupied, the less redundancy (or less correlation) between the descriptions. There is no redundancy if the index assignment matrix is full. In general, for MDQ of K descriptions, the redundancy rate carried by the MDQ versus the single description is $1 - \log_2 L / \sum_{k=1}^K \log_2 L_k$ [62], where L is the number of codecells of the central quantizer and $L_k \leq L$ is the number of codecells of side quantizer k with $L \leq \prod_{k=1}^K L_k$.

A side benefit of MDC is that it provides rate scalability inherently. The source

¹This index assignment means the mapping from a central quantizer codeword to side quantizer codewords and should be distinguished from the binary index assignment presented in Section 2.2.

1							
2	3						
	4	5					
		6	7				
			8	9			
				10	11		
					12	13	
						14	15

(a) $d = 2$

1	2						
3	4	6					
	5	7	8				
		9	10	11			
				12	14		
				13	15	16	
					17	18	20
						19	21

(b) $d = 3$

Figure 2.3: The index assignment for two two-description scalar quantizer proposed by Vaishampayan, where d denotes the number of diagonals occupied by the central quantizer indices.

generates and sends all descriptions to the channels without knowing the channel conditions nor the hardware resources of the decoders. Receivers with limited bandwidth and/or computational resources could subscribe to only a subset of descriptions, while full-equipped receivers decode all descriptions for the best performance.

For a more comprehensive coverage on MDC, please refer to Goyal's survey paper [63].

Chapter 3

Index Assignment for Joint Source-Channel MAP Decoding

3.1 Overview

In the introduction we emphasized that the JSC decoder should not be considered as an isolated component in lossy communication systems. Its behavior depends not only on the source and channel statistics, but also on the quantizer index assignment. Thus a natural question begs our attention: if and how the quantizer index assignment can be optimized such that the JSC MAP decoder can better exploit the residual source redundancy. If we regard the redundancy left in the fixed-rate suboptimal source code as a primitive type of channel code, then index assignment optimization plays a similar role as the design of a channel code. However, this connection between the index assignment and subsequent MAP decoding has not been discussed in the existing literature on joint source-channel coding, probably hindered by the highly intricate interplay between the two. Experiment results show that the performance gap of the best and worst index assignment could be significant. In particular, those index assignment schemes designed for hard-decision decoding, such as GC and NBC, turn out to perform poorly if they are used in tandem with a joint source-channel MAP decoder for Gaussian Markov sources. This appears to be a common defect of

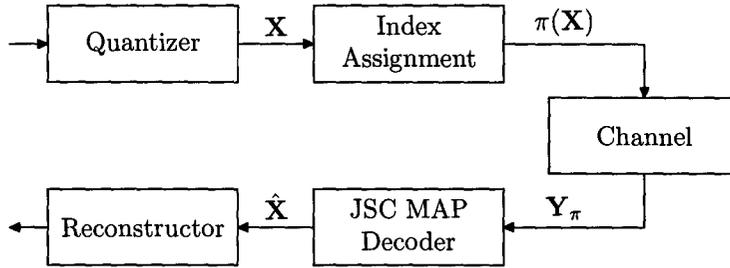


Figure 3.1: System diagram.

existing methods of joint source-channel MAP decoding [25, 26, 31, 29, 27, 28] that overlook index assignment of source codes altogether.

This chapter is a study, the first of this type to the best of our knowledge, on optimal index assignment for joint source-channel decoding. We cast the problem in a general framework of quadratic assignment problem (QAP) that encompasses various sources and different distortion metrics. Since QAP problem is NP-hard, general solutions are necessarily based on heuristic algorithms. Fortunately, for some special and practically important cases, we are able to explore some properties of the objective function to construct good index assignments analytically. Experimental results show that optimized index assignments with respect to joint source-channel MAP decoding outperform those designed for hard-decision decoders considerably.

This chapter has the following presentation flow. Section 3.2 formulates the problem of optimal index assignment for MAP decoding. In Section 3.3, the structures of optimal index assignment in relation to the amount of memory in Gaussian Markov sources are discussed. We then propose an optimal index assignment for highly correlated Gaussian Markov sources in Section 3.4. In Section 3.5, a simulated annealing algorithm is proposed to solve the index assignment problem for general sources. Simulation results are reported in Section 3.6 to corroborate our derivation and analysis. Section 3.7 concludes.

3.2 Problem Formulation

The system under study is schematically depicted in Figure 3.1. Suppose that a first-order stationary Markov sequence is first quantized by a scalar or vector quantizer, and then transmitted in fixed-length code via a binary symmetric channel. Let the quantized input sequence to the BSC channel, which is also approximately Markov, be $\mathbf{X} = x_0x_1 \cdots x_{L-1} \in \mathbb{C}^L$, where $\mathbb{C} = \{\alpha_1, \alpha_2, \cdots, \alpha_N\}$ is the quantizer codebook. An index assignment of \mathbb{C} is a mapping $\pi : \mathbb{C} \rightarrow \{0, 1\}^n$, $n = \lceil \log_2 N \rceil$. Let the output binary sequence of the BSC channel be \mathbf{Y}_π .

The receiver performs joint source-channel MAP decoding to find the most probable input sequence \mathbf{X} given the received sequence \mathbf{Y}_π , namely

$$\hat{\mathbf{X}}(\pi) = \underset{\mathbf{X}}{\operatorname{argmax}} P(\mathbf{X}|\mathbf{Y}_\pi). \quad (3.1)$$

Using Bayes' theorem,

$$\begin{aligned} P(\mathbf{X}|\mathbf{Y}_\pi) &= \frac{P(\mathbf{Y}_\pi|\mathbf{X}) \cdot P(\mathbf{X})}{P(\mathbf{Y}_\pi)} \\ &\propto P(\mathbf{Y}_\pi|\mathbf{X}) \cdot P(\mathbf{X}), \end{aligned} \quad (3.2)$$

we have

$$\begin{aligned} \hat{\mathbf{X}}(\pi) &= \underset{\mathbf{X}}{\operatorname{argmax}} P(\mathbf{Y}_\pi|\mathbf{X}) \cdot P(\mathbf{X}) \\ &= \underset{\mathbf{X}}{\operatorname{argmax}} [\log P(\mathbf{Y}_\pi|\mathbf{X}) + \log P(\mathbf{X})]. \end{aligned} \quad (3.3)$$

Write \mathbf{Y}_π as $\mathbf{Y} = y_0y_1 \cdots y_{L-1} \in \mathbb{C}^L$. Since the BSC channel is memoryless, the probability of receiving \mathbf{Y}_π when \mathbf{X} is transmitted can be expressed as

$$P(\mathbf{Y}_\pi|\mathbf{X}) = \prod_{l=0}^{L-1} P_\pi(y_l|x_l) \quad (3.4)$$

with

$$P_\pi(y_l|x_l) = (1 - p_c)^{n-d(\pi(x_l),\pi(y_l))} p_c^{d(\pi(x_l),\pi(y_l))}, \quad (3.5)$$

where p_c is the BSC crossover probability, and $d(\cdot, \cdot)$ is the Hamming distance. The a priori probability $P(\mathbf{X})$ can be expressed by the chain rule of a Markov process as

$$P(\mathbf{X}) = P(x_0) \prod_{l=1}^{L-1} P(x_l|x_{l-1}). \quad (3.6)$$

Thus, for a fixed π , the MAP decoding problem becomes

$$\hat{\mathbf{X}}(\pi) = \underset{\mathbf{X}}{\operatorname{argmax}} \left\{ \log P(x_0) + \log P_\pi(y_0|x_0) + \sum_{l=1}^{L-1} [\log P(x_l|x_{l-1}) + \log P_\pi(y_l|x_l)] \right\}, \quad (3.7)$$

which can be solved by dynamic programming in a weighted directed acyclic graph thanks to the additivity of the objective function [25, 26].

Since the MAP sequence $\hat{\mathbf{X}}(\pi)$ is a function of index assignment π , the problem of MAP-optimized index assignment arises. Ideally, one would like to use an optimal index assignment π^* such that joint source-channel MAP decoding minimizes a suitable distortion metric of the resulting output sequence. To appreciate the posed problem, let us use an example to show how index assignment affects the performance of MAP sequence estimation.

Example 1. Consider a first-order stationary Markov source with a state space of four symbols. The logarithm (base 2) of the stationary distribution is

$$[\log P(\alpha_i)]_{i=1,2,3,4} = [-2.65, -1.56, -1.56, -2.65]$$

\mathbf{X}	$\log P(\mathbf{X})$	$d(\mathbf{X}, \mathbf{Y})$	$\log P(\mathbf{Y} \mathbf{X})$	$\log P(\mathbf{X}) + \log P(\mathbf{Y} \mathbf{X})$
11 10 11 01	-20.48	0	-0.08	-20.56
11 00 11 01	-8.74	1	-6.71	-15.45
11 11 11 01	-5.01	1	-6.71	-11.72
11 10 01 01	-12.93	1	-6.71	-19.64
11 10 10 01	-12.80	1	-6.71	-19.51
...

Table 3.1: MAP decoding with index assignment π_1

and the logarithm of the transition matrix is

$$[\log P(\alpha_i|\alpha_j)]_{\substack{i=1,2,3,4 \\ j=1,2,3,4}} = \begin{bmatrix} -0.46 & -1.91 & -7.78 & -14.33 \\ -3.00 & -0.59 & -2.27 & -8.87 \\ -8.87 & -2.27 & -0.59 & -3.00 \\ -14.33 & -7.78 & -1.91 & -0.46 \end{bmatrix}.$$

The channel is a BSC with crossover probability $p_c = 0.01$. From (3.5) we have

$$\begin{aligned} \log P(y|x) &= [n - d(x, y)] \log(1 - p_c) + d(x, y) \log p_c \\ &= -0.01[n - d(x, y)] - 6.64d(x, y). \end{aligned} \quad (3.8)$$

Given a source sequence $\alpha_2\alpha_2\alpha_2\alpha_3$, we compare the performance of two different index assignments: $\pi_1 = (00, 11, 01, 10)$ and $\pi_2 = (00, 01, 10, 11)$.

If π_1 is used, the input to the BSC is:

$$\mathbf{X} = 11 11 11 01.$$

If during transmission, the fourth bit is flipped, the decoder receives:

$$\mathbf{Y} = 11 10 11 01.$$

\mathbf{X}	$\log P(\mathbf{X})$	$d(\mathbf{X}, \mathbf{Y})$	$\log P(\mathbf{Y} \mathbf{X})$	$\log P(\mathbf{X}) + \log P(\mathbf{Y} \mathbf{X})$
01 00 01 10	-8.74	0	-0.08	-8.82
00 00 01 10	-7.29	1	-6.71	-14.00
01 10 01 10	-8.37	1	-6.71	-15.08
01 01 01 10	-5.01	1	-6.71	-11.72
01 00 00 10	-12.80	1	-6.71	-19.51
...

Table 3.2: MAP decoding with index assignment π_2

The decoder searches through all possible input sequences, as listed in Table 3.1, and finds the one of maximum *a posteriori* probability $\hat{X}(\pi_1) = \alpha_2\alpha_2\alpha_2\alpha_3$, correcting the bit error.

Encoding the same symbol sequence with π_2 , we have

$$\mathbf{X} = 01\ 01\ 01\ 10.$$

Flipping also the fourth bit, we receive:

$$\mathbf{Y} = 01\ 00\ 01\ 10.$$

The same MAP decoding strategy (see Table 3.2) produces output sequence $\hat{X}(\pi_2) = \alpha_2\alpha_1\alpha_2\alpha_3$, resulting in a symbol error.

In the above example, π_1 collaborates with MAP decoding better than π_2 in correcting bit errors, because the former index assignment makes sequences of few bit errors to have very small probabilities. This is reminiscent of the design principle of the convolutional code to maximize the minimum free distance [64]. In fact, the dynamic programming algorithm used to solve the joint source-channel MAP decoding problem is nothing but a generalized version of the Viterbi algorithm [13]. By analogy, a good index assignment for joint source-channel MAP decoding should increase the “distance” between two likely sequences. However, unlike convolutional code whose

performance can be estimated by the minimum free distance, it is very difficult to analytically quantify the performance of MAP decoding. In order to have a bearing on the problem of optimizing the index assignment for MAP decoding, we need to make some necessary simplifications.

For relatively good BSC channels of small crossover probability p_c , the event of two consecutive bit errors is very rare. Therefore, it suffices to study isolated single-bit errors. Suppose that $\pi(\alpha_i)\pi(\alpha_j)$ are two consecutive codeword indices and a single bit error occurs in $\pi(\alpha_j)$. The error causes the index $\pi(\alpha_j)$ to be received as $\pi(\alpha_k)$ for some k such that $d(\pi(\alpha_j), \pi(\alpha_k)) = 1$, resulting in a symbol error. From (3.7), if $P(\alpha_k|\alpha_i)$ is less than $P(\alpha_j|\alpha_i)$, the MAP decoder is very likely to correct this symbol error. In other words, a good index assignment can collaborate with the source redundancy to guide the decision process of MAP sequence estimation to overrule the channel observation α_k by the correct symbol α_j , given that the preceding symbol α_i is decoded correctly.

In case that the MAP decoding fails to correct all bit errors, we introduce a distortion function $D : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{R}^+$ to quantify the penalty of decoding a transmitted symbol α_j into an erroneous symbol α_k . If our goal is to minimize symbol error rate, we choose $D(j, k)$ to be the Hamming distortion, which is defined as

$$D(j, k) = \begin{cases} 0 & \text{if } j = k, \\ 1 & \text{if } j \neq k. \end{cases} \quad (3.9)$$

If L_1 or L_2 error metric is desired, we can use the absolute error distortion $D(j, k) = \|\alpha_j - \alpha_k\|_1$ or the squared error distortion $D(j, k) = \|\alpha_j - \alpha_k\|_2^2$

Therefore, we pose the problem of optimal index assignment for joint source-channel MAP decoding as

$$\min_{\pi} \sum_{i=1}^N \sum_{j=1}^N \left[P_i P_{ij} \sum_{k:d(\pi(\alpha_j), \pi(\alpha_k))=1} H(P_{ik} - P_{ij}) D(j, k) \right], \quad (3.10)$$

where and also in the sequel $P_{ij} = P(\alpha_j|\alpha_i)$ and $P_i = P(\alpha_i)$. Here $H(\cdot)$ is the Heaviside unit step function:

$$H(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases} \quad (3.11)$$

For each index assignment π , define an $N \times N$ adjacency matrix $A^{(\pi)} = [A_{jk}^{(\pi)}]$:

$$A_{jk}^{(\pi)} = \begin{cases} 1, & \text{if } d(\pi(\alpha_j), \pi(\alpha_k)) = 1; \\ 0, & \text{otherwise.} \end{cases} \quad (3.12)$$

If $N = 2^n$, then A represents the edges of the n -dimensional hypercube. Clearly, A is a 0-1 symmetric matrix with all the elements on the main diagonal being zero and the number of 1's in each row and each column being n , or

$$\sum_{j=1}^N A_{jk}^{(\pi)} = n, \quad \text{and} \quad \sum_{k=1}^N A_{jk}^{(\pi)} = n. \quad (3.13)$$

Now the objective function in (3.10) can be written as

$$\begin{aligned} C(\pi) &= \sum_{i=1}^N \sum_{j=1}^N \left[P_i P_{ij} \sum_{k=1}^N H(P_{ik} - P_{ij}) D(j, k) A_{jk}^{(\pi)} \right] \\ &= \sum_{j=1}^N \sum_{k=1}^N \left[D(j, k) \sum_{i=1}^N P_i P_{ij} H(P_{ik} - P_{ij}) \right] A_{jk}^{(\pi)} \\ &= \sum_{j=1}^N \sum_{k=1}^N B_{jk} A_{jk}^{(\pi)}, \end{aligned} \quad (3.14)$$

where

$$B_{jk} = D(j, k) \sum_{i=1}^N P_i P_{ij} H(P_{ik} - P_{ij}). \quad (3.15)$$

Note that the adjacency matrix $A^{(\pi)}$ is symmetric, whereas matrix B is generally not. Hence we replace B by $\bar{B} = (B + B^T)/2$. Now optimal quantizer index assignment

with respect to MAP decoding can be stated as

$$\pi^* = \operatorname{argmin}_{\pi} \sum_{j=1}^N \sum_{k=1}^N \bar{B}_{kj} A_{jk}^{(\pi)}, \quad (3.16)$$

which is a quadratic assignment problem [65, 66]. In QAP the matrix \bar{B} is called the *flow matrix*.

3.3 Index Assignment Structures and the Source Memory

Algorithmically, optimal quantizer index assignment for MAP decoding poses a challenge because the underlying QAP problem is generally NP-hard. In hope for an efficient solution, one has to discover and exploit useful structures of the flow matrix \bar{B} . Next we study some special but practically important cases.

3.3.1 Flow Matrix for Memoryless Sources

When the source is memoryless, we have $P_{ij} = P_j$ for $1 \leq i, j \leq N$. The flow matrix B in QAP can be simplified as

$$\begin{aligned} B_{jk} &= D(j, k) \sum_{i=1}^N P_i P_j H(P_k - P_j) \\ &= D(j, k) P_j H(P_k - P_j) \sum_{i=1}^N P_i \\ &= D(j, k) P_j H(P_k - P_j). \end{aligned} \quad (3.17)$$

If the source is further uniformly distributed, we have $P_i = P_j$ for $1 \leq i, j \leq N$ and $B_{jk} = D(j, k) P_j \propto D(j, k)$. Then (3.14) reduces to

$$\pi^* = \operatorname{argmin}_{\pi} \sum_{j=1}^N \sum_{k=1}^N D(j, k) A_{jk}^{(\pi)}. \quad (3.18)$$

Note that under these conditions MAP decoding reduces to hard decision decoding due to lack of source memory. For uniform scalar quantization of a uniform source, the absolute error distortion function is reduced to $D(j, k) = |j - k|\Delta$, where Δ is the quantization step size. This special case was studied by Harper [50], who shows that the NBC is one of the optimal index assignments. For the mean-squared distortion $D(j, k) = (j - k)^2\Delta^2$, the NBC is also optimal for uniform scalar quantization [43, 44, 45, 49].

3.3.2 Flow Matrix for Gaussian Markov Sources

Consider a Gaussian Markov process

$$x_{t+1} = \rho x_t + w_t, \quad (3.19)$$

where $\rho < 1$ is the *correlation coefficient* and w_t is the *process noise*, assumed to be white Gaussian, with zero-mean and variance σ_0^2 . Then we have

$$p(x_t) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-x_t^2/2\sigma_1^2}, \quad (3.20)$$

$$p(x_{t+1}|x_t) = \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-(x_{t+1}-\rho x_t)^2/2\sigma_0^2}, \quad (3.21)$$

where $\sigma_1^2 = \sigma_0^2/(1 - \rho^2)$. If the source is uniformly quantized into N codecells with step size Δ , the prior probability and the transition probability can be expressed as

$$P_i = \int_{(i-1-N/2)\Delta}^{(i-N/2)\Delta} p(x) dx \approx p\left(\left(i - \frac{N+1}{2}\right)\Delta\right) \cdot \Delta, \quad (3.22)$$

$$\begin{aligned} P_{ij} &= \frac{\int_{(i-1-N/2)\Delta}^{(i-N/2)\Delta} \int_{(j-1-N/2)\Delta}^{(j-N/2)\Delta} p(y|x)p(x) dy dx}{\int_{(i-1-N/2)\Delta}^{(i-N/2)\Delta} p(x) dx} \\ &\approx p\left(\left(j - \frac{N+1}{2}\right)\Delta \middle| \left(i - \frac{N+1}{2}\right)\Delta\right) \cdot \Delta. \end{aligned} \quad (3.23)$$

Let $i' = i - \frac{N+1}{2}$, $j' = j - \frac{N+1}{2}$ and $k' = k - \frac{N+1}{2}$. Then it is straightforward from (3.21) and (3.23) that $P_{ik} > P_{ij}$ if and only if $k > j$ and $i' > (j' + k')/2\rho$, or when $k < j$ and $i' < (j' + k')/2\rho$.

We rewrite the flow matrix B_{jk} in (3.17) as

$$B_{jk} = D(j, k)G_{jk}, \quad (3.24)$$

where $G_{jk} = \sum_{i=1}^N P_i P_{ij} H(P_{ik} - P_{ij})$. Then for Gaussian Markov sources, we have

$$G_{jk} = \begin{cases} \sum_{i' \leq \frac{j'+k'}{2\rho}} P_i P_{ij}, & j > k; \\ \sum_{i' \geq \frac{j'+k'}{2\rho}} P_i P_{ij}, & j \leq k. \end{cases} \quad (3.25)$$

Since P_i and $P_{i,j}$ are both nonnegative, given any j , the value of G_{jk} increases with k until $k = j$, where G_{jk} attains its maxima, and then decreases with k when $k > j$, forming a unimodal discrete function with the ridge along the main diagonal.

Now if we adopt Hamming distortion, $D(j, k)$ is a matrix of zeros on the main diagonal and ones elsewhere. Because the main diagonal of the adjacency matrix of any index assignment are always filled with zeros, we can choose arbitrary values for the diagonal elements of $D(j, k)$ without affecting the solution to optimization problem. For example, we can set them to be ones, then $D(j, k)$ is an all one matrix and $B_{jk} = G_{jk}$. We also have $B_{jk} = |j - k|\Delta G_{jk}$ and $B_{jk} = (j - k)^2 \Delta^2 G_{jk}$ if absolute error and mean-squared error are chosen respectively. It is straightforward to derive the properties of \bar{B} from the expressions of B for these cases.

3.3.3 Flow Matrix and Source Memory

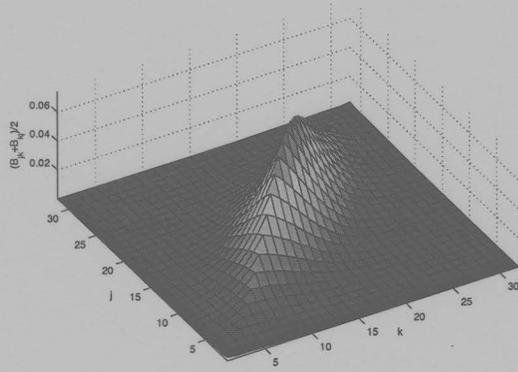
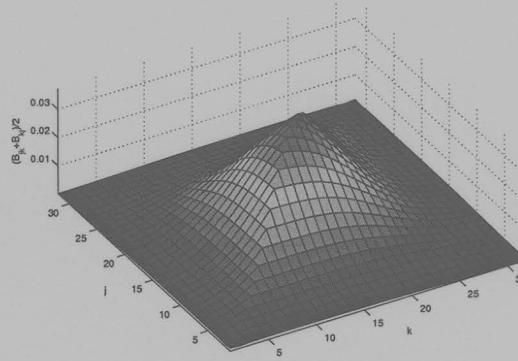
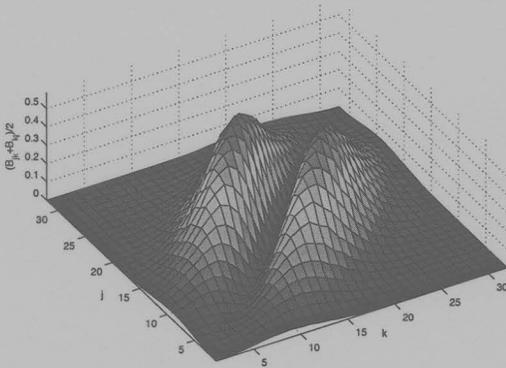
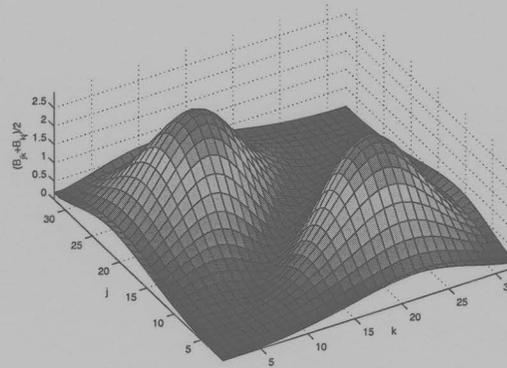
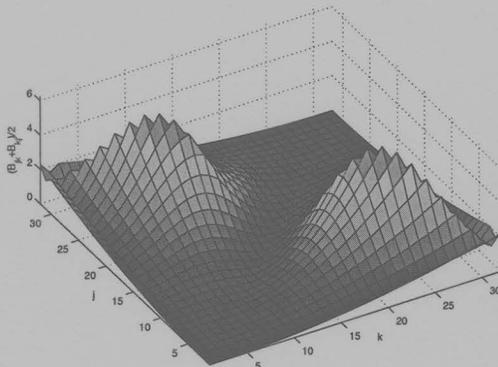
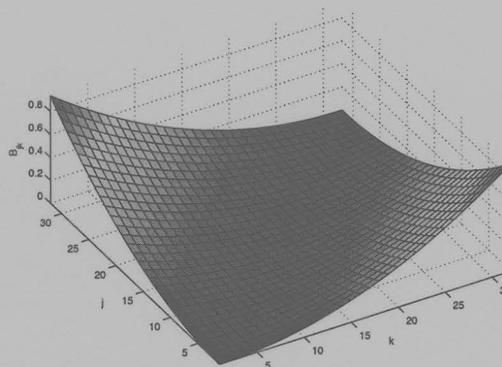
It is interesting and helpful to see how the structures of the matrix \bar{B} change as we change the source from memoryless to highly correlated, and from uniform to Gaussian. We consider \bar{B} in (3.16) for uniform scalar quantization. Figure 3.2 plots the values of the elements of matrix \bar{B} for six different cases: 1) Gaussian Markov source with correlation coefficient $\rho = 0.9$, and the metric for evaluating the distortion for

MAP decoding is Hamming distortion; 2) Gaussian Markov source with correlation coefficient $\rho = 0.5$, and Hamming distortion; 3) Gaussian Markov source with correlation coefficient $\rho = 0.9$, with mean squared error (MSE) being the distortion measure; 4) Gaussian Markov source with correlation coefficient $\rho = 0.5$, MSE; 5) memoryless Gaussian source ($\rho = 0$), MSE, and 6) uniformly distributed memoryless source, MSE. Note that the flow matrices are calculated without any approximation as introduced in (3.22) and (3.23).

The element \bar{B}_{jk} is the expected distortion $D(j, k)$ weighted by the approximate likelihood that MAP sequence estimation fails to correct the channel error from α_j to α_k . Therefore, to minimize the objective function (3.16) when designing quantizer index assignment, the 0-1 hypercube adjacency matrix $A^{(\pi)}$ should be permuted to move 1's away the peak positions of \bar{B} . In other words, good quantizer index assignment for MAP decoding should assign index pairs of Hamming distance 1 to pairs of codewords that correspond to small values in \bar{B} .

For Gaussian Markov sources, when the correlation ρ is high, matrix \bar{B} is a unimodal function with a ridge along the main diagonal if error of MAP decoding is measured by Hamming distortion (Figure 3.2(a)). This is because when j and k are close, the transition probabilities P_{ij} and P_{ik} are high, MAP decoding does not have much discriminating power if the symbol α_j is received as α_k due to a bit error. The situation is different when mean squared error is the distortion measure (Figure 3.2(c)). The values of \bar{B} are small in a narrow band along the main diagonal because the distortion of mistakenly decoding α_k to α_j is quadratic in $|k - j|$. Interestingly, on the other hand, the values of \bar{B} decrease rapidly as moving away from the main diagonal, creating two parallel ridges near the main diagonal. This is because the Markov transition probability P_{kj} decreases rapidly in $|k - j|$ for large ρ , so that MAP sequence estimation has a high chance to correct a single bit error that mix up α_k and α_j .

The role of joint source-channel MAP decoding can be intuitively understood by observing the behavior of \bar{B} as ρ decreases from 0.9 to 0 in Figure 3.2. When $\rho = 0$,

(a) $\rho = 0.9$, Hamming distortion(b) $\rho = 0.5$, Hamming distortion(c) $\rho = 0.9$, MSE(d) $\rho = 0.5$, MSE(e) $\rho = 0$, MSE

(f) Memoryless uniform source, MSE

Figure 3.2: The flow matrix of the optimal index assignment problem ($N = 32$) for five Gaussian Markov Sources: (a) $\rho = 0.9$, Hamming distortion. (b) $\rho = 0.5$, Hamming distortion. (c) $\rho = 0.9$, MSE. (d) $\rho = 0.5$, MSE. (e) $\rho = 0$, MSE and (f) a memoryless uniform source, MSE.

```

0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0
0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0
0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0
0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0
0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0
1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0
0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1
0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0
0 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1
0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1
1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0

```

(a) GC

(b) NBC

```

0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0

```

(c) FBC

(d) HarperA

```

0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

(e) HarperB

(f) SA

Figure 3.3: The adjacency matrix of various index assignments ($N = 16$): (a) Gray code, (b) Natural binary code, (c) Folded binary code, (d) HarperA code, (e) HarperB code, and (f) code found by simulated annealing algorithm for Gaussian Markov source ($\rho = 0.9$) and MSE, with decimal form (12 9 4 3 2 10 14 15 13 5 1 0 8 6 11 7).

the source becomes memoryless. MAP decoding degenerates to hard-decision decoding, and the MSE error of decoding α_j to α_k is quadratic in $|k - j|$. This relation can be clearly seen in Figure 3.2(f) when the source is memoryless and uniform. For memoryless Gaussian sources, the shape of \bar{B} results from weighting that of memoryless uniform source by Gaussian distribution, as one can visualize by comparing Figure 3.2(e) and Figure 3.2(f). Cases in between $\rho = 0$ and $\rho = 0.9$ are shown in Figure 3.2(b) and Figure 3.2(d).

Figure 3.2 alerts us to the fact that optimal quantizer index assignment for MAP decoding can change drastically from highly correlated sources to memoryless sources. This is reflected by the sensitivity of the flow matrix \bar{B} to the amount of source memory, which is clearly illustrated by graphs of \bar{B} in different cases of Figure 3.2. For instance, the structure of the flow matrix \bar{B} for memoryless source (see Figure 3.2(f)) can easily explain why the GC, the NBC and the FBC are good index assignments for hard decision decoding of uniform memoryless sources. The adjacency matrices A for these codes when $N = 16$ are plotted in Figure 3.3(a), Figure 3.3(b) and Figure 3.3(c) respectively. We can see that these codes have a large number of 1's placed along the main diagonals of their adjacency matrices. Therefore, they assign index pairs of Hamming distance 1 to pairs of source codewords that have small values in \bar{B} .

In a sharp contrast, if the source is Gaussian Markov and has a significant amount of memory, good index assignments should place 1's of the corresponding adjacency matrix away from the main diagonal, as dictated by the flow matrices in Figure 3.2(a) and Figure 3.2(c). Such index assignments will be discussed in the next section.

3.4 Index Assignment for Gaussian Markov Sources

For highly correlated Gaussian Markov sources, as exhibited by Figure 3.2(a) and Figure 3.2(c), \bar{B} has large values situated on and near the main diagonals. Therefore, an optimal index assignment corresponds to a permutation that pushes 1's of the adjacency matrix as far away from the main diagonal as possible. This observation

allows us to formulate optimal index assignment for MAP decoding as the following combinatoric optimization problem:

$$\begin{aligned} & \underset{\pi}{\text{maximize}} && \delta \\ & \text{subject to} && A_{ij}^{(\pi)} = 0, |i - j| \leq \delta, 1 \leq i, j \leq N. \end{aligned} \quad (3.26)$$

In [67] we proposed an approximate solution to the above problem that achieves $\delta = N/4 = 2^{n-2}$. Recently we found that in the literature of graph theory, the solution to the problem (3.26) is called the separation number [68] or anti-bandwidth [69, 70] of the hypercube. Its dual problem, the bandwidth problem [71, 72] of the hypercube, is to minimize the maximum distance of the 1's to the main diagonal among all permutations of matrix A , which is formulated as follows

$$\begin{aligned} & \underset{\pi}{\text{minimize}} && \delta \\ & \text{subject to} && A_{ij}^{(\pi)} = 0, |i - j| \geq \delta, 1 \leq i, j \leq N. \end{aligned} \quad (3.27)$$

Clearly, the study of the bandwidth problem can help find the worst index assignment for highly correlated Gaussian Markov sources.

These two problems have been studied four decades ago and solved partly by Harper [52, 73]. Harper's algorithms are very simple. The bandwidth of the hypercube can be achieved by numbering the codewords in the Hales order that is defined as follows.

Definition 1. The *Hales order*, \leq_H , on the set of all binary codewords $\{0, 1\}^n$, is defined by $u \leq_H v$ if

1. $w(u) < w(v)$, or
2. $w(u) = w(v)$ and u is greater than v in lexicographic order relative to the right-to-left order of the coordinates,

where $w(\cdot)$ is the Hamming weight of a binary codeword.

The solution of the anti-bandwidth problem, i.e., the optimal index assignment

in our formulation, can be constructed by first numbering the codewords with even Hamming weights and then numbering the codewords with odd Hamming weights, in the Hales order. We refer to this index assignment as HarperA code and name the bandwidth achieving permutation as HarperB code in the rest of this chapter.

Figure 3.3(d) shows the adjacency matrix of HarperA code for $N = 16$. The reader should compare the adjacency matrices of HarperA code and Gray code, i.e., Figure 3.3(d) vs. Figure 3.3(a), to appreciate how optimal index assignment varies according to the source memory.

Furthermore, Harper pointed out that the solution of the bandwidth of the n -dimensional hypercube is

$$\delta = \sum_{m=0}^{n-1} \binom{m}{\lfloor \frac{m}{2} \rfloor}. \quad (3.28)$$

However, no proof of this formula has been published in the literature. In his recent book [73], Harper noted that the proof of the bandwidth formula of hypercube “is surprisingly difficult”.

In [21], we observed a recursive structure of the Hales-numbered hypercube, based on which a very simple proof of 3.28 is presented. The recursive structure also leads us to the solution of the anti-bandwidth problem of the hypercube and its proof, which is a new result in the literature.

Theorem 1. *The solution of the anti-bandwidth problem (3.26) is*

$$\delta = 2^{n-1} - \sum_{m=0}^{n-2} \binom{m}{\lfloor \frac{m}{2} \rfloor}. \quad (3.29)$$

This value enables us to quickly determine if a particular index assignment is good or not, simply by comparing its anti-bandwidth value with δ .

The recursive structure of Hales numbering also help us understand the structures of the adjacency matrices of HarperA code and HarperB code. For more details about the hypercube bandwidth and antibandwidth and their proofs, please refer to Appendix A.

Algorithm 1: Using simulated annealing to find a good index assignment.

Data: Initial index assignment π_0 , initial temperature T

Result: Index assignment π

```

1  $\pi \leftarrow \pi_0$ 
2 repeat
3   repeat
4     Get  $\pi'$ , a neighboring index assignment of  $\pi$ 
5      $\Delta C \leftarrow C(\pi') - C(\pi)$ 
6     if  $\Delta C < 0$  then  $\pi \leftarrow \pi'$ 
7     else  $\pi \leftarrow \pi'$  with probability  $e^{-\Delta C/T}$ 
8   until  $\pi$  stabilized
9   Reduce temperature  $T$ 
10 until cost reduction too small or too many iterations

```

3.5 Simulated Annealing Algorithm

The optimization problem of (3.16) is a quadratic assignment problem, which is NP-hard. When the number of codewords N gets large, the computational complexity will be prohibitively high. Therefore, it is necessary to resort to some approximate optimization techniques, such as simulated annealing [56, 57, 58]. The algorithm is given in Algorithm 1, which is very similar to that used in [58]. The neighboring index assignment of π is obtained by exchanging two codewords in π (or substitute a codeword by an unused codeword when $N < 2^n$). The initial index assignment can be chosen manually such that the adjacency matrix approximately matches the flow matrix, or by selecting the code which gives the lowest cost among a set of commonly used codes.

3.6 Simulation Results

Various index assignments are tested on a scalar-quantized zero-mean, unit-variance, first-order Gaussian-Markov process of correlation coefficient $\rho = 0.9$. Uniform scalar quantizer was used for $N = 8, 16, 32, 64, 128, 256$ codecells, where the quantization step size is optimized such that the mean-squared quantization error is minimized.

		Code in Decimal
SER	Best	0 3 5 6 1 2 4 7
	Worst	0 1 2 3 7 6 5 4
MSE	Best	0 1 3 2 6 4 5 7
	Worst	0 7 1 2 5 3 4 6
	SA	3 1 5 7 6 2 0 4

Table 3.3: The best and worst codes found by exhaustive simulation for SER and MSE ($N = 8, p_c = 0.001$), and the code found by the simulated annealing algorithm for MSE ($N = 8$)

The simulated channel was BSC with crossover probability $p_c = 10^{-1}, 10^{-2}, 10^{-3}$. The symbol error rate (SER) and MSE are chosen as the performance metrics. The cost $C(\pi)$ of each index assignment, defined in (3.14), is also listed for comparison.

The index assignments tested include the GC, the NBC, the FBC, the HarperA code and the HarperB code. For $N = 8$, the best and worst index assignments are found by exhaustive simulation over all 840 distinct index assignments over a BSC channel with $p_c = 0.001$. The best and worst index assignments are evaluated only for $N = 8$ because when $N = 4$ there are only three distinct index assignments, and when $N = 16$, this number races up to about 5.45×10^{10} and we cannot afford the simulation! The simulated annealing algorithm in Algorithm 1 is also implemented to find the best index assignment for MSE distortion, which is denoted as the SA code in the sequel. For SER distortion, the codes found by the simulated annealing algorithm are either exactly the same as or very close to (in performance) to the HarperB code and will not be listed. Table 3.3 lists the best and worst codes for SER and MSE distortions, as well as the SA code for MSE, all for $N = 8$. The SA codes for $N > 8$ are listed in Table 3.5 .

Table 3.4 presents the symbol error rate of MAP decoding with different index assignments. One can see that the HarperA code outperforms other codes tested in almost all cases. The relative gain is larger for smaller p_c because the assumption that errors can be isolated is well satisfied. We also observe strong correlation between

the performances and the cost $C(\pi)$ of the index assignments. For $N = 8$, the best code found among all possible codes is actually the HarperA code and the worst code is the FBC code. Gray code is also very close to the worst code in both the cost function and the simulation performance.

Table 3.6 gives the mean-squared error of the sample index assignments. For smaller p_c , the performances of the SA codes are the best among all codes tested, and the correlation between the value of the cost function $C(\pi)$ and the MSE remains very high, whereas for larger p_c , this good behavior is greatly compromised because the assumption that errors can be isolated is no longer well satisfied. The adjacency matrix of the SA code for $N = 16$ is given in Figure 3.3(f). It is very interesting to see how the code tries to evade the two parallel peaks along the main diagonal of the flow matrix \bar{B} , as shown in Figure 3.2(c). Besides the SA code, HarperA code remains a very good code for larger N , where δ in (3.26) becomes so large that the 1's in the adjacency matrix A of the HarperA code (Figure 3.3(d)) can avoid the two parallel peaks effectively. Although the GC, NBC and FBC have much higher MSE than the SA code and the HarperA code for larger N and smaller p_c , they are not the worst. One could easily imagine that the worst index assignment should have 1's in the adjacency matrix distributed on both sides of main diagonal, along the two peaks of the flow matrix, which is exactly what we saw on the adjacency matrix of the HarperB code, given in Figure 3.3(e). This property is a straightforward result following the recursive structure of the Hales numbering as shown in Section A.5. The simulation results justify our conjecture: the HarperB code has much higher MSE than all other codes tested for larger N and smaller p_c . For $N = 8$, the best index assignment found by exhaustive search is identical to the SA code, validating the efficacy of the simulated annealing algorithm proposed.

The index assignment optimization algorithm is also tested on a non-Gaussian Markov source quantized into eight codewords, with the prior probability vector (0.18,

0.19, 0.17, 0.17, 0.10, 0.04, 0.08, 0.07) and the Markov transition probability matrix

$$\begin{bmatrix} 0.00 & 0.15 & 0.00 & 0.25 & 0.41 & 0.20 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.93 & 0.07 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.99 & 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 0.00 & 0.00 \\ 0.01 & 0.64 & 0.00 & 0.00 & 0.06 & 0.00 & 0.29 & 0.00 \\ 0.00 & 0.04 & 0.00 & 0.00 & 0.00 & 0.00 & 0.30 & 0.66 \\ 0.03 & 0.29 & 0.00 & 0.01 & 0.43 & 0.00 & 0.11 & 0.13 \\ 0.01 & 0.01 & 0.01 & 0.96 & 0.00 & 0.00 & 0.00 & 0.02 \\ 0.04 & 0.55 & 0.00 & 0.40 & 0.01 & 0.00 & 0.00 & 0.00 \end{bmatrix}.$$

The NBC is chosen as the initial index assignment for the simulated annealing algorithm. The evolution of the cost function during the process of the simulated annealing is shown in Figure 3.4. The best code found is (0, 2, 6, 1, 4, 7, 5, 3), whose performance, measured by SER, is compared against those of other sample codes, as shown in Table 3.7. We can see that the code found by the simulated annealing algorithm outperforms all other codes tested.

3.7 Conclusion

In this chapter we investigated the problem of optimal index assignment of the source codewords (fixed-length code) with respect to joint source-channel MAP decoding given the source statistic and certain distortion measure. The problem is formulated as a quadratic assignment problem. It is shown that the HarperA code is optimal or near optimal with respect to joint source-channel MAP decoding for Gaussian Markov sources of high correlation and large N and small p_c for both SER and MSE distortion metrics.

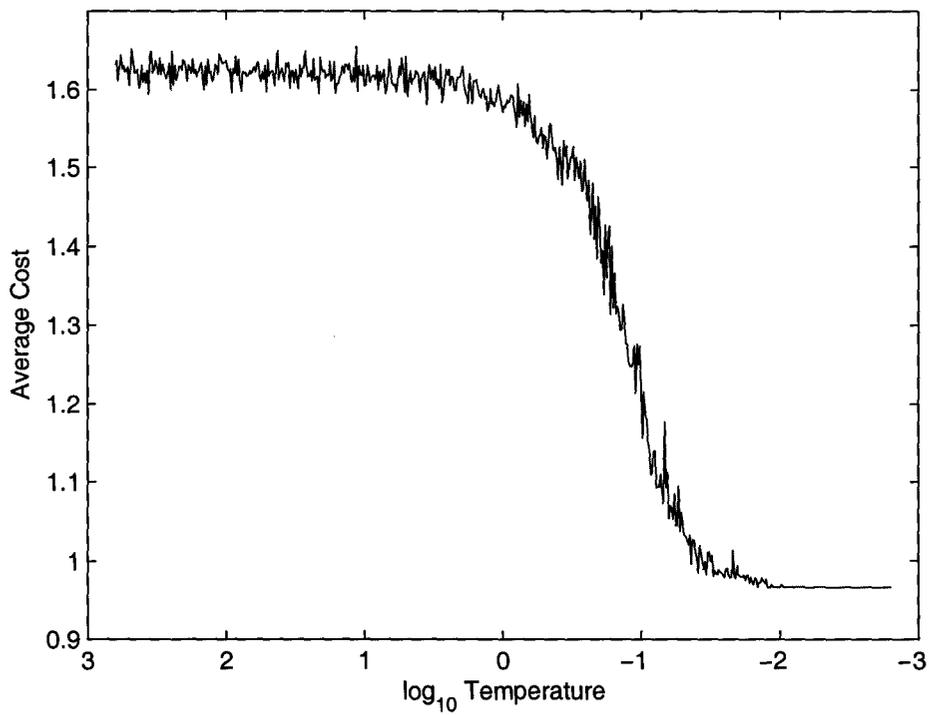


Figure 3.4: Evolution of $C(\pi)$ in the process of simulated annealing algorithm for a non-Gaussian Markov source ($N = 8$).

N	Code	$C(\pi)$	SER		
			$p_c = 10^{-1}$	$p_c = 10^{-2}$	$p_c = 10^{-3}$
8	Best	0.1682	0.1621	0.0150	0.0015
	Worst	0.5241	0.1999	0.0227	0.0026
	GC	0.5551	0.1996	0.0209	0.0022
	NBC	0.3974	0.1749	0.0184	0.0020
	FBC	0.5241	0.1999	0.0227	0.0026
	HarperA	0.1682	0.1621	0.0150	0.0015
	HarperB	0.2664	0.1705	0.0177	0.0020
16	GC	0.8603	0.2456	0.0276	0.0029
	NBC	0.6510	0.2255	0.0238	0.0026
	FBC	0.7917	0.2528	0.0285	0.0033
	HarperA	0.0917	0.2025	0.0169	0.0020
	HarperB	0.2425	0.2237	0.0214	0.0023
32	GC	1.2077	0.3076	0.0352	0.0038
	NBC	0.9846	0.2858	0.0309	0.0032
	FBC	1.1276	0.3140	0.0362	0.0038
	HarperA	0.0450	0.2608	0.0225	0.0024
	HarperB	0.2308	0.2897	0.0282	0.0030
64	GC	1.5950	0.3660	0.0431	0.0046
	NBC	1.3681	0.3449	0.0391	0.0040
	FBC	1.5116	0.3733	0.0443	0.0047
	HarperA	0.0211	0.3243	0.0309	0.0031
	HarperB	0.2149	0.3566	0.0358	0.0039
128	GC	2.0106	0.4224	0.0526	0.0055
	NBC	1.7824	0.4021	0.0473	0.0051
	FBC	1.9259	0.4267	0.0519	0.0056
	HarperA	0.0090	0.3926	0.0406	0.0041
	HarperB	0.2090	0.4219	0.0452	0.0049
256	GC	2.5464	0.4861	0.0617	0.0064
	NBC	2.3177	0.4688	0.0581	0.0060
	FBC	2.4612	0.4901	0.0625	0.0065
	HarperA	0.0138	0.4738	0.0522	0.0053
	HarperB	0.3517	0.5109	0.0592	0.0064

Table 3.4: SER of different index assignments for Gaussian Markov source of correlation coefficient $\rho = 0.9$ ($N = 8, 16, 32, 64, 128, 256$)

N	Code in Decimal															
16	12	9	4	3	2	10	14	15	13	5	1	0	8	6	11	7
32	1	8	25	16	11	19	18	10	30	6	7	15	5	20	12	29
	9	17	24	26	27	0	2	3	23	22	14	31	4	13	21	28
64	16	13	31	21	28	52	25	55	62	61	56	59	49	43	33	40
	39	46	15	34	50	18	3	10	6	36	0	12	5	23	20	30
	27	24	29	17	60	53	9	57	41	45	63	47	51	35	58	42
	48	54	38	32	2	7	14	4	22	11	8	1	44	19	26	37
128	81	79	14	25	28	73	110	42	104	7	76	107	21	69	97	100
	103	13	37	41	117	44	109	121	47	93	124	61	95	127	126	55
	23	54	51	59	27	30	18	56	48	24	90	88	80	72	83	75
	114	10	86	66	106	34	78	3	0	6	20	17	68	65	96	71
	99	102	9	5	33	12	36	85	116	113	101	77	105	108	119	39
	111	53	43	45	125	40	15	29	31	63	46	62	60	57	89	91
	123	58	122	120	26	92	50	94	82	74	19	22	16	2	64	67
	70	98	1	84	8	87	4	112	118	32	11	115	52	38	49	35
256	13	41	109	73	62	69	93	121	84	213	88	28	1	217	133	81
	76	21	61	25	137	8	157	205	4	44	220	117	140	37	181	253
	185	173	124	188	152	148	236	49	52	145	164	56	168	16	248	233
	105	244	161	241	229	200	176	246	243	202	163	178	196	100	231	32
	96	224	112	198	64	226	102	235	34	128	193	99	195	130	114	208
	66	210	83	82	106	18	86	90	250	147	166	238	215	71	170	75
	3	219	135	139	123	222	43	111	150	154	119	95	207	39	23	51
	27	126	10	6	78	30	14	15	142	46	12	159	63	255	183	191
	187	190	175	189	89	9	54	85	77	5	92	221	58	45	125	29
	141	149	156	153	57	53	60	17	24	169	20	252	172	237	249	245
	184	180	204	177	165	36	108	120	40	104	72	232	136	116	68	48
	132	113	33	97	225	101	65	80	0	227	228	240	201	242	192	160
	230	194	216	212	197	162	209	129	144	98	67	214	199	211	146	131
	218	203	234	107	103	35	115	70	2	87	91	134	74	11	138	7
		79	19	94	223	151	206	118	127	110	155	254	171	31	251	247
	47	179	239	26	22	158	167	59	182	174	38	50	186	122	42	55

Table 3.5: Index assignments found by the simulated annealing algorithm for Gaussian Markov sources with correlation coefficient $\rho = 0.9$ and for MSE ($N = 16, 32, 64, 128, 256$).

N	Code	$C(\pi)$	MSE		
			$p_c = 10^{-1}$	$p_c = 10^{-2}$	$p_c = 10^{-3}$
8	Best	1.2559	0.5822	0.0557	0.0068
	Worst	2.1948	1.1556	0.1119	0.0183
	SA	1.2559	0.5822	0.0557	0.0068
	GC	1.4670	0.6313	0.0637	0.0087
	NBC	1.7795	0.8616	0.0832	0.0098
	FBC	1.7874	0.7892	0.0994	0.0141
	HarperA	1.8561	1.2229	0.0936	0.0118
	HarperB	2.2457	0.1065	0.1100	0.0159
16	SA	0.6890	0.5634	0.0337	0.0033
	GC	1.4505	0.6399	0.0644	0.0071
	NBC	1.4393	0.7538	0.0686	0.0084
	FBC	1.7527	0.7557	0.0815	0.0123
	HarperA	1.3231	1.3109	0.0687	0.0118
	HarperB	1.9991	1.1063	0.0888	0.0109
32	SA	0.4684	0.8639	0.0332	0.0034
	GC	1.4269	0.6194	0.0595	0.0076
	NBC	1.3713	0.7021	0.0583	0.0071
	FBC	1.7456	0.7522	0.0789	0.0095
	HarperA	0.8879	1.3861	0.0471	0.0075
	HarperB	2.0843	1.3154	0.0919	0.0119
64	SA	0.2385	1.0794	0.0332	0.0043
	GC	1.4321	0.6166	0.0579	0.0075
	NBC	1.3457	0.6743	0.0562	0.0065
	FBC	1.7342	0.7583	0.0756	0.0094
	HarperA	0.5502	1.4152	0.0388	0.0049
	HarperB	2.1097	1.5494	0.0891	0.0115
128	SA	0.1193	1.2068	0.0365	0.0032
	GC	1.4415	0.6201	0.0590	0.0071
	NBC	1.3349	0.6619	0.0560	0.0070
	FBC	1.7272	0.7537	0.0698	0.0088
	HarperA	0.3015	1.6752	0.0348	0.0030
	HarperB	2.2378	1.8276	0.0975	0.0134

Continued on next page...

Table 3.6: MSE of different index assignments for Gaussian Markov source of correlation coefficient $\rho = 0.9$ ($N = 8, 16, 32, 64, 128, 256$)

N	Code	$C(\pi)$	MSE		
			$p_c = 10^{-1}$	$p_c = 10^{-2}$	$p_c = 10^{-3}$
256	SA	0.2108	2.0284	0.0431	0.0038
	GC	1.4480	0.6292	0.0595	0.0067
	NBC	1.3465	0.6855	0.0575	0.0063
	FBC	1.7347	0.7644	0.0704	0.0081
	HarperA	0.4371	2.6181	0.0417	0.0035
	HarperB	3.3605	2.2345	0.1392	0.0201

Table 3.6 : Continued from previous page.

N	Code	$C(\pi)$	SER		
			$p_c = 10^{-1}$	$p_c = 10^{-2}$	$p_c = 10^{-3}$
8	SA	1.0982	0.0557	0.0041	0.0005
	GC	3.5629	0.0737	0.0066	0.0010
	NBC	1.7540	0.0646	0.0047	0.0006
	FBC	3.9403	0.0721	0.0063	0.0009
	HarperA	3.1749	0.0593	0.0052	0.0006
	HarperB	2.2693	0.0592	0.0039	0.0005

Table 3.7: Symbol error rate of different index assignments for a non-Gaussian Markov source ($N = 8$)

Chapter 4

Joint Source-Channel Decoding of Multiple Descriptions

4.1 Overview

In this chapter we propose a joint source-channel multiple description framework for distributed communication and estimation of signals. Distributed signal estimation and quantization was treated in the literature [74], and related problem of distributed detection was also studied [75, 76]. But these problems have not been studied in a setting of multiple descriptions. The works of this and next chapters are motivated by modern distributed signal transmission and processing applications in lossy networks populated by resource-deprived transmitters and receivers of varied capabilities.

In the proposed JSC-MD framework, the encoder complexity is kept at a minimum by coding source(s) using a multiple description quantizer, either fixed rate or variable rate. Any channel impairments are to be handled by the redundancy of $K \geq 2$ correlated descriptions rather than forward error correction. If only one description is received a coarse version of the source signal can be reconstructed. More received descriptions can refine each other to obtain an estimation of higher fidelity. The code diversity of MDQ and the path diversity of the network can be exploited by JSC decoding to mitigate transmission errors and improve signal reconstruction.

To achieve resource scalability of the JSC MD decoder we develop a family of JSC estimation techniques of different complexities and performances, ranging from the fast and simple hard-decision decoder to sophisticated graph based decoders.

Conventional MDQ decoder only considers the scenario where multiple descriptions are transmitted over erasure (on-off) channels [63], where each description is either received or lost. Recently, the problem of MD estimation in presence of bit errors is also studied. In [62, 77], a memoryless source encoded by fixed rate two description scalar quantization (2DSQ) is estimated by applying the “turbo” principle: two descriptions are decoded iteratively by exchanging the “soft” information rather than making a hard decision on the information bits. A similar iterative solution is given in [78] for the problem of MAP decoding of variable-length and 2DSQ coded Markov sequences. However, none of these solutions is optimal. The proposed JSC-MD framework adopts a more general channel model that allows both erasure and bit errors, and furthermore we provide exact solutions to the problems of MAP and MMSE estimation of MDQ coded Markov sequences.

When used for MDQ decoding, the proposed JSC-MD approach has an added operational advantage over the current MDQ design. It generates an output sequence (the most probable one given the source and channel statistics) consisting entirely of the codewords of the central quantizer, rather than a mixture of codewords of the central and K side decoders. As such the JSC-MD approach offers a side benefit of unifying the treatment of the 2^K cases for different subsets of received descriptions. Instead of employing $2^K - 1$ decoders as required by the existing MDQ decoding process, we need only one MDQ decoder. This overcomes a great operational difficulty currently associated with the MDQ decoding process.

First we develop JSC MAP and MMSE decoding algorithms for fixed rate MDQ in this chapter, and then extend the JSC MAP decoding algorithm to variable rate MDQ in Chapter 5. The remainder of this chapter is organized as follows. Section 4.2 formulates the JSC-MD problem. Section 4.3 constructs a weighted directed acyclic graph to model the JSC-MD MAP estimation/decoding problem. This graph

construction converts distributed MAP estimation into a problem of longest path in the graph, which is polynomially solvable. The complexity results are derived. Section 4.5 applies the proposed JSC-MD approach to distributed MAP estimation of hidden Markov state sequences in lossy networks. This problem is motivated by sensor networks of heterogeneous nodes with resource scalability requirements. With the same MD code transmitted over the entire network, the empowered MD decoders can obtain exact MAP solution using a graph based algorithm, while deprived MD decoders can obtain approximate solutions using algorithms of various complexities. Section 4.6 investigates the problem of distributed MMSE decoding of MDQ. It turns out that JSC-MD MMSE decoding can be performed by generalizing the well-known forward-backward algorithm to multiple descriptions. Simulation results are reported in Section 4.7. Section 4.8 concludes this chapter.

4.2 Problem Formulation

Figure 4.1 schematically depicts the JSC-MD signal communication system motivated in the introduction. The variant of the JSC-MD system for distributed estimation will be introduced in Section 4.5. In Figure 4.1 the input to the system is a finite Markov sequence $\chi^{\mathcal{N}} = \chi_1, \chi_2, \dots, \chi_{\mathcal{N}}$. A K -description MDQ first maps a source symbol (if multiple description scalar quantization (MDSQ) is used) or a block of source symbols (if multiple description vector quantization (MDVQ) is used) to a codeword of the central quantizer $q : \mathbb{R}^{\iota} \rightarrow \mathbb{C} = \{c_1, c_2, \dots, c_L\}$, where L is the number of codecells of the central quantizer and ι is the VQ dimension. Note that $\iota = 1$ for scalar quantization. Let the codebooks of the K side quantizers be $\mathbb{C}_k = \{c_{k,1}, c_{k,2}, \dots, c_{k,L_k}\}$, $1 \leq k \leq K$, where $L_k \leq L$ is the number of codecells of side quantizer k , $L \leq \prod_{k=1}^K L_k$. The K -description MDQ is specified by K index assignment functions $\lambda_k : \mathbb{C} \rightarrow \mathbb{C}_k$, $1 \leq k \leq K$ [61, 79]. The redundancy carried by the K descriptions versus the single description can be reflected by a rate $1 - \log_2 L / \sum_{k=1}^K \log_2 L_k$.

Due to the expediency on the part of resource-deprived MDQ encoders, a decoder

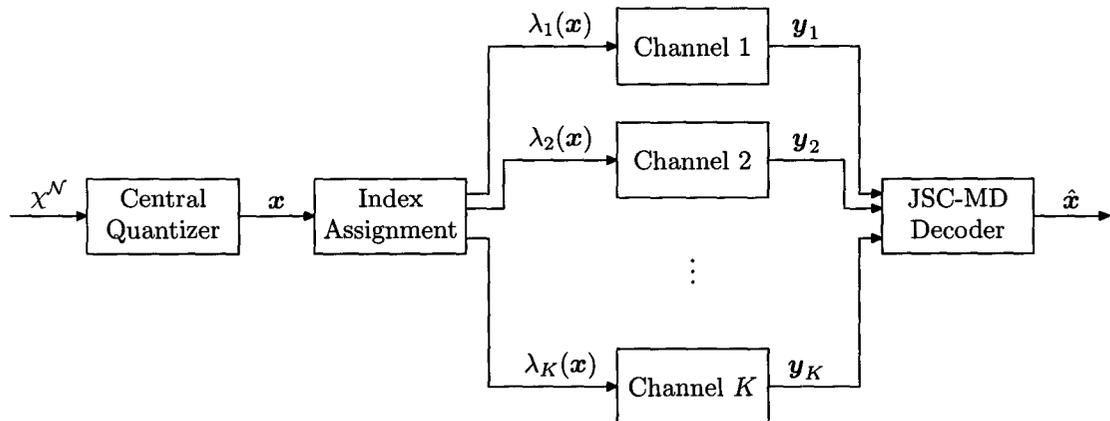


Figure 4.1: Block diagram of a MDQ based communication system with a JSC-MD decoder.

is furnished with rich forms of statistical redundancy:

- the memory of the Markov source that is unexploited by suboptimal source code;
- residual source redundancy for lack of entropy coding;
- the correlation that is intentionally introduced among the K descriptions of MDQ.

The remaining question or challenge is naturally how these intra- and inter-description redundancies can be fully exploited in a distributed resource-constrained environment.

Let $\mathbf{x} = x_1 x_2 \cdots x_N \in \mathbb{C}^N$ be the output sequence of $\chi^{\mathcal{N}}$ produced by the central quantizer, $N = \mathcal{N}$ for MDSQ, or $N = \iota \mathcal{N}$ for MDVQ with ι being the VQ dimension. The K descriptions of MDQ, $\lambda_k(\mathbf{x}) \in \mathbb{C}_k^N$, $1 \leq k \leq K$, are transmitted via K noisy channels. In this work we use a quite general model for the K channels. The only requirements are that these channels are memoryless, independent, and do not introduce phase errors such as insertion or deletion of code symbols or bits. In the existing literature on MDQ, only packet erasure errors are considered in MDQ decoding. Our channel model accommodates bit errors as well. This is an important expansion because bit errors can indeed happen in a received description in reality,

particularly so in wireless network communications. Denote the received code streams by $\mathbf{y}_k = y_{k,1}y_{k,2}\cdots y_{k,N}$, with $y_{k,n}$ being the n^{th} codeword of description k that is observed by the decoder.

Having the source and channel statistics and knowing the structure of MDQ, the decoder can perform JSC-MD decoding of sequences \mathbf{y}_k , $1 \leq k \leq K$, to best reconstruct \mathbf{x} . The JSC criterion can be maximum *a posteriori* probability or minimum mean-square error. For concreteness and clarity, we formulate the JSC-MD problem for distributed MAP decoding of MDQ. As we will see in subsequent sections, the formulation for other distributed sequence estimation and decoding problems requires only minor modifications. In a departure from the current practice of designing multiple side decoders (up to $2^K - 1$ of them!), our JSC-MD system offers a single unified MDQ decoder that operates the same way regardless what subset of the K descriptions are available to the decoder. For JSC decoding of single description scalar quantized Markov sequences, please refer to [26, 31, 25, 27, 29, 32].

In JSC-MD distributed MAP decoding a decoder reconstructs, given the observed sequences \mathbf{y}_k , ($1 \leq k \leq K$, some of which may be empty), the input sequence \mathbf{x} such that the *a posteriori* probability $P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K)$ is maximized. Namely, the MAP MDQ decoder emits

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{C}^N}{\operatorname{argmax}} P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K). \quad (4.1)$$

Comparing the proposed JSC MDQ decoder via distributed MAP sequence estimation with the existing symbol-by-symbol MDQ decoders, one sees an obvious distinction. The JSC decoder always generates codewords of the central quantizer even when it does not have all the K descriptions, while hard-decision MDQ decoders have to face the difficulty that the intersection of side quantizer outputs may not even exist.

By Bayes' theorem we have

$$\begin{aligned}
P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) &= \frac{P(\mathbf{x})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K|\mathbf{x})}{P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K)} \\
&\stackrel{(a)}{\propto} P(\mathbf{x})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K|\mathbf{x}) \\
&= P(\mathbf{x})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K|\lambda_1(\mathbf{x}), \lambda_2(\mathbf{x}), \dots, \lambda_K(\mathbf{x})) \\
&\stackrel{(b)}{=} P(\mathbf{x}) \prod_{k=1}^K P(\mathbf{y}_k|\lambda_k(\mathbf{x})) \\
&\stackrel{(c)}{=} \prod_{n=1}^N \left\{ P(x_n|x_{n-1}) \prod_{k=1}^K P_k(y_{k,n}|\lambda_k(x_n)) \right\}.
\end{aligned} \tag{4.2}$$

In the above derivation, step (a) is due to the fact that \mathbf{y}_1 through \mathbf{y}_K are fixed in the objective function for $\mathbf{x} \in \mathcal{C}^N$; step (b) is because of the mutual independency of the K channels; and step (c) is under the assumption that \mathbf{x} , the output of the central quantizer, is first-order Markov and the channels are memoryless. This assumption is a very good approximation if the original source sequence χ^N before MDQ is first-order Markov, or a high-order Markov sequence χ^N is vector quantized into K descriptions.

In (4.2) we also let $P(x_1|x_0) = P(x_1)$ as convention. $P_k(\mathbf{b}'|\mathbf{b})$ is the probability of receiving a codeword $\mathbf{b} = b_1 b_2 \dots b_B$ from channel k as $\mathbf{b}' = b'_1 b'_2 \dots b'_B$. Because the channel is memoryless, we have

$$P_k(\mathbf{b}'|\mathbf{b}) = \prod_{i=1}^B P_k(b'_i|b_i). \tag{4.3}$$

Specifically, if the K channels can be modeled as memoryless error-and-erasure channels (EEC), where each bit is either transmitted intact, or inverted, or erased (the erasure can be treated as the substitution with a new symbol '\$\$'), then $\mathbf{b} \in$

$\{0, 1\}^B$, $\mathbf{b}' \in \{0, 1, \$\}^B$ and

$$P_k(b'_i|b_i) = \begin{cases} p_{\phi,k}, & \text{if } b'_i = \$; \\ (1 - p_{\phi,k})(1 - p_{c,k}), & \text{if } b'_i = b_i; \\ (1 - p_{\phi,k})p_{c,k}, & \text{otherwise} \end{cases} \quad (4.4)$$

where $p_{\phi,k}$ is the erasure probability and $p_{c,k}$ is the inversion or crossover probability for channel k , $1 \leq k \leq K$.

The proposed JSC-MD framework is also suitable for additive white Gaussian noise (AWGN) channels. If b_i is binary phase-shift keying (BPSK) modulated and transmitted through channel k that is AWGN, then

$$P_k(b'_i|b_i) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(b'_i - b_i)^2 / 2\sigma_k^2} \quad (4.5)$$

where σ_k^2 is the variance of noise of channel k .

The prior distribution $P(x)$ and transition probability matrix $P(x_n|x_{n-1})$ for the first-order Markov sequence \mathbf{x} can be determined from the source distribution and the particular MDQ in question.

In the case of MDSQ, if the stationary probability density function of the source is $p_s(\chi)$ and the conditional probability density function is $p_s(\chi_n|\chi_{n-1})$, then

$$P(x_1) = \int_{\chi:q(\chi)=x_1} p_s(\chi) d\chi \quad (4.6)$$

and

$$P(x_n|x_{n-1}) = \frac{\iint_{\substack{\chi_1:q(\chi_1)=x_n \\ \chi_2:q(\chi_2)=x_{n-1}}} p_s(\chi_1|\chi_2)p_s(\chi_2) d\chi_2 d\chi_1}{\int_{\chi:q(\chi)=x_{n-1}} p_s(\chi) d\chi}. \quad (4.7)$$

If MDVQ is the source coder of the system, the transition probability matrix for $P(x_n|x_{n-1})$'s can be determined numerically either from a known close-form source distribution or from a training set.

In the literature MDQ is mostly advocated as a measure against packet erasure

errors in diversity networks. In this case we can reformulate (4.1) using the received packets instead of all K packets.

4.3 Joint Source-Channel Multiple Description MAP Decoding

In this section, we devise a graph based algorithm for JSC-MD MAP decoding algorithm. Combining (4.1) and (4.2), we have

$$\hat{\mathbf{x}} = \operatorname{argmax}_{\mathbf{x} \in \mathbb{C}^N} \sum_{n=1}^N \left\{ \log P(x_n | x_{n-1}) + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(x_n)) \right\}. \quad (4.8)$$

Because of the additivity of (4.8), we can structure the MAP estimation problem into the following subproblems:

$$w(n, x_n) = \max_{\mathbf{x} \in \mathbb{C}^{n-1}} \sum_{i=1}^n \left\{ \log P(x_i | x_{i-1}) + \sum_{k=1}^K \log P_k(y_{k,i} | \lambda_k(x_i)) \right\}, \quad (4.9)$$

$$x_n \in \mathbb{C}, \quad 1 \leq n \leq N.$$

The subproblems $w(\cdot, \cdot)$ can be expressed recursively as

$$w(n, x_n) = \max_{\mathbf{x} \in \mathbb{C}^{n-1}} \left\{ \sum_{i=1}^{n-1} \left[\log P(x_i | x_{i-1}) + \sum_{k=1}^K \log P_k(y_{k,i} | \lambda_k(x_i)) \right] \right. \\ \left. + \log P(x_n | x_{n-1}) + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(x_n)) \right\} \quad (4.10)$$

$$= \max_{c \in \mathbb{C}} \left\{ w(n-1, c) + \log P(x_n | c) \right\} + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(x_n)).$$

Then, the solution of the optimization problem (4.1) is given recursively in a backward

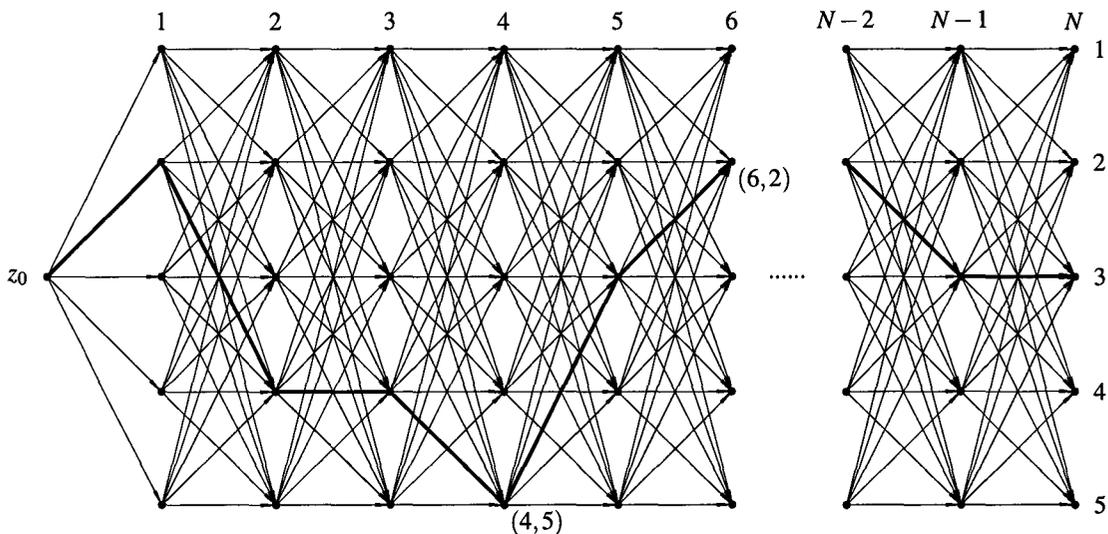


Figure 4.2: Graph G constructed for the JSC-MD MAP decoding ($L = 5$).

manner by

$$\begin{aligned} \hat{x}_N &= \operatorname{argmax}_{c \in \mathbb{C}} w(N, c). \\ \hat{x}_{n-1} &= \operatorname{argmax}_{c \in \mathbb{C}} \left\{ w(n-1, c) + \log P(\hat{x}_n | c) \right\}, \quad 2 \leq n \leq N. \end{aligned} \quad (4.11)$$

The recursion of $w(n, x_n)$ allows us to reduce the MAP estimation problem to one of finding the longest path in a weighted directed acyclic graph (WDAG) [29], as shown in Figure 4.2. The underlying graph G has $LN + 1$ vertices, which consists of N stages with L vertices in each stage. Each stage corresponds to a codeword position in \mathbf{x} . Each vertex in a stage represents a possible codeword at the position. There is also one starting node z_0 , corresponding to the beginning of \mathbf{x} .

In the construction of the graph G , each node is associated with a codeword $x \in \mathbb{C}$ at a sequence position n , $1 \leq n \leq N$, and hence labeled by a pair (n, x) . From node $(n-1, b)$ to node (n, a) , $a, b \in \mathbb{C}$, there is a directed edge, whose weight is

$$\log P(a|b) + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(a)).$$

From the starting node z_0 to each node $(1, a)$, there is an edge whose weight is

$$\log P(a) + \sum_{k=1}^K \log P_k(y_{k,1} | \lambda_k(a)).$$

In graph G , the solution of the subproblem $w(n, a)$ is the weight of the longest path from the starting node s to node (n, a) , which can be calculated recursively using dynamic programming. The MAP decoding problem is then converted into finding the longest path in graph G from the starting node z_0 to nodes $(N, c), c \in \mathbb{C}$. By tracing back step by step to the starting node z_0 as given in (4.11), the MDQ decoder can reconstruct the input sequence \mathbf{x} to $\hat{\mathbf{x}}$, the optimal result defined in (4.1).

Now we analyze the complexity of the proposed algorithm. The dynamic programming algorithm proceeds from the starting node z_0 to the nodes (N, c) , through all LN nodes in G . The value of $w(n, a)$ can be evaluated in $O(L)$ time, according to (4.10). The quantities $\log P(a|b)$ and $\log P_k(y_{k,n} | \lambda_k(a))$ can be precomputed and stored in lookup tables so that they will be available to the dynamic programming algorithm in $O(1)$ time. Hence the term $\sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(a))$ in (4.10) can be computed in $O(K)$ time. Therefore, the total time complexity of the dynamic programming algorithm is $O(L^2NK)$. The reconstruction of the input sequence takes only $O(N)$ time, given that the selections in (4.11) (and in (4.10) as well) are recorded, which results in a space complexity of $O(LN)$.

4.4 Complexity Reduction of JSC-MD MAP Decoding Algorithm

In [29], the recursive formula derived for MAP decoding of Markov sequences transmitted over noisy channels was converted into a matrix search problem. The complexity of the matrix search algorithm can be reduced if the underlying matrix satisfies a property called monotonicity. In this section, we show that this complexity reduction technique can also be applied to the JSC-MD MAP decoding algorithm to reduce the

computational complexity under the same condition.

A two-dimensional matrix $A = A(a, b)$ is said to be *totally monotone* with respect to row maxima if the following relation holds:

$$A(a, b) \leq A(a, b') \Rightarrow A(a', b) \leq A(a', b'), \quad a < a', b < b'. \quad (4.12)$$

A sufficient condition for (4.12) is

$$A(a, b') + A(a', b) \leq A(a, b) + A(a', b'), \quad a < a', b < b' \quad (4.13)$$

which is also known as the Monge condition. If an $n \times n$ matrix A is totally monotone, then the row maxima of A can be found in $O(n)$ time [80].

To show that this linear-time matrix search algorithm can be applied to the JSC-MD MAP decoding algorithm, we need to convert the recursion formula in Chapter 4 into a matrix search form. We rewrite (4.10) as

$$w(n, x_n) = \max_{c \in \mathcal{C}} \left\{ w(n-1, c) + \log P(x_n | c) \right\} + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(x_n)). \quad (4.14)$$

Then for each $1 \leq n \leq N$, we define an $L \times L$ matrix A_n such that

$$A_n(a, b) = w(n-1, b) + \log P(a|b) + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(a)). \quad (4.15)$$

Now one can see that the computation task for JSC-MD MAP decoding is to find the row maxima of matrix A_n .

To apply the linear-time matrix search algorithm to the JSC-MD MAP decoding problem, we only need to show that matrix A_n satisfies the total monotonicity. Substituting A_n in (4.15) for A in (4.13), we have

$$\log P(a|b') + \log P(a'|b) \leq \log P(a|b) + \log P(a'|b'), \quad a < a', b < b' \quad (4.16)$$

which is a sufficient condition for A_n to have the total monotonicity and therefore, for the fast algorithm to be applicable. This condition, which depends only on the source statistics not the channels, is exactly the same as the one derived in [29]. It was shown by [29] that (4.16) holds if the source is Gaussian Markov, which includes a large family of signals studied in practice and theory. Accordingly, the time complexity of MAP decoding of MDSQ can be reduced to $O(LNK)$ for Gaussian Markov sequences. The linear dependency of the MAP MDSQ decoding algorithm in the sequence length N and source codebook size L makes it comparable to the complexity of typical Viterbi-type decoders for single description.

4.5 Distributed Multiple-Description Estimation of Hidden Markov Sequences

In this section we apply the proposed JSC-MD MAP estimation technique to solve the problem of hidden Markov sequence estimation in a resource-constrained sensor network.

4.5.1 Problem Setup

Consider, for example, a sensor network in an inaccessible area to monitor the local weather system for years with no or little maintenance. The objective is to remotely estimate the time sequence of weather patterns: sunny, rainy, cloudy and so on, which are not directly observable by processing nodes in the sensor network. Some of the processing nodes are well-equipped and easily-maintained processing centers, while others need to run autonomously on limited power supply and react to certain weather conditions on their own rather than being instructed by the central control.

The time series of weather patterns can be viewed as the state sequence of a hidden Markov model (HMM). Let the state space of the HMM be $\mathbb{S} = \{s_1, s_2, \dots, s_M\}$, s_m for sunny, rain, etc. A weather change from s_i to s_j has the state transition

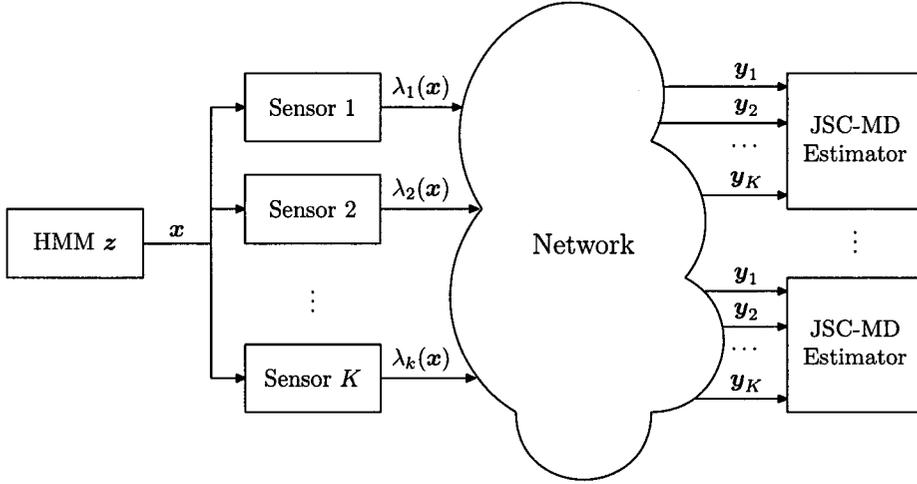


Figure 4.3: JSC-MD sequence estimation in sensor network.

probability $P_S(s_j|s_i)$. The corresponding HMM outputs, which are observed and communicated by sensors, are a real-valued feature vector $\mathbf{x} \in \mathbb{R}^d$ (temperature, pressure, moisture, wind speed, etc.) of probability $P_O(\mathbf{x}|s_i, s_j)$. The communication of observed HMM output sequence $\mathbf{x} = x_1 x_2 \dots$ is conducted at a low bit rate against channel noise and losses. To maximize their operational lifetime the sensors have to do without sophisticated source coding and forgo channel coding altogether. A viable solution under such stringent conditions is to produce and transmit $K \geq 2$ fixed-rate descriptions of \mathbf{x} without entropy coding.

One technique to produce K fixed-rate descriptions of \mathbf{x} at low cost is multiple description lattice vector quantizer (MDLVQ) [79, 61]. Another technique is the use of multiple collaborative sensors to generate multiple descriptions directly. Each element of feature vector $\mathbf{x} \in \mathbb{R}^d$ can be measured by one or few primitive sensor(s). Those measurements are in fact correlated multiple descriptions. In either design the problem at processing nodes is one of JSC-MD sequence estimation. Let $\lambda_k(\mathbf{x})$, $1 \leq k \leq K$, be the K description sequences of \mathbf{x} , which are transmitted through K diversity paths in the sensor network. The JSC-MD system is illustrated in Figure 4.3.

4.5.2 Distributed MAP Sequence Estimation

Let $\mathbf{y}_k = y_{k,1}y_{k,2}\cdots y_{k,N}$ be the received sequence from diversity channel k , $1 \leq k \leq K$. Our task is to estimate the hidden state sequence $\mathbf{z} = z_1z_2\cdots z_N \in \mathbb{S}^N$ of weather patterns, given the K noisy time sequences of atmosphere attributes produced by the HMM: $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K$. With the resource-scalability in mind, we take an approach of MAP estimation:

$$\hat{\mathbf{z}}, \hat{\mathbf{x}} = \underset{\mathbf{z} \in \mathbb{S}^N, \mathbf{x} \in \mathbb{C}^N}{\operatorname{argmax}} P(\mathbf{x}, \mathbf{z} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K). \quad (4.17)$$

Analogously to (4.2) we use Bayes' theorem and the independence of the K memoryless channels to obtain

$$\begin{aligned} P(\mathbf{x}, \mathbf{z} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) &\propto P(\mathbf{z})P(\mathbf{x} | \mathbf{z})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K | \mathbf{x}, \mathbf{z}) \\ &= \prod_{n=1}^N \left\{ P_S(z_n | z_{n-1}) P_O(x_n | z_n, z_{n-1}) P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K | \mathbf{x}) \right\} \\ &= \prod_{n=1}^N \left\{ P_S(z_n | z_{n-1}) P_O(x_n | z_n, z_{n-1}) \prod_{k=1}^K P_k(y_{k,n} | \lambda_k(x_n)) \right\}. \end{aligned} \quad (4.18)$$

We can also devise a graph based algorithm to solve the sequence estimation problem of (4.18). Combining (4.17) and (4.18) and taking logarithm, we have

$$\hat{\mathbf{z}}, \hat{\mathbf{x}} = \underset{\mathbf{z} \in \mathbb{S}^N, \mathbf{x} \in \mathbb{C}^N}{\operatorname{argmax}} \sum_{n=1}^N \left\{ \log P_S(z_n | z_{n-1}) + \log P_O(x_n | z_n, z_{n-1}) + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(x_n)) \right\}. \quad (4.19)$$

Here we make the convention that $P_S(z_1 | z_0) = P(z_1)$ and $P_O(x_1 | z_1, z_0) = P(x_1 | z_1)$.

Then, the MAP estimate of the sequence of hidden Markov states is given by

$$\hat{\mathbf{z}} = \operatorname{argmax}_{\mathbf{z} \in \mathbb{S}^N} \sum_{n=1}^N \left\{ \log P_S(z_n | z_{n-1}) + \xi(z_n, z_{n-1}) \right\} \quad (4.20)$$

where

$$\xi(z_n, z_{n-1}) = \max_{x \in \mathbb{C}} \left\{ \log P_O(x | z_n, z_{n-1}) + \sum_{k=1}^K \log P_k(y_{k,n} | \lambda_k(x)) \right\}.$$

Using the same technique as used in (4.10), we structure the above optimization problem into a nested set of subproblems:

$$w(n, z_n) = \max_{\mathbf{z} \in \mathbb{S}^{n-1}} \sum_{i=1}^n \left\{ \log P_S(z_i | z_{i-1}) + \xi(z_i, z_{i-1}) \right\}, \quad z_n \in \mathbb{S}, \quad 1 \leq n \leq N \quad (4.21)$$

which can be expressed recursively by

$$w(n, z_n) = \max_{s \in \mathbb{S}} \left\{ w(n-1, s) + \log P_S(z_n | s) + \xi(z_n, s) \right\}. \quad (4.22)$$

This recursion form also enables us to solve the sequence estimation problem of (4.18) by finding the longest path in a WDAG. The WDAG G contains $MN + 1$ vertices: a starting node z_0 and N stages with M vertices in each stage. Each stage corresponds to a position in time sequence \mathbf{z} . Each vertex in a stage represents a possible HMM state at the position. The starting node z_0 corresponds to the beginning of the sequence \mathbf{z} . From node $(n-1, a)$ to node (n, b) , $a, b \in \mathbb{S}$, there is a directed edge with weight:

$$\log P_S(b|a) + \xi(b, a).$$

From the starting node z_0 to each node $(1, a)$, there is an edge whose weight is

$$\log P_S(a) + \max_{x \in \mathbb{C}} \left\{ \log P_O(x|a) + \sum_{k=1}^K \log P_k(y_{k,1} | \lambda_k(x)) \right\}.$$

In graph G , the solution of the subproblem $w(n, s)$ is the weight of the longest path

from the starting node z_0 to node (n, s) , which can be calculated recursively using dynamic programming. The distributed MAP estimation problem is then converted into finding the longest path in graph G from the starting node z_0 to nodes (N, s) , $s \in \mathbb{S}$. Tracing back step by step to the starting node z_0 generates the optimally estimated HMM state sequence \hat{z} .

To analyze the complexity of the proposed algorithm, we notice that the dynamic programming algorithm proceeds through all MN nodes in G . The value of $w(n, s)$ can be evaluated in $O(M)$ time, according to (4.22). The quantities $\log P_S(b|a)$ and $\log P_O(x|z_n, s)$ can be precomputed and stored in lookup tables so that they will be available in the dynamic programming process in $O(1)$ time. The term $\xi(b, a)$ can be computed in $O(KL)$ time. Therefore, the total time complexity of this algorithm is $O(M^2NKL)$. The space complexity is $O(MN)$.

4.5.3 Resource Scalability

If a network node is not bounded by energy and computing resources, it can use the relatively expensive MAP algorithm that taps all available redundancies to obtain the best estimate of HMM state sequence, knowing the statistics of HMM and underlying noisy channels. This JSC-MD framework can be used as an asymmetric codec in the Wyner-Ziv spirit, which strips the encoders to the bone while empowering the decoders. More importantly, it also offers a resource-scalability. If a node in the sensor network needs to estimate z but is severely limited in resources, it can still do so using the same MDQ code, albeit probably at a lesser estimation accuracy. The simplest hence most resource-conserving approximate solution is to first perform a hard-decision MDQ decoding of received descriptions $y_{k,n}$'s to \hat{x}_n , and then estimate z_n to be

$$\hat{z}_n = \operatorname{argmax}_{z \in \mathbb{S}} P_S(z) P_O(\hat{x}_n|z). \quad (4.23)$$

The hard-decision MDQ decoding takes only $O(K)$ operations. Also, in the above approximation, we replace $P_S(z_n|z_{n-1})$ by $P_S(z_n)$ and $P_O(\hat{x}_n|z_n, z_{n-1})$ by $P_O(\hat{x}_n|z_n)$

in (4.18). This is to minimize the resource requirement for estimating z by ignoring the source memory. Consequently, the total time complexity of the fast algorithm reduces to $O(N(K + M))$, as opposed to $O(M^2NKL)$ for the full fledged MAP sequence estimation algorithm. The space requirement drops even more drastically to $O(M + K)$ from $O(MN)$.

Between the exact $O(M^2NKL)$ graph based algorithm and the least expensive $O(N(K + M))$ algorithm, many trade-offs can be made between the resource level and performance of the decoder. For instance, another possible JSC-MD estimation emerges:

$$\hat{z}_n = \operatorname{argmax}_{z \in \mathcal{S}} P_S(z) P_O \left(\max_{x \in \mathcal{C}} \prod_{k=1}^K P_k(y_{k,n} | \lambda_k(x)) \middle| z \right). \quad (4.24)$$

This leads to an $O(N(KL + M))$ HMM state sequence estimation algorithm. The algorithm is slightly more expensive than the one based on (4.23) but offers better performance because the MDQ decoding is done with the knowledge of channel statistics.

4.5.4 Application in Distributed Speech Recognition

Given the success of HMM in speech recognition [81], we envision the potential use of the JSC-MD estimation technique for remote speech recognition. For instance, the cell phones transmit quantized speech signals via channels to processing centers and the recognized texts are sent back or forward to other destinations. This will offer mobile users speech recognition functionality without requiring heavy computing power on handsets and fast draining batteries. Also, the network speech recognizer can prompt a user to repeat in case of difficulties, the user's revocalization can be used as extra descriptions to improve the JSC-MD estimation performance.

4.6 Resource-scalable JSC-MD MMSE Decoding

The JSC-MD distributed MAP estimation problem discussed above is to track the discrete states of a hidden Markov model. Likewise, the cost function (4.1) for distributed MAP decoding of MDQ requires the output symbols to be discrete codewords of the central quantizer. This may be desirable or even necessary, if the quantizer codewords communicated correspond to discrete states of semantic meanings, such as in some recognition and classification applications. But in network communication of a continuous signal $\chi^N = \chi_1, \chi_2, \dots, \chi_N$, the JSC-MD output can be real valued. In this case a JSC-MD distributed MMSE decoding scheme of resource scalability is preferred, which is the topic of this section.

The goal of the JSC-MD MMSE decoding is to reconstruct χ_n such that its mean squared error is minimized. This problem can be solved by calculating the conditional expectation of χ_n as

$$E(\chi_n | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) = \sum_{l=1}^L P(x_n = l | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) \frac{\int_{\chi \in V_l} \chi p(\chi) d\chi}{\int_{\chi \in V_l} p(\chi) d\chi} \quad (4.25)$$

where V_l is cell l of the central quantizer. Hence we need to estimate the *a posteriori* probability $P(x_n | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K)$. Equivalently, we estimate

$$P(x_n = l, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K), \quad l \in \mathcal{C}. \quad (4.26)$$

We can solve the above estimation problem for the JSC-MD distributed MMSE decoding by extending the well-known BCJR (forward-backward) algorithm [12] to multiple observation sequences. For notational convenience let $\mathbf{y}_k^{a \sim b}$, $a < b$, be the consecutive subsequence $y_{k,a}, y_{k,a+1}, \dots, y_{k,b}$ of an observation sequence \mathbf{y}_k . Define

$$\begin{aligned} \alpha_n(l) &= P(x_n = l, \mathbf{y}_1^{1 \sim n}, \mathbf{y}_2^{1 \sim n}, \dots, \mathbf{y}_K^{1 \sim n}) \\ \beta_n(l) &= P(\mathbf{y}_1^{n+1 \sim N}, \mathbf{y}_2^{n+1 \sim N}, \dots, \mathbf{y}_K^{n+1 \sim N} | x_n = l) \\ \gamma_n(l', l) &= P(x_n = l, y_{1,n}, y_{2,n}, \dots, y_{K,n} | x_{n-1} = l'). \end{aligned} \quad (4.27)$$

Then we have

$$\begin{aligned}
P(x_n = l, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) &= P(x_n = l, \mathbf{y}_1^{1 \sim n}, \mathbf{y}_2^{1 \sim n}, \dots, \mathbf{y}_K^{1 \sim n}) \\
&\quad \cdot P(\mathbf{y}_1^{n+1 \sim N}, \mathbf{y}_2^{n+1 \sim N}, \dots, \mathbf{y}_K^{n+1 \sim N} | x_n = l) \\
&= \alpha_n(l) \cdot \beta_n(l).
\end{aligned} \tag{4.28}$$

The last step is due to the fact that $\mathbf{y}_k^{1 \sim n}$ and $\mathbf{y}_k^{n+1 \sim N}$ are independent given x_n . The terms $\alpha_n(l)$ and $\beta_n(l)$ can be recursively computed by

$$\begin{aligned}
\alpha_n(l) &= \sum_{l'=0}^{L-1} P(x_{n-1} = l', x_n = l, \mathbf{y}_1^{1 \sim n}, \mathbf{y}_2^{1 \sim n}, \dots, \mathbf{y}_K^{1 \sim n}) \\
&= \sum_{l'=0}^{L-1} \left\{ P(x_{n-1} = l', \mathbf{y}_1^{1 \sim n-1}, \mathbf{y}_2^{1 \sim n-1}, \dots, \mathbf{y}_K^{1 \sim n-1}) \right. \\
&\quad \left. \cdot P(x_n = l, y_{1,n}, y_{2,n}, \dots, y_{K,n} | x_{n-1} = l') \right\} \\
&= \sum_{l'=0}^{L-1} \alpha_{n-1}(l') \cdot \gamma_n(l', l),
\end{aligned} \tag{4.29}$$

and

$$\begin{aligned}
\beta_n(l) &= \sum_{l'=0}^{L-1} P(x_{n+1} = l', \mathbf{y}_1^{n+1 \sim N}, \mathbf{y}_2^{n+1 \sim N}, \dots, \mathbf{y}_K^{n+1 \sim N} | x_n = l) \\
&= \sum_{l'=0}^{L-1} \left\{ P(x_{n+1} = l', y_{1,n+1}, y_{2,n+1}, \dots, y_{K,n+1} | x_n = l) \right. \\
&\quad \left. \cdot P(\mathbf{y}_1^{n+2 \sim N}, \mathbf{y}_2^{n+2 \sim N}, \dots, \mathbf{y}_K^{n+2 \sim N} | x_{n+1} = l') \right\} \\
&= \sum_{l'=0}^{L-1} \gamma_{n+1}(l, l') \cdot \beta_{n+1}(l').
\end{aligned} \tag{4.30}$$

By definition the term $\gamma_n(\cdot, \cdot)$ can be computed by

$$\begin{aligned}
\gamma_n(l', l) &= P(x_n = l, y_{1,n}, y_{2,n}, \dots, y_{K,n} | x_{n-1} = l') \\
&= P(x_n = l | x_{n-1} = l') \cdot Pr(y_{1,n}, y_{2,n}, \dots, y_{K,n} | x_n = l) \\
&= P(x_n = l | x_{n-1} = l') \cdot \prod_{k=1}^K P_k(y_{k,n} | \lambda_k(l)).
\end{aligned} \tag{4.31}$$

Now we analyze the complexity of the proposed JSC-MD MMSE algorithm. For each n , $1 \leq n \leq N$, we need to calculate the value of $\alpha_n(l)$, $\beta_n(l)$ and $\gamma_n(l', l)$. As explained in the complexity analysis of JSC-MD MAP algorithm, the term

$$\sum_{k=1}^K P_k(y_{k,n} | \lambda_k(a))$$

in (4.31) can be computed in $O(K)$ time. Thus, the matrix $\gamma_n(l', l)$, $(l, l') \in \mathbb{C}^2$, can be computed in $O(L^2 K)$ time. The value of $\alpha_n(l)$ and $\beta_n(l)$ can be computed in $O(L)$ time. Therefore, the total complexity is $O(L^2 KN)$, which has the same order with the complexity of JSC-MD MAP algorithm as derived in Section 4.3.

On the other hand, if the input sequence \mathbf{x} is i.i.d. equation (4.28) is reduced to

$$\begin{aligned}
P(x_n = l, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) &= P(x_n = l, y_{1,n}, y_{2,n}, \dots, y_{K,n}) \\
&= P(x_n = l) \cdot P(y_{1,n}, y_{2,n}, \dots, y_{K,n} | x_n = l) \\
&= P(x_n = l) \cdot \prod_{k=1}^K P_k(y_{k,n} | \lambda_k(l)), \quad 1 \leq n \leq N.
\end{aligned} \tag{4.32}$$

This is also the scheme for hard-decision MDQ MMSE decoding in midst of inversion and erasure errors.

If sequence \mathbf{x} is i.i.d. the complexity of JSC-MD MMSE decoding is reduced to $O(LKN)$ as exhibited by (4.32). For memoryless sources, MMSE sequence estimation is degenerated to MMSE symbol-by-symbol decoding. Even if \mathbf{x} is not memoryless, in consideration of resource scalability, (4.32) can still be used as a less demanding

alternative for network nodes not having sufficient resources to perform full-fledged JSC-MD MMSE decoding. The approximation is good if the source memory is weak. To the extreme, the weakest network nodes of severe source constraints can always resort to a hard-decision MD decoding, which takes only $O(KN)$ time to decode a multiple-description coded sequence \mathbf{x} of length N . The important point is that all three decoders of complexities ranging from $O(L^2KN)$ to $O(KN)$ operate on the same MD code streams distributed in the network. The reader can continue to the next section for further discussions on the issue of resource scalability.

4.7 Simulation Results

The proposed resource-aware JSC-MD distributed MAP and MMSE decoding algorithms are implemented and evaluated via simulations. The simulation inputs are first-order, zero-mean, unit-variance Gaussian Markov sequences of different correlation coefficient ρ . A fixed-rate 2DSQ proposed in [61] is used as the encoder in our simulations. The 2DSQ is uniform and is specified by the index assignment matrix shown in Figure 2.3(b) [61]. The central quantizer has $L = 21$ codecells and the two side quantizers each has $L_1 = L_2 = 8$ codecells. For each description k , $k = 1, 2$, the codeword index $\lambda_k(x)$ is transmitted in fixed-length code of three bits.

The channels are simulated to be error-and-erasure channels with identical erasure probability p_ϕ and inversion probability p_c varying. We report and discuss below the simulation results for different combinations of p_c , p_ϕ and ρ .

First, we evaluate the performance of the JSC-MD distributed MAP decoder. The performance measure is symbol error rate (SER), which is the probability that a symbol of the input Markov sequence is incorrectly decoded. Since the input source is Gaussian Markov, the $O(LKN)$ MAP algorithm of Section 4.3 can be used by the resource-rich network nodes to obtain the optimal estimation. However, resource-deprived network nodes can also decode whatever received description(s) of the same 2DSQ code, using a simple energy-conserving $O(KN)$ hard-decision MDQ decoder as

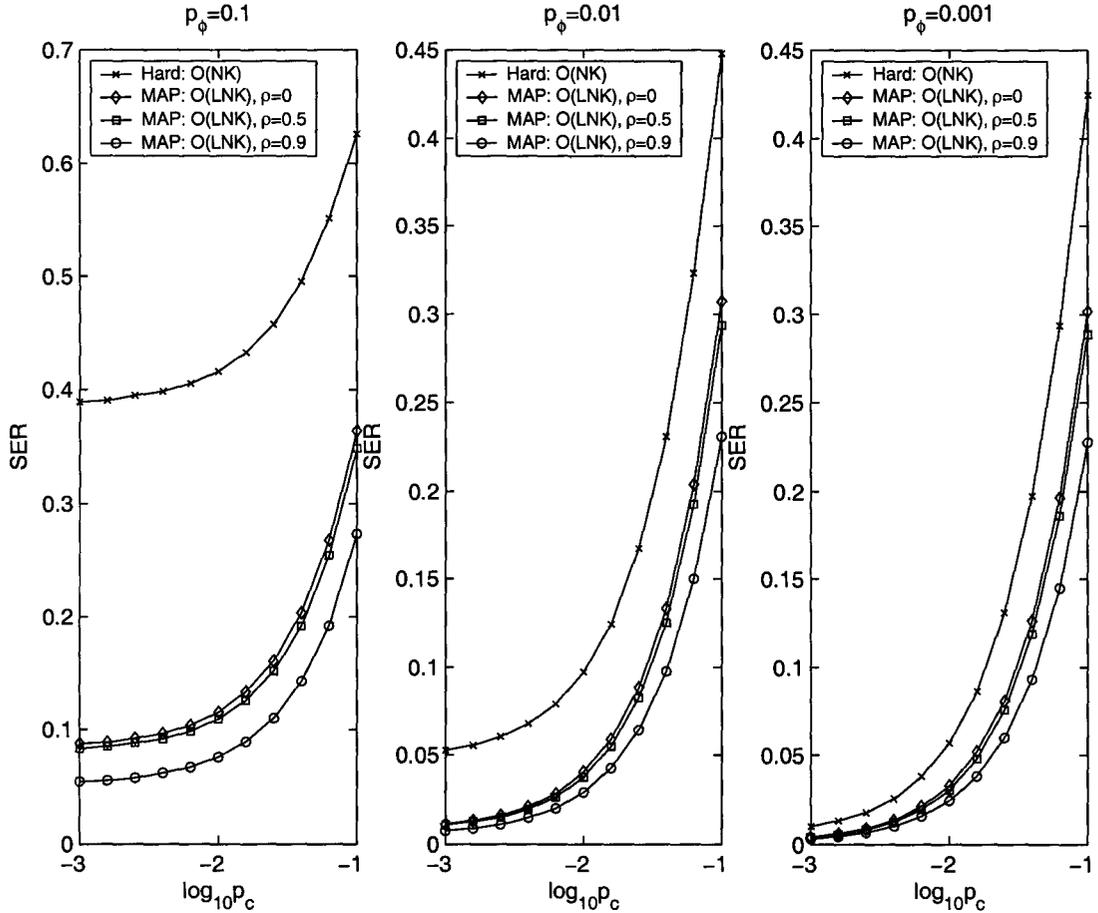


Figure 4.4: Symbol error rates of JSC-MD distributed MAP decoder and MDQ hard-decision decoder with $\rho = 0, 0.5, 0.9$.

given in (4.23). The simulation results are plotted in Figure 4.4. Over all values of ρ , p_c and p_ϕ , the JSC-MD MAP decoder outperforms the hard-decision MDQ decoder. As expected, the performance gap between the two decoders increases as the amount of memory in the Markov source (ρ) increases. This is because the hard-decision MDQ decoder cannot benefit from the residual source redundancy left by the suboptimal primitive 2DSQ encoder.

In the case of JSC-MD distributed MMSE decoding, we evaluate three decoders of different complexities (hence different resource requirements): the exact $O(L^2KN)$ algorithm derived in (4.28), the simplified $O(LKN)$ algorithm given in (4.32), and the

conventional $O(KN)$ hard-decision MDQ decoder which calculates the intersection of side quantizer decoder outputs. The performance measure for MMSE decoding is naturally the signal-to-noise ratio. The simulation results are plotted in Figure 4.5-4.7, with the correlation coefficient being 0, 0.5 and 0.9 respectively. The trade-offs between the complexity and performance of a decoder can be clearly seen in these figures. Given ρ, p_c, p_ϕ , the SNR increases as the decoder complexity increases. The JSC-MD MMSE decoder achieves the highest SNR, because it utilizes both inter- and intra-description correlations. The performance of the algorithm given in (4.32) is in the middle, which is $O(L)$ faster than the full-fledged JSC-MD MMSE decoder but $O(L)$ slower than the hard-decision MDQ decoder. This decoder reduces complexity or energy requirement by making a good use of the inter-description correlation only. The hard-decision MDQ decoder is the simplest and fastest but has the lowest SNR. This is because it ignores the intra-description correlation and does not make good use of the inter-description correlation. As in the MAP case, the performance gap between the first two MMSE decoders increases as the intra-description redundancy (ρ) increases. When $\rho = 0$, the first two algorithms become the same.

Under both MAP and MMSE criteria, the performance gap between different algorithms increases as the erasure error probability p_ϕ increases, indicating that the JSC-MD distributed decoder can make a better use of inter-description correlation in the event of packet loss. As the erasure error probability increases in the network, the proposed JSC-MD decoder enjoys up to 8 dB gain over the hard-decision MD decoders.

Finally, we point out that even when source memory is weak (see the curves for $\rho = 0$), the JSC-MD distributed decoders still have an advantage over the hard-decision MDQ decoders that cannot handle the bit errors within a received description effectively.

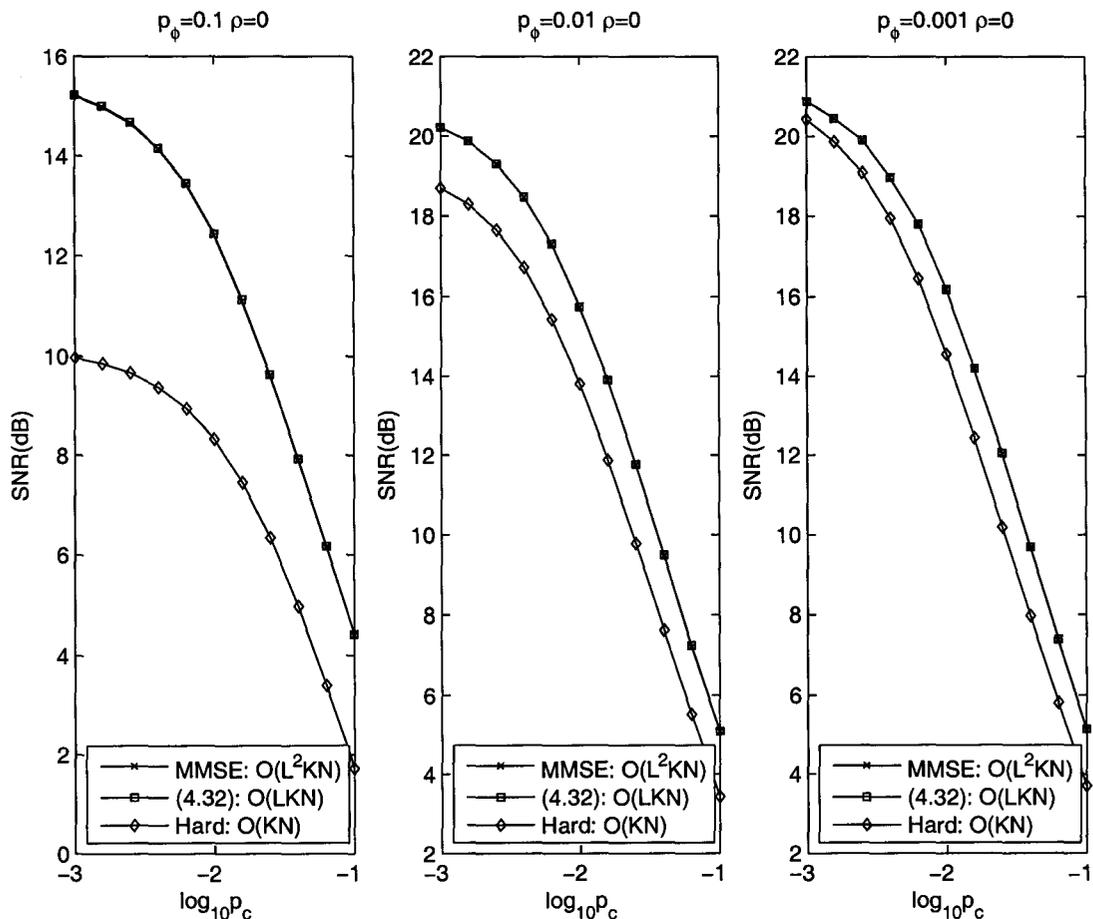


Figure 4.5: SNR performances of different MDQ decoders ($\rho = 0$).

4.8 Conclusions

We propose a joint source-channel multiple description approach to resource-scalable network communications. The encoder complexity is kept to the minimum by fixed rate multiple description quantization. The resulting MD code streams are distributed in the network and can be reconstructed to different qualities depending on the resource levels of receiver nodes. Algorithms for distributed MAP and MMSE sequence estimation are developed, and they exploit intra- and inter-description redundancies jointly to correct both bit errors and erasure errors. The new algorithms outperform the existing hard-decision MDQ decoders by large margins (up to 8dB). If the source

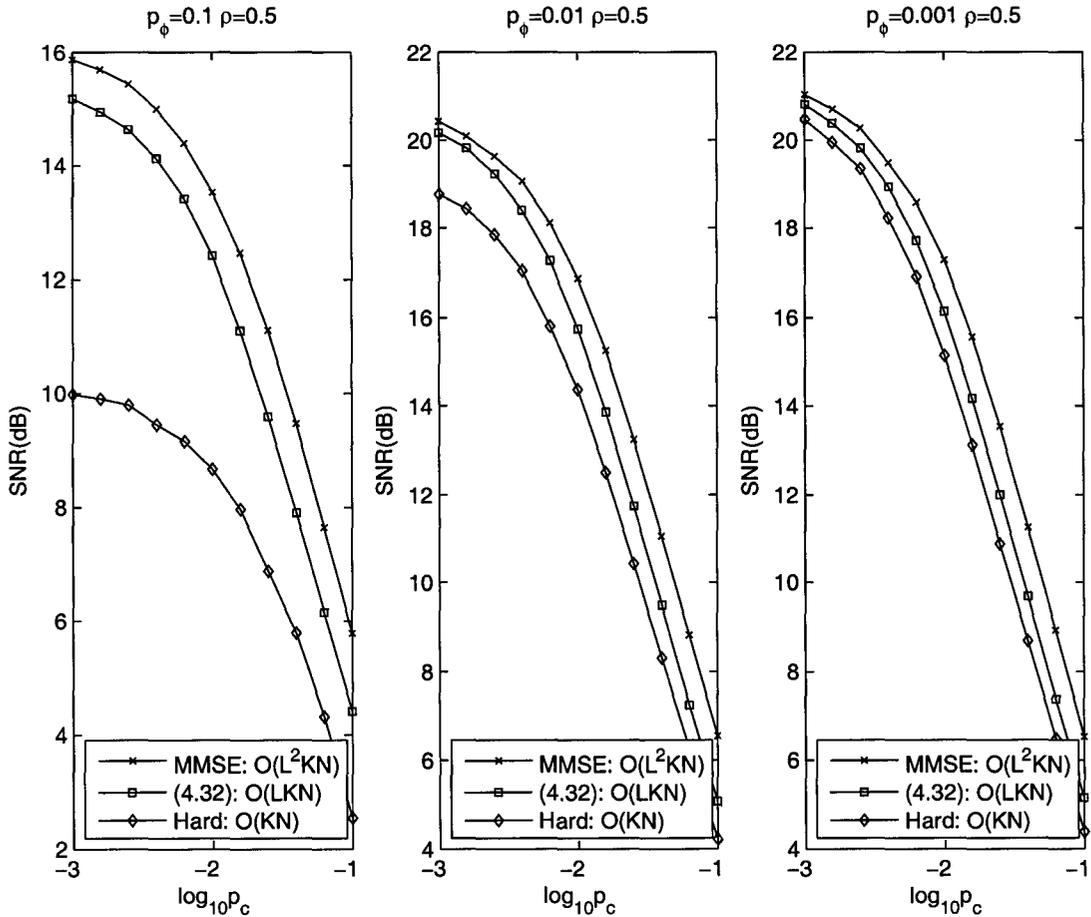


Figure 4.6: SNR performances of different MDQ decoders ($\rho = 0.5$).

is Gaussian Markov, the complexity of the JSC-MD distributed MAP estimation algorithm is $O(LNK)$, which is the same as the classic Viterbi algorithm for single description.

Operationally, the new MDQ decoding technique unifies the treatments of different subsets of descriptions available at a decoder, overcoming the difficulty of having a large number of side decoders that hinders the design of a good hard-decision MDQ decoder.

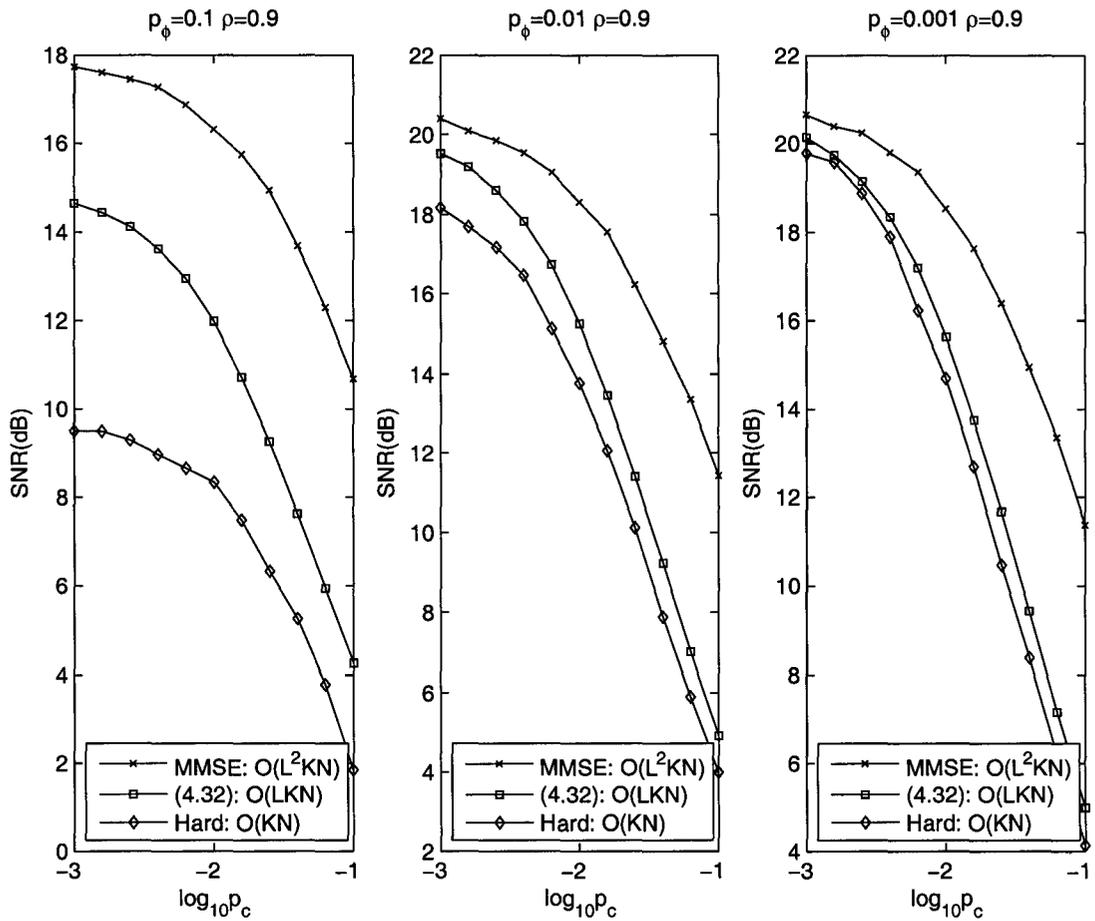


Figure 4.7: SNR performances of different MDQ decoders ($\rho = 0.9$).

Chapter 5

Joint Source-Channel MAP Decoding of Variable-Length Coded Multiple Descriptions

5.1 Overview

In the preceding chapter, we proposed a JSC-MD framework to estimate fixed-rate multiple description coded Markov sequences. We now extend this work to variable-rate multiple description codes. Variable-length code achieves a rate closer to the entropy, but it is very sensitive to channel noise. A minor channel error can cause loss of synchronization of source symbols. The desynchronization can cause complete failure of a hard-decision multiple description decoder for the following two reasons. First, the intersection of multiple desynchronized descriptions may be empty. Second, the length of a decoded description may also be different. To solve this problem, we propose a JSC variable-length coded multiple description (VLCMD) MAP decoding algorithm that can simultaneously utilize the inter-description and intra-description redundancies, circumventing the difficulty of hard-decision decoding in merging multiple descriptions. In contrast, the previous work [78] on JSC decoding of MDSQ and VLC coded Markov sequences was a separation approach: the memory of the

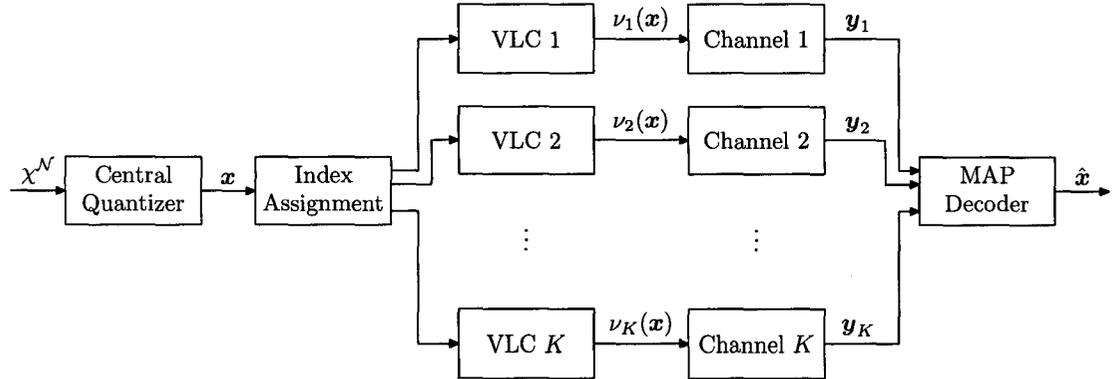


Figure 5.1: Block diagram of a MDSQ based communication system with a MAP decoder.

Markov source and the correlation between the descriptions of MDSQ were exploited in tandem.

The above problem can also be posed as one of MAP sequence estimation, and be solved by a graph based algorithm which is similar to the one in Chapter 4. The difference is that in Chapter 4, the graph is a two dimensional symbol-based trellis, while in this chapter, it is a $K + 1$ dimensional bit-based trellis and the trellis structure depends on the design of the particular VLC. For MDSQ-coded Gaussian Markov sequences the algorithm complexity can also be reduced.

The rest of this chapter is structured as follows. Section 5.2 formulates a general framework for joint source-channel MAP decoding of MDQ-coded Markov sequences. Section 5.3 presents a longest path algorithm for solving the MAP MDQ decoding problem. In Section 5.4, a more efficient solution is developed for Gaussian Markov sequences. Simulation results are reported in Section 5.5. Section 5.6 concludes.

5.2 Problem Formulation

Figure 5.1 schematically depicts the proposed joint source-channel MDQ decoding system. The notions are directly borrowed from Chapter 4, which are briefly summarized as follows. The input to the system is a finite Markov sequence χ^N . A central

quantizer $q : \mathbb{R}^L \rightarrow \mathbb{C}$ maps a source symbol (MDSQ) or a block of source symbols (MDVQ) to a codeword in central codebook $\mathbb{C} = \{c_1, c_2, \dots, c_L\}$, where L is the number of code cells of the central quantizer. $\mathbf{x} = x_1 x_2 \dots x_N \in \mathbb{C}^N$ is the output sequence of χ^N produced by the central quantizer, where N is the number of symbols in \mathbf{x} .

Unlike the definition in Chapter 4, the codebooks of the k 'th side quantizer, denoted $\mathbb{C}_k = \{c_{k,1}, c_{k,2}, \dots, c_{k,L_k}\}$, is a VLC codebook. In other words, the number of bits contained in two codewords can be different, i.e., there exists a pair i, j , $1 \leq i, j \leq L_k$ such that $|c_{k,i}| \neq |c_{k,j}|$, where $|\cdot|$ is the number of bits in a bitstream. The K -description variable-rate MDQ is then specified by K index assignment functions $\nu_k : \mathbb{C} \rightarrow \mathbb{C}_k$ [79].

The K descriptions of MDQ, $\nu_k(\mathbf{x}) \in \mathbb{C}_k^N$, $k = 1, 2, \dots, K$, are transmitted via K noisy channels. We also assume that the K noisy channels are memoryless, mutually independent, and do not introduce phase errors such as insertion or deletion of code symbols or bits. Consequently, a received description may have inversion or/and erasure errors, but it has the same number of bits as the one generated by MDQ. Denote the received code streams by \mathbf{y}_k , with length $N_k = |\nu_k(\mathbf{x})| = \sum_{n=1}^N |\nu_k(x_n)|$.

Since VLC is used, the parsing of \mathbf{y}_k is not unique. Any given \mathbf{x} with $|\nu_k(\mathbf{x})| = N_k$ uniquely determines a parsing of \mathbf{y}_k , which is called the parsing of \mathbf{y}_k with respect to \mathbf{x} . It parses the bit stream \mathbf{y}_k into a sequence of codewords delimited by $(b_{k,0}, b_{k,1}, \dots, b_{k,M})$. We write the n^{th} codeword parsed out of \mathbf{y}_k as $y_k(b_{k,m-1}, b_{k,m}]$, where $b_{k,0} = 0$, $b_{k,m} - b_{k,m-1} = |\nu_k(x_m)|$, $1 \leq m \leq M$ and $b_{k,M} = N_k$. Note that M can be different from N due to phase errors of the parsing.

Having the source and channel statistics and knowing the design of MDQ, the MDQ decoder can perform joint source-channel decoding of \mathbf{y}_k , $k = 1, 2, \dots, K$, to best reconstruct \mathbf{x} . We develop a single unified MDQ decoder that operates the same way regardless what subset of the K descriptions is available to the decoder. Our MDQ decoder takes the approach of MAP sequence estimation, and it reconstructs, given the observed sequences \mathbf{y}_k , $k = 1, 2, \dots, K$, the input sequence \mathbf{x} such that the

a *posteriori* probability $P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K)$ is maximized. Namely,

$$\begin{aligned}\hat{\mathbf{x}} &= \operatorname{argmax}_{\mathbf{x} \in \mathbb{C}^*} P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) \\ &= \operatorname{argmax}_{\mathbf{x} \in \mathbb{C}^*} \log P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K).\end{aligned}\quad (5.1)$$

Similar to the derivation of (4.2) in Chapter 4, we have

$$\begin{aligned}P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K) &= \frac{P(\mathbf{x})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K|\mathbf{x})}{P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K)} \\ &\propto P(\mathbf{x})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K|\mathbf{x}) \\ &= P(\mathbf{x})P(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K|\nu_1(\mathbf{x}), \nu_2(\mathbf{x}), \dots, \nu_K(\mathbf{x})) \\ &= P(\mathbf{x}) \prod_{k=1}^K P(\mathbf{y}_k|\nu_k(\mathbf{x})) \\ &\stackrel{(a)}{=} \prod_{m=1}^{l(\mathbf{x})} \left\{ P(x_m|x_{m-1}) \prod_{k=1}^K P_k(y_k(b_{k,m-1}, b_{k,m})|\nu_k(x_m)) \right\},\end{aligned}\quad (5.2)$$

where we let $P(x_1|x_0) = P(x_1)$ as convention. Note that $l(\mathbf{x})$ is the number of symbols in the input sequence \mathbf{x} .

5.3 JSC-VLCMD MAP decoding

Now we devise a graph based algorithm for JSC-VLCMD MAP decoding. Combining (5.1) and (5.2), we have

$$\hat{\mathbf{x}} = \operatorname{argmax}_{\substack{\mathbf{x} \in \mathbb{C}^* \\ \nu(\mathbf{x}) = \mathcal{N}}} \sum_{m=1}^{l(\mathbf{x})} \left\{ \log P(x_m|x_{m-1}) + \sum_{k=1}^K \log P_k(y_k(b_{k,m-1}, b_{k,m})|\nu_k(x_m)) \right\}, \quad (5.3)$$

where $\mathbf{N} = (N_1, \dots, N_K)$, $\boldsymbol{\nu}(\mathbf{x}) = (|\nu_1(\mathbf{x})|, \dots, |\nu_K(\mathbf{x})|)$. The additivity of (5.3) breaks the MAP estimation problem into the following subproblems:

$$w(\mathbf{n}, a) = \max_{\substack{\mathbf{x} \in \mathbb{C}^* \\ \boldsymbol{\nu}(\mathbf{x}) = \mathbf{n} \\ x_{l(\mathbf{x})} = a}} \sum_{m=1}^{l(\mathbf{x})} \left\{ \log P(x_m | x_{m-1}) + \sum_{k=1}^K \log P_k(y_k(b_{k,m-1}, b_{k,m}) | \nu_k(x_m)) \right\}, \quad (5.4)$$

where $\mathbf{n} = (n_1, \dots, n_K)$, $1 \leq n_k \leq N_k$, $k = 1, 2, \dots, K$, $a \in \mathbb{C}$. Then, the solution of the optimization problem (5.1) is given by

$$\hat{\mathbf{x}} = \arg \max_{c \in \mathbb{C}} w(\mathbf{N}, c), \quad (5.5)$$

The solution to subproblems $w(\cdot, \cdot)$ can be expressed recursively as

$$\begin{aligned} w(\mathbf{n}, a) &= \max_{\substack{\mathbf{x} \in \mathbb{C}^* \\ \boldsymbol{\nu}(\mathbf{x}) = \mathbf{n} \\ x_{l(\mathbf{x})} = a}} \left\{ \sum_{m=1}^{l(\mathbf{x})-1} \left[\log P(x_m | x_{m-1}) + \sum_{k=1}^K \log P_k(y_k(b_{k,m-1}, b_{k,m}) | \nu_k(x_m)) \right] \right. \\ &\quad \left. + \log P(a | x_{l(\mathbf{x})-1}) + \sum_{k=1}^K \log P_k(y_k(b_{k,l(\mathbf{x})-1}, b_{k,l(\mathbf{x})}) | \nu_k(a)) \right\} \\ &= \max_{b \in \mathbb{C}} \left\{ w(\mathbf{n} - \boldsymbol{\nu}(a), b) + \log P(a | b) \right\} + \sum_{k=1}^K \log P_k(y_k(n_k - |\nu_k(a)|, n_k) | \nu_k(a)). \end{aligned} \quad (5.6)$$

The above recursion reduces the MAP estimation problem to one of finding the longest path in a weighted directed acyclic graph [29], which is given in Figure 5.2. The underlying graph G has $L \prod_{k=1}^K N_k + 1$ vertices, which forms a hyper-trellis of dimension $K + 1$, where K dimensions represent the K received bitstreams $\mathbf{y}_1, \dots, \mathbf{y}_K$, and the remaining dimension corresponds to L codecells of the central quantizer. There is also one starting node s , corresponding to the beginning of \mathbf{x} .

We use a $(K + 1)$ -dimensional vector (\mathbf{n}, x) , $1 \leq n_k \leq N_k$, $x \in \mathbb{C}$ to label a node in G . From node $(\mathbf{n} - \boldsymbol{\nu}(a), b)$ to node (\mathbf{n}, a) , $\mathbf{n} = (n_1, \dots, n_K)$, $a, b \in \mathbb{C}$, there is a

directed edge, with weight

$$\log P(a|b) + \sum_{k=1}^K \log P_k(y_k(n_k - |\nu_k(a)|, n_k) | \nu_k(a)). \quad (5.7)$$

From the starting node s to each node $(\nu(a), a)$, there is an edge whose weight is

$$\log P(a) + \sum_{k=1}^K \log P_k(y_k(0, |\nu_k(a)|) | \nu_k(a)). \quad (5.8)$$

In graph G , the solution of the subproblem $w(\mathbf{n}, a)$ is the weight of the longest path from the starting node s to node (\mathbf{n}, a) , which can be calculated recursively using dynamic programming. The MAP decoding problem is then converted into finding the longest path in graph G from the starting node s to nodes (\mathbf{N}, c) , $c \in \mathbb{C}$.

To analyze the algorithm complexity we note that the dynamic programming algorithm proceeds from the starting node s to the nodes (\mathbf{N}, c) , through all $L \prod_{k=1}^K N_k$ nodes in G . The quantities $\log P_k(y_k(n_k - |\nu_k(a)|, n_k) | \nu_k(a))$, $\log P(a)$ and $\log P(a|b)$ can be precomputed and stored in lookup tables so that they will be available to the dynamic programming algorithm in $O(1)$ time. By tracing back step by step to the starting node s , the MDQ decoder can reconstruct the input sequence \mathbf{x} to $\hat{\mathbf{x}}$, the optimal result as defined in (5.1). Hence (5.7) and (5.8) can be computed in $O(K)$ time. Therefore the value of $w(\mathbf{n}, a)$ can be evaluated in $O(L + K)$ time, according to (5.6). Thus the total time complexity of this algorithm is $O(L(L + K) \prod_{k=1}^K N_k)$. To reconstruct the input sequence, the selection in (5.6) should be recorded at each node, which results in a space complexity of $O(L \prod_{k=1}^K N_k)$.

The fixed-rate joint source-channel MDQ decoding presented in Chapter 4 can be regarded as a special case of our proposed algorithm, where all codewords for a side description have the same length, say, B_k , $k = 1, \dots, K$, $B_k = N_k/N$. Then the parsing of \mathbf{y}_k can be uniquely determined, namely

$$y_k(0, B_k], y_k(B_k, 2B_k], \dots, y_k(N_k - B_k, N_k].$$

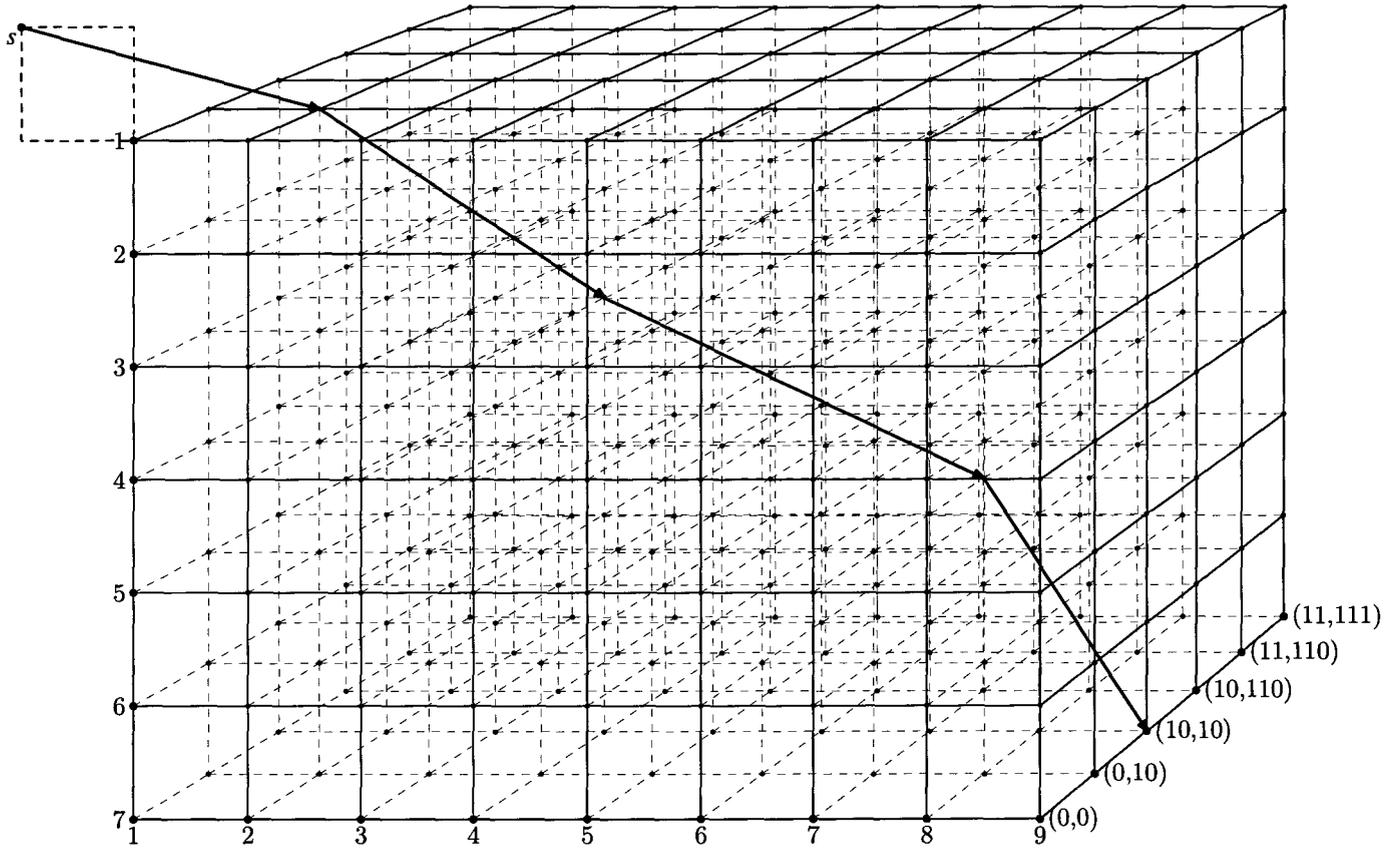


Figure 5.2: Graph G constructed for the joint source-channel MDQ decoding ($K = 2$, $L = 6$, $\mathcal{C}_1 = \{0, 10, 11\}$, $\mathcal{C}_2 = \{0, 10, 110, 111\}$, $N_1 = 7$ and $N_2 = 9$).

Therefore, in graph G only nodes (mB_1, \dots, mB_K, a) , $1 \leq m \leq M, a \in \mathbb{C}$ are achievable nodes, in the sense that $w(mB_1, \dots, mB_K, a) \neq -\infty$. Hence, other nodes can be removed from the graph G without losing the optimality. The remaining graph is a two dimensional graph as given in Figure 4.2.

5.4 Complexity Reduction of JSC-VLCMD MAP Decoding Algorithm

In Section 4.4, the computational complexity of the JSC-MD MAP decoding algorithm was reduced using a fast matrix search algorithm. The same technique can be applied to JSC-VLCMD MAP decoding algorithm.

To see this, we write (5.6) into another form as follows

$$w(\mathbf{n} + \boldsymbol{\nu}(a), a) = \max_{b \in \mathbb{C}} \left\{ w(\mathbf{n}, b) + \log P(a|b) \right\} + \sum_{k=1}^K \log P_k(y_k(n_k, n_k + |\nu_k(a)|) | \nu_k(a)). \quad (5.9)$$

For each \mathbf{n} , Define an $L \times L$ matrix $A_{\mathbf{n}}$ such that

$$A_{\mathbf{n}}(a, b) = w(\mathbf{n}, b) + \log P(a|b) + \sum_{k=1}^K \log P_k(y_k(n_k, n_k + |\nu_k(a)|) | \nu_k(a)). \quad (5.10)$$

Then (5.6) is equivalent to finding the row maxima of $A_{\mathbf{n}}$. Substituting $A_{\mathbf{n}}$ in (5.10) for A in (4.13), we have

$$\log P(a|b') + \log P(a'|b) \leq \log P(a|b) + \log P(a'|b'), \quad a < a', b < b', \quad (5.11)$$

which is the same condition as (4.16). Therefore, we can apply the fast matrix search algorithm to reduce the computational complexity of the JSC-VLCMD MAP decoding algorithm to $O((L + K) \prod_{k=1}^K N_k)$ for Gaussian Markov sequences.

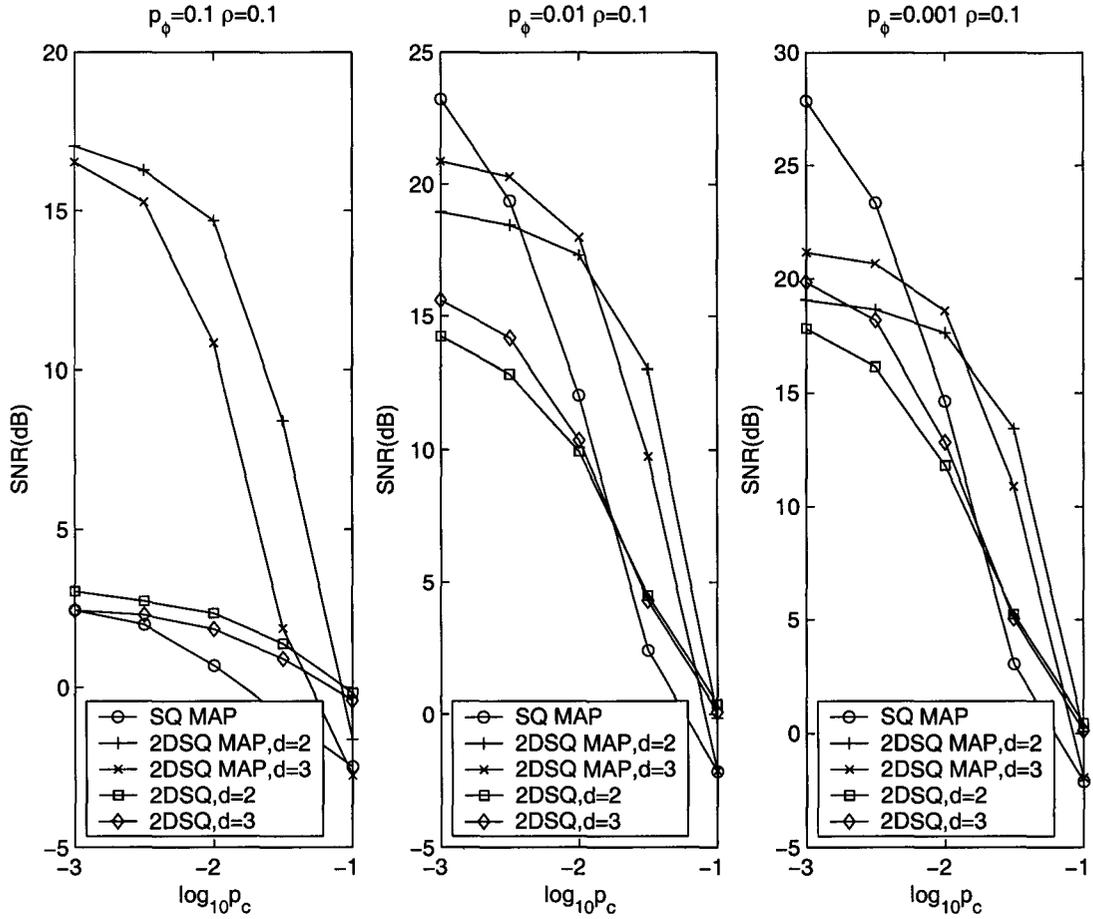


Figure 5.3: SNR performances of different MDQ decoders ($\rho = 0.1$), where d is the number of diagonals in the 2DSQ index assignment matrix.

5.5 Simulation Results

We implemented the proposed JSC-VLCMD MAP decoding algorithm and tested it on three first-order, zero-mean, unit-variance Gaussian Markov sequences with the correlation coefficient ρ being 0.1, 0.5 and 0.9 respectively. Two two-description scalar quantizers used in Chapter 4 were also used in our experiments, which are uniform and have the index assignment matrices shown in Figure 2.3. One of them has $L = 15$ central codecells, and the other $L = 21$ codecells. For both 2DSQ's, the two side quantizers each has $L_1 = L_2 = 8$ codecells.

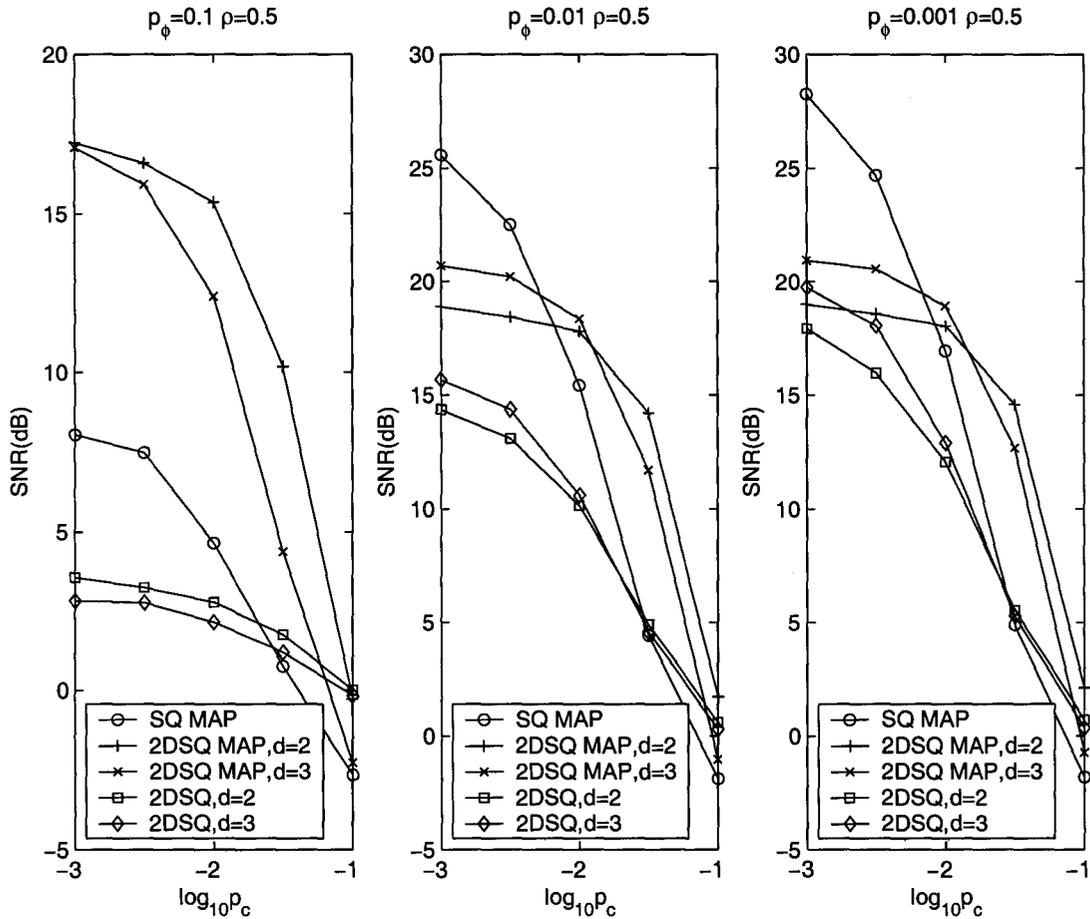


Figure 5.4: SNR performances of different MDQ decoders ($\rho = 0.5$), where d is the number of diagonals in the 2DSQ index assignment matrix.

For each description k , $k = 1, 2$, Huffman codes are generated according to the distribution of the side quantization codecell. The encoded bitstreams $\nu_k(\mathbf{x})$ are then transmitted over two error-and-erasure channels with erasure probability p_ϕ and inversion probability p_c varying. The new MDQ decoding algorithm is compared with two other schemes: 1) MAP decoding of single description scalar quantization of 64 codecells, and 2) conventional hard-decision MDQ decoding of above two 2DSQ schemes. The competing scalar quantizer (SQ) is uniform and coded by a VLC so that it matches the 2DSQ's in rate and codecell structure. The system performance measure is the signal-to-noise ratio.

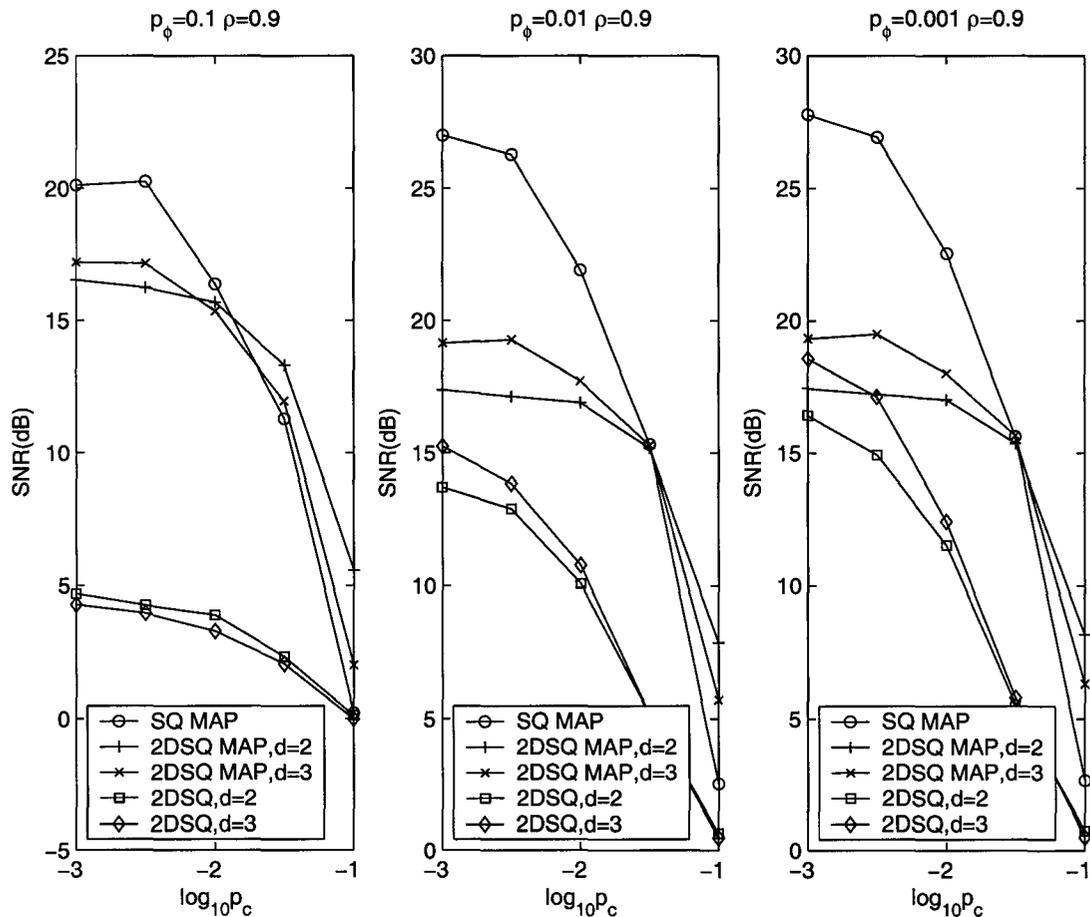


Figure 5.5: SNR performances of different MDQ decoders ($\rho = 0.9$), where d is the number of diagonals in the 2DSQ index assignment matrix.

The simulation results are plotted in Figure 5.3, Figure 5.4 and Figure 5.5. Over all values of ρ , p_c and p_ϕ , the JSC-VLCMD MAP decoder outperforms the conventional hard-decision MDQ decoder, regardless the level of correlation between the two side descriptions. Not surprisingly, the performance gap between the two approaches increases as the amount of memory in the Markov source (ρ) increases. This is because the hard-decision MDQ decoder cannot benefit from the residual source redundancy in \mathbf{x} . The gap also increases as the erasure error probability p_ϕ increases, indicating that the proposed VLCMD MAP decoder can make a better use of inter-description correlation. Also, as expected, the MAP SQ scheme achieves higher SNR than the

VLCMD MAP scheme when the channel quality is very good, but the former loses to the latter as the channel condition deteriorates. This is when the redundancy of MDQ starts to pay off. More interestingly, we notice that joint source-channel MAP decoding of MDQ is advantageous even when the source memory is weak (see the curves for $\rho = 0.1$).

5.6 Conclusions

We proposed a framework for optimal (in MAP sense) joint source-channel decoding of Markov sequences compressed by entropy coded MDQ. This framework allows both inter-description and intra-description correlations to be exploited for correcting bit errors as well as erasure errors. It is suitable for lossy communications involving low-power inexpensive encoders.

The new MDQ decoding technique unifies the treatments of different subsets of descriptions available at the decoder, overcoming the difficulty of having a large number of side decoders that hinders the design of a good hard-decision MDQ decoder. Moreover, our joint source-channel decoder considers simultaneously the processes of decoding and merging of multiple descriptions, thus evades the difficulty in merging two desynchronized descriptions, which hard decision MDQ decoders have to face.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

This thesis advances the state of the art of joint source channel coding by solving some open problems in asymmetric signal communication systems with resource constraints as described in the introduction. In brief, a resource-deprived encoder resorts to fixed rate or simple variable rate quantizers to encode the source sequence without explicit channel codes. The suboptimality of the source encoder leaves significant amount of redundancy of various forms in the encoded sequence. It is the responsibility of the decoder to exploit these redundancies to overcome channel impairments and estimate the input source signal. This scenario is common in many applications of wireless networks and sensor networks.

First, we opened up an inquiry into the impact of index assignment of source codewords on JSC MAP decoding, and found some interesting results that have practical significance. For fixed rate quantizers optimal index assignment for JSC MAP decoding can be formulated as a quadratic assignment problem. Although QAP is generally NP-hard, for some special yet important cases, the optimal index assignment can be found analytically. For Gaussian Markov sources of high correlation and for the Hamming distortion metric, we show that optimal index assignment corresponds to hypercube antibandwidth. The design method of this index assignment

and the optimality proof are also given. For more general sources and distortion metrics, approximate algorithms such as simulated annealing can be employed to search for good index assignments.

Second, we expanded the application domain of JSC decoding to include multiple descriptions, and demonstrated the benefits of combining two popular techniques for signal communication and estimation in lossy networks. We developed the JSC-MD MAP decoding algorithms for both fixed-length and variable-length codes. The computational complexity of both algorithms can be further reduced if the source is Gaussian Markov. We also extended the classical forward backward algorithm to multiple descriptions and developed a JSC-MD MMSE decoding algorithm.

We also introduced the new concept of resource-scalable network communication, and proposed the JSC-MD framework as a unified solution to suit decoders of different resource levels. The defining feature of such a flexible system is that different JSC-MD decoding algorithms can take the same bit streams as input, and obtain decoded results of different accuracy according to the available bandwidth and computational capability of an individual decoder.

6.2 Future Works

In Chapter 3, we investigated the problem of optimal index assignment design for fixed rate quantization with respect to JSC MAP decoding. A natural question to ask is what is the optimal index assignment for *variable rate* quantization and JSC MAP decoding. In this section, we first show why this problem is difficult. Intuitions borrowed from the resynchronization properties of variable-length codes are then provided. Examples are employed to show that the index assignment optimization can improve the decoding performance considerably.

If the source is encoded by variable-length codes, the problem of optimal index assignment with respect to joint source-channel MAP decoding becomes much more intricate. This is because the MAP optimization has to search through all possible

N	Code	Code in Binary	MEPL
5	Best	011,10,00,11,010	0.5445
	Huffman	011,10,11,01,000	0.6182
	Worst	100,00,01,11,101	0.9803
6	Best	1101,111,00,10,01,1100	0.4992
	Huffman	1100,111,01,10,00,1101	0.5363
	Worst	1010,100,00,01,11,1011	1.0825
7	Best	0101,110,011,00,10,111,0100	0.4420
	Huffman	0101,110,00,10,111,011,0100	0.5791
	Worst	0000,111,110,10,01,001,0001	1.3876
8	Best	01001,0101,111,00,10,011,110,01000	0.4488
	Huffman	11111,1110,101,00,01,110,100,11110	0.6929
	Worst	11111,1110,110,10,01,001,000,11110	1.4857
8	Best	1001,1010,111,00,01,110,1011,1000	0.3810
	Huffman	1000,1110,101,00,01,110,1111,1001	0.6900
	Worst	1111,1110,110,10,01,001,0001,0000	1.6220
9	Best	1000,0111,101,110,00,111,010,1001,0110	0.3728
	Huffman	0001,1011,001,111,01,110,100,1010,0000	0.7076
	Worst	0111,1111,101,100,00,010,110,0110,1110	1.1854

Table 6.1: MEPL of various VLC index assignments for Gaussian Markov source of correlation coefficient $\rho = 0.9$ when $p_c = 0.01$ ($N = 5, 6, 7, 8, 9$)

parsings of the received binary sequence. Note that these parsings could have different phases. As a result, channel errors can easily cause loss of synchronization, making the index assignment problem very difficult to analyze.

The synchronization recovery ability of VLC has been extensively studied in literature for hard-decision decoding [82, 83, 84, 85, 86]. In [82] and [83], Maxted and Monaco provided an analytical solution to calculate the synchronization resuming ability for a given VLC. This work is further extended in [85] by Swaszek and DiCiccio. In [86], Zhou and Zhang introduced a quick code construction algorithm which can find codes with strong synchronization recovery ability with two intuitions: 1) *Neighboring relation*: the codes constructed by the algorithm have as many neigh-

boring pairs as possible which are pairs of codewords of the same length at Hamming distance one. 2) *Suffix relation*: the codes have as many pairs as possible of codewords of different length such that the shorter codeword is a suffix of the longer codeword. These properties are effective in improving the resynchronization ability for Huffman (hard-decision) decoding because 1) When a single bit error occurs, the probability that the corrupted codeword is decoded into another existing codeword with the same length is relatively high, which will not cause loss of synchronization, and 2) If the synchronization has been lost, some codewords may be divided into two or more parts. If the last part happens to be a shorter codeword, the synchronization resumes.

The problem of VLC index assignment for joint source-channel MAP decoding is formulated as follows. Given an information source with N symbols, let the length of codeword c_i be $l_i, i = 1, 2, \dots, N$. The codeword length l_i can be found by any entropy encoders such as Huffman encoding algorithm[87]. The objective is to find a VLC that satisfies these codeword length constraints and emits the best performance with respect to joint source-channel MAP decoding. A meaningful performance metric, also widely used in recent literature, is the Levenshtein distance [88] between the decoding result and the original sequence. This is to find the minimum number of insertions, deletions and substitutions of symbols to transform one sequence into the other. In case of channel errors, the decoded sequence may diverge from the original sequence around the flipped bits. This diverged path (after alignment) is called the *divergent segment* [32]. The average length (in symbols) of the divergent segment caused by single-bit errors is called the mean error propagation length (MEPL).

As discussed in Chapter 3, a good index assignment for joint source-channel MAP decoding should increase the “distance” between two likely sequences. In other words, a sequence obtained by flipping one bit of a likely sequence should be an unlikely one. Note that the symbols in a divergent segment may not obey the Markov model. Therefore, the longer the divergent segment, the higher the chance that the flipped sequence is an unlikely sequence. Hence, the two properties introduced in [86] are

generally not good. One difficulty here is how to involve the transition probabilities into this problem except for some special cases. For example, for a Gaussian Markov source of high correlation, the neighboring and suffix relations between codewords that are close in the code space are very unfavorable.

The above intuitions are supported by experimental data. In Table 6.1, the MEPL's [86] of different index assignments for the Gaussian Markov source of correlation coefficients $\rho = 0.9$ are listed. The best and worst index assignments, found by exhaustive simulation, are compared with the Huffman code. To verify the above intuition, please see the worst index assignments: there are a lot of suffix and neighboring relations between adjacent codewords. The MEPL's of the worst index assignments can be as large as three to four times those of the best ones, which also suggests the great potential of this problem.

Appendix A

Proof of the Bandwidth and Antibandwidth of the Hypercube

A.1 Overview

In Chapter 3 we show that the problem of optimal index assignment for Gaussian Markov sources of high correlation and for Hamming distortion metric is equivalent to the antibandwidth problem of the hypercube. In this appendix, we will discuss the bandwidth and antibandwidth formulae for the hypercube and present their proofs.

In the field of graph labeling [89, 90], the problem of graph bandwidth has been extensively studied [72, 91, 71] and has found many applications such as parallel computations, VLSI circuit design, etc. First, we restate the definitions of vertex numbering and graph bandwidth, most of which are adopted from Harper's book [73]. A *numbering* of a vertex set V is any function $\eta : V \rightarrow \{1, 2, \dots, |V|\}$, which is one-to-one (and therefore onto). A numbering η uniquely determines a total order, \leq_η , on V as follows: $u \leq_\eta v$ if $\eta(u) < \eta(v)$ or $\eta(u) = \eta(v)$. Conversely, a total order defined on V uniquely determines a numbering of the graph.

The *bandwidth* of a numbering η of a graph $G = (V, E)$ is the maximum difference

$$bw(\eta) = \max_{\{u,v\} \in E} |\eta(u) - \eta(v)|. \quad (\text{A.1})$$

The *bandwidth* of a graph G is the minimum bandwidth over all numberings, η , of G , i.e.

$$bw(G) = \min_{\eta} bw(\eta). \quad (\text{A.2})$$

Another vertex numbering problem related to graph bandwidth is what we call the antibandwidth problem. It is posed by reversing the objective of vertex numbering in that we now want to maximize the minimum distance between any adjacent pair of vertices. The antibandwidth problem of a graph $G = (V, E)$ is defined as

$$abw(G) = \max_{\eta} \min_{\{u,v\} \in E} |\eta(u) - \eta(v)|. \quad (\text{A.3})$$

The antibandwidth problem was previously studied by Leung et al. in [92]. In [68], the graph antibandwidth problem was investigated by Miller and Pritikin under the name of *separation number*. Some antibandwidth related problems were also studied, such as the edge-bandwidth [93] and the cyclic antibandwidth [69].

For hypercubes, the bandwidth and antibandwidth problems are also well studied. In [52], Harper proposes a class of bandwidth achieving numberings for the hypercube and presents the exact bandwidth value without a proof. One such bandwidth achieving numbering is the Hales numbering defined in [73]. In Section A.2, we will highlight the recursive structure of the Hales numbering, and propose a simple proof of the bandwidth formula in Section A.3 based on this property.

In [93], a tight bound of the antibandwidth of the hypercube is given. Raspaud et al. further approximate the antibandwidth value of the hypercube up to the third-order term in [69]. However, in his classical 1966 paper [52], Harper has presented the antibandwidth achieving numbering for the hypercube, which is a variation of the Hales numbering. With the recursive structure of the Hales numbering, we determine in Section A.4 the exact value of the antibandwidth for the hypercube, which is a new result in the literature. In Section A.5, we will further explore the recursive structure of the adjacency matrix of the Hales numbered hypercube. This structure has applications in the index assignment design for joint source-channel decoding,

which is discussed in Chapter 3.

A.2 Recursive Structure of the Hales Numbering

The graph of the n -dimensional cube, Q_n , has vertex set $\{0, 1\}^n$, the n -fold Cartesian product of $\{0, 1\}$. Thus $|V_{Q_n}| = 2^n$. Q_n has an edge between two vertices (n -tuples of 0s and 1s) if they differ in exactly one entry.

The *Hales order* [73], \leq_H , on V_{Q_n} , is defined by $u \leq_H v$ if

1. $w(u) < w(v)$, or
2. $w(u) = w(v)$ and u is greater than or equal to v in lexicographic order relative to the right-to-left order of the coordinates,

where $w(\cdot)$ is the Hamming weight of a vertex of Q_n . This total order determines a numbering, $H_n : V_{Q_n} \rightarrow \{1, 2, \dots, 2^n\}$, which is called *Hales numbering*. For $n = 4$, for instance, the Hales order is $0000 < 0001 < 0010 < 0100 < 1000 < 0011 < 0101 < 1001 < 0110 < 1010 < 1100 < 0111 < 1011 < 1101 < 1110 < 1111$.

The Hales numbering has a nice recursive structure as follows:

Theorem 1. *We define a $2^n \times n$ $(0, 1)$ -matrix S_n as*

$$S_n = \begin{bmatrix} A_0^{(n)} \\ A_1^{(n)} \\ \vdots \\ A_n^{(n)} \end{bmatrix}, \quad (\text{A.4})$$

where $A_k^{(n)}$, $0 \leq k \leq n$, is an $\binom{n}{k} \times n$ $(0, 1)$ -matrix satisfying the following recursive formula

$$A_k^{(n)} = \begin{bmatrix} A_{k-1}^{(n-1)} & \mathbf{1} \\ A_k^{(n-1)} & \mathbf{0} \end{bmatrix}, \quad 1 \leq k \leq n-1, \quad (\text{A.5})$$

where $\mathbf{0}$ and $\mathbf{1}$ are column vectors containing only 0s and 1s respectively. As the base case, we have $A_0^{(n)} = \mathbf{0}^T$ and $A_n^{(n)} = \mathbf{1}^T$. Then the row vectors of S_n , from top to bottom, are all vertices of Q_n in the increasing Hales order.

Proof 1. By definition, it is sufficient to show that the row vectors of $A_k^{(n)}$, $0 \leq k \leq n$, are all distinct vectors with Hamming weight k , which are sorted, from top to bottom, in the decreasing lexicographic order relative to the right-to-left order of the coordinates.

We prove by induction on n . The above assertion is trivially true for $n = 1$. Assume the assertion holds for $n - 1 \geq 1$. Now for n , $A_0^{(n)}$ is a vector of Hamming weight 0 and $A_n^{(n)}$ a vector of Hamming weight n , so the assertion trivially holds. For $1 \leq k \leq n - 1$, the first $\binom{n-1}{k-1}$ vectors of $A_k^{(n)}$ are all distinct and have Hamming weight k by the induction assumption that all row vectors in $A_{k-1}^{(n-1)}$ are distinct and have Hamming weight $k - 1$. Further, these vectors are in the decreasing lexicographic order because they share the same rightmost bit and all vectors in $A_k^{(n-1)}$ are sorted. By the same argument the next $\binom{n-1}{k}$ vectors of $A_k^{(n)}$ are distinct, of Hamming weight k , and sorted in the decreasing lexicographic order as well. Combining the above facts and (A.5) concludes that the row vectors of $A_k^{(n)}$ are distinct, of Hamming weight k , and in the decreasing lexicographic order. \square

A.3 Proof of the Hypercube Bandwidth Formula

In [52] and [73, Corollary 4.3], Harper shows that the Hales numbering minimizes the bandwidth of the n -cube, i.e.

$$bw(H_n) = bw(Q_n). \quad (\text{A.6})$$

He also gives the exact bandwidth value of the hypercube as follows.

Theorem 2 (Harper, [73], Corollary 4.4). *For the n -cube Q_n , we have*

$$bw(Q_n) = \sum_{m=0}^{n-1} \binom{m}{\lfloor \frac{m}{2} \rfloor}. \quad (\text{A.7})$$

Although the above result has been known for forty years, no proof seemed to appear in the literature. In his recent book [73], Harper noted that the proof of the bandwidth formula of hypercube (Theorem 2) “is surprisingly difficult”. In the following we present a quite simple proof, which is based on the recursive structure of Hales numbering as given in Theorem 1.

Proof 2. For any two vertices $u, v \in V_{Q_n}$, the condition $\{u, v\} \in E_{Q_n}$ implies that $|w(u) - w(v)| = 1$. Without loss of generality, let $w(v) = w(u) + 1$. If u is a row vector of $A_k^{(n)}$, denoted by $u \in A_k^{(n)}$, $0 \leq k \leq n-1$, then v must be a row vector of $A_{k+1}^{(n)}$, i.e. $v \in A_{k+1}^{(n)}$. If we define

$$d_k^{(n)} = \max_{\{u,v\} \in E_{Q_n}, u \in A_k^{(n)}, v \in A_{k+1}^{(n)}} |H_n(u) - H_n(v)|, \quad (\text{A.8})$$

for $0 \leq k \leq n-1$, then the bandwidth of the Hales numbering becomes

$$bw(H_n) = \max_{0 \leq k \leq n-1} d_k^{(n)}. \quad (\text{A.9})$$

It is trivial that $d_0^{(n)} = d_{n-1}^{(n)} = n$. For $1 \leq k \leq n-2$, according to (A.5), $u \in A_k^{(n)}$ indicates two possibilities: $u \in [A_{k-1}^{(n-1)} \ 1]$ and $u \in [A_k^{(n-1)} \ 0]$. We define

$$d_{k,b}^{(n)} = \max_{\{u,v\} \in E_{Q_n}, u \in [A_{k-b}^{(n-1)} \ b], v \in A_{k+1}^{(n)}} |H_n(u) - H_n(v)|, \quad (\text{A.10})$$

for $1 \leq k \leq n-2$ and $b \in \{0, 1\}$, then it is obvious

$$d_k^{(n)} = \max \{d_{k,0}^{(n)}, d_{k,1}^{(n)}\}. \quad (\text{A.11})$$

Now we derive a recursive formula for $d_{k,1}^{(n)}$, $1 \leq k \leq n-2$. Consider $\{u, v\} \in E_{Q_n}$

with $u \in [A_{k-1}^{(n-1)} \mathbf{1}]$ and $v \in A_{k+1}^{(n)}$. Using the recursive formula of (A.5), we can expand $A_{k+1}^{(n)}$ as follows

$$A_{k+1}^{(n)} = \begin{bmatrix} A_k^{(n-1)} & \mathbf{1} \\ A_{k+1}^{(n-1)} & \mathbf{0} \end{bmatrix}. \quad (\text{A.12})$$

It is easy to see that v must be a row vector of $[A_k^{(n-1)} \mathbf{1}]$. Moreover, $u_{1:n-1} \in A_{k-1}^{(n-1)}$, $v_{1:n-1} \in A_k^{(n-1)}$ and $\{u_{1:n-1}, v_{1:n-1}\}$ is an edge in the $(n-1)$ -dimensional hypercube, where $u_{1:n-1}$ denotes the vector formed by the leftmost $n-1$ bits of u . Since by Theorem 1, the rows of S_n , respectively S_{n-1} , from top to bottom, are the vertices of Q_n , respectively, Q_{n-1} , in increasing Hales order and because the number of rows in $[A_k^{(n-1)} \mathbf{0}]$ is $\binom{n-1}{k}$, we obtain

$$|H_n(u) - H_n(v)| = \binom{n-1}{k} + |H_{n-1}(u_{1:n-1}) - H_{n-1}(v_{1:n-1})|. \quad (\text{A.13})$$

Since there is a one-to-one correspondence between the edges $\{u, v\} \in E_{Q_n}$ with $u \in [A_{k-1}^{(n-1)} \mathbf{1}]$, $v \in [A_k^{(n-1)} \mathbf{1}]$ and the edges $\{u_{1:n-1}, v_{1:n-1}\} \in E_{Q_{n-1}}$ with $u_{1:n-1} \in A_{k-1}^{(n-1)}$, $v_{1:n-1} \in A_k^{(n-1)}$, it follows further that

$$d_{k,1}^{(n)} = \binom{n-1}{k} + d_{k-1}^{(n-1)}, \quad 1 \leq k \leq n-2. \quad (\text{A.14})$$

In order to derive a recursion for $d_{k,0}^{(n)}$, $1 \leq k \leq n-2$, consider now $\{u, v\} \in E_{Q_n}$ with $u \in [A_k^{(n-1)} \mathbf{0}]$ and $v \in A_{k+1}^{(n)}$. Then v could be either in $[A_k^{(n-1)} \mathbf{1}]$ or in $[A_{k+1}^{(n-1)} \mathbf{0}]$. But since the rows of S_n are in increasing Hales order, from top to bottom, it suffices to consider only $v \in [A_{k+1}^{(n-1)} \mathbf{0}]$ in the maximization of (A.10) for $d_{k,0}^{(n)}$. Further, similarly to the discussion for $d_{k,1}^{(n)}$, we obtain

$$d_{k,0}^{(n)} = \binom{n-1}{k} + d_k^{(n-1)}, \quad 1 \leq k \leq n-2. \quad (\text{A.15})$$

Making further the convention that $d_{-1}^{(n)} = 0$ and $d_n^{(n)} = 0$ for any $n \geq 1$, relations

(A.11), (A.14),(A.15) and the fact $d_0^{(n)} = d_{n-1}^{(n)} = n$ imply that

$$d_k^{(n)} = \binom{n-1}{k} + \max\{d_{k-1}^{(n-1)}, d_k^{(n-1)}\}, \quad 0 \leq k \leq n-1. \quad (\text{A.16})$$

In order to complete the proof of the theorem we next prove the following assertion.

Assertion. For any $n \geq 1$, we have

$$d_k^{(n)} \leq \sum_{m=0}^{n-1} \binom{m}{\lfloor \frac{m}{2} \rfloor}, \quad 0 \leq k \leq n-1. \quad (\text{A.17})$$

Moreover, (A.17) holds with equality if $k = \lfloor \frac{n}{2} \rfloor$.

Proof of Assertion. We prove the assertion by induction on n . The base case $n = 1$ follows immediately since $d_0^{(1)} = 1 = \binom{0}{0}$ and $0 = \lfloor \frac{1-1}{2} \rfloor$. Assume now that the assertion is true for $n-1 \geq 1$. Hence,

$$d_k^{(n-1)} \leq \sum_{m=0}^{n-2} \binom{m}{\lfloor \frac{m}{2} \rfloor}, \quad 0 \leq k \leq n-2, \quad (\text{A.18})$$

with equality when $k = \lfloor \frac{n-1}{2} \rfloor$. Consider $k, 1 \leq k \leq n-1$. Using (A.16) together with the induction hypothesis we obtain that

$$\begin{aligned} d_k^{(n)} &= \binom{n-1}{k} + \max\{d_{k-1}^{(n-1)}, d_k^{(n-1)}\} \\ &\leq \binom{n-1}{\lfloor \frac{n}{2} \rfloor} + \sum_{m=0}^{n-2} \binom{m}{\lfloor \frac{m}{2} \rfloor} \\ &= \sum_{m=0}^{n-1} \binom{m}{\lfloor \frac{m}{2} \rfloor}, \end{aligned} \quad (\text{A.19})$$

which proves inequality (A.17). Assume now that $k = \lfloor \frac{n}{2} \rfloor$. Then when n is even, $k-1 = \lfloor \frac{n}{2} \rfloor - 1 = \lfloor \frac{n-1}{2} \rfloor$, and $d_{k-1}^{(n-1)}$ achieves equality in (A.18). When n is odd, $k = \lfloor \frac{n}{2} \rfloor = \lfloor \frac{n-1}{2} \rfloor$, hence $d_k^{(n-1)}$ also achieves equality in (A.18). Both cases lead to equality in (A.19), thus completing the proof of the assertion. Now the conclusion of Theorem 2 follows immediately. \square

A.4 The Antibandwidth Formula of Hypercubes

According to Harper [52], an antibandwidth achieving numbering for the hypercube can be obtained as follows. First number the vertices with even Hamming weights in increasing Hales order, and then number the vertices with odd Hamming weights, also in the increasing Hales order. Let us denote this numbering by $\tilde{H}_n(\cdot)$. An example of such a numbering for $n = 4$ is $0000 < 0011 < 0101 < 1001 < 0110 < 1010 < 1100 < 1111 < 0001 < 0010 < 0100 < 1000 < 0111 < 1011 < 1101 < 1110$. In this section, we provide a closed-form formula for the solution of the antibandwidth problem on n -cubes, which is a new result.

Theorem 3. *For the numbering \tilde{H}_n , we have*

$$abw(\tilde{H}_n) = 2^{n-1} - \sum_{m=0}^{n-2} \binom{m}{\lfloor \frac{m}{2} \rfloor}. \quad (\text{A.20})$$

Proof 3. Let \tilde{S}_n be the $2^n \times n$ $(0, 1)$ -matrix whose rows from top to bottom are the vertices of Q_n in increasing order of \tilde{H}_n . Using the recursive structure in (A.5), we obtain

$$\tilde{S}_n = \begin{bmatrix} A_0^{(n)} \\ A_2^{(n)} \\ \vdots \\ A_{2\lfloor \frac{n}{2} \rfloor}^{(n)} \\ A_1^{(n)} \\ A_3^{(n)} \\ \vdots \\ A_{2\lfloor \frac{n-1}{2} \rfloor + 1}^{(n)} \end{bmatrix} = \begin{bmatrix} A_0^{(n-1)} & \mathbf{0} \\ A_1^{(n-1)} & \mathbf{1} \\ \vdots & \\ A_{n-1}^{(n-1)} & b \\ A_0^{(n-1)} & \mathbf{1} \\ A_1^{(n-1)} & \mathbf{0} \\ \vdots & \\ A_{n-1}^{(n-1)} & \bar{b} \end{bmatrix} = \begin{bmatrix} S_{n-1} & \mathbf{B} \\ S_{n-1} & \bar{\mathbf{B}} \end{bmatrix}, \quad (\text{A.21})$$

where $b = 0$ if n is odd and $b = 1$ if n is even. We also use the relations $A_0^{(n)} = \mathbf{0}^T = [A_0^{(n-1)} \ \mathbf{0}]$ and $A_n^{(n)} = \mathbf{1}^T = [A_{n-1}^{(n-1)} \ \mathbf{1}]$. Note that \mathbf{B} is a column vector of 2^{n-1} elements whose definition is straightforward and omitted here. Two observations regarding \tilde{S}_n are crucial for the further development, namely

- O1) the leftmost $n - 1$ bits of the top (bottom) 2^{n-1} vertices form a Hales numbered hypercube in $n - 1$ dimensions, i.e. S_{n-1} (see (A.4));
- O2) if u is the vertex on some row $i, i \leq 2^{n-1}$, then the vertex on row $i + 2^{n-1}$ is u' obtained by flipping the last bit of u .

Consider now an edge $\{u, v\} \in E_{Q_n}$. Clearly the Hamming weights $w(u)$ and $w(v)$ have different parities. Assume without restricting the generality that $w(u)$ is odd and $w(v)$ is even, hence u is in the top half and v is in the bottom half of \tilde{S}_n . Then according to observation O2, if $v = u'$ we have

$$\tilde{H}_n(v) - \tilde{H}_n(u) = 2^{n-1} = 2^{n-1} + H_{n-1}(v_{1:n-1}) - H_{n-1}(u_{1:n-1}), \quad (\text{A.22})$$

where the last equality is due to the fact that $v_{1:n-1} = u_{1:n-1}$. On the other hand, if $v \neq u'$, then

$$\begin{aligned} \tilde{H}_n(v) - \tilde{H}_n(u) &= \tilde{H}_n(u') - \tilde{H}_n(u) + \tilde{H}_n(v) - \tilde{H}_n(u') \\ &= 2^{n-1} + H_{n-1}(v_{1:n-1}) - H_{n-1}(u'_{1:n-1}) \\ &= 2^{n-1} + H_{n-1}(v_{1:n-1}) - H_{n-1}(u_{1:n-1}), \end{aligned} \quad (\text{A.23})$$

where the second equality is due to O2 and O1 noting that both v and u' are in the bottom half of \tilde{S}_n . Moreover, the last equality in (A.23) follows from $u'_{1:n-1} = u_{1:n-1}$.

Finally, by combining (A.22) and (A.23), we obtain

$$\begin{aligned}
abw(Q_n) &= \min_{\{u,v\} \in E_{Q_n}} |\tilde{H}_n(v) - \tilde{H}_n(u)| \\
&= \min_{\{u,v\} \in E_{Q_n}} |2^{n-1} + H_{n-1}(u_{1:n-1}) - H_{n-1}(v_{1:n-1})| \\
&\stackrel{(a)}{=} \min_{\{u_{1:n-1}, v_{1:n-1}\} \in E_{Q_{n-1}}} \{2^{n-1}, 2^{n-1} + H_{n-1}(u_{1:n-1}) - H_{n-1}(v_{1:n-1})\} \\
&\stackrel{(b)}{=} 2^{n-1} - \max_{\{u_{1:n-1}, v_{1:n-1}\} \in E_{Q_{n-1}}} \{|H_{n-1}(v_{1:n-1}) - H_{n-1}(u_{1:n-1})|\} \\
&= 2^{n-1} - bw(Q_{n-1}) \\
&= 2^{n-1} - \sum_{m=0}^{n-2} \binom{m}{\lfloor \frac{m}{2} \rfloor},
\end{aligned} \tag{A.24}$$

where (a) is due to the fact that $2^{n-1} + H_{n-1}(u_{1:n-1}) - H_{n-1}(v_{1:n-1}) > 0$ and when $\{u, v\}$ varies over E_{Q_n} , $\{u_{1:n-1}, v_{1:n-1}\}$ varies over $E_{Q_{n-1}} \cup \{\{\nu, \nu\} | \nu \in Q_{n-1}\}$. Finally, (b) follows from the fact that the set of values $H_{n-1}(v_{1:n-1}) - H_{n-1}(u_{1:n-1})$ is symmetric relative to 0. \square

Then based on Harper's result that \tilde{H}_n is the antibandwidth achieving numbering, we have the following corollary.

Corollary 1.

$$abw(Q_n) = 2^{n-1} - \sum_{m=0}^{n-2} \binom{m}{\lfloor \frac{m}{2} \rfloor}. \tag{A.25}$$

A.5 Recursive Structure of the Adjacency Matrices

It is well known that the bandwidth of a numbering η of a graph G is equal to the bandwidth of the adjacency matrix of G numbered by η . However, in some applications such as the index assignment optimization for joint source-channel decoding [67], the bandwidth (antibandwidth) is not the only factor that affects the performance of the communication system. The distribution of 1's in the adjacency matrix also

plays a role in the system. This motivates us to study the structure of the adjacency matrix of the hypercube numbered by the Hales numbering H_n and its variant \tilde{H}_n .

Given a graph $G = (V, E)$, for two sorted vertex subsets $V_1 \subseteq V$ and $V_2 \subseteq V$ the *adjacency matrix* of V_1 and V_2 is a $|V_1| \times |V_2|$ matrix such that for any $u \in V_1$ and $v \in V_2$, the value in position (u, v) is 1 if u and v are adjacent and 0 otherwise. The *adjacency matrix of a graph numbered by a numbering η* is the adjacency matrix of V and V both sorted by η .

Let M_n be the $2^n \times 2^n$ adjacency matrix of Q_n numbered by H_n . Recall from Theorem 1 that matrix S_n has as rows all vertices of Q_n sorted by H_n . Consider the submatrices $A_k^{(n)}$ of dimension $\binom{n}{k} \times n$ and $A_{k'}^{(n)}$ of dimension $\binom{n}{k'} \times n$ in S_n , and let the $\binom{n}{k} \times \binom{n}{k'}$ matrix $M_{k,k'}^{(n)}$ be the adjacency matrix of node sets represented by $A_k^{(n)}$ and $A_{k'}^{(n)}$. Then matrices $M_{k,k'}^{(n)}$, $0 \leq k, k' \leq n$, form the $2^n \times 2^n$ adjacency matrix of the Hales numbered hypercube: $M_n = [M_{k,k'}^{(n)}]$. Obviously, $M_{k,k'}^{(n)}$ is an all-zero matrix if $|k - k'| \neq 1$. Therefore, we have

$$M_n = \begin{bmatrix} \mathbf{0} & M_{0,1}^{(n)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ M_{1,0}^{(n)} & \mathbf{0} & M_{1,2}^{(n)} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & M_{2,1}^{(n)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & M_{n-1,n}^{(n)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & M_{n,n-1}^{(n)} & \mathbf{0} \end{bmatrix}. \quad (\text{A.26})$$

Similarly, we have the adjacency matrix of Q_n with the vertices numbered in the

order of \tilde{H}_n

$$\tilde{M}_n = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & M_{0,1}^{(n)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & M_{2,1}^{(n)} & M_{2,3}^{(n)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & M_{2\lfloor \frac{n}{2} \rfloor, 2\lfloor \frac{n-1}{2} \rfloor + 1}^{(n)} \\ M_{1,0}^{(n)} & M_{1,2}^{(n)} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & M_{3,2}^{(n)} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & M_{2\lfloor \frac{n-1}{2} \rfloor + 1, 2\lfloor \frac{n}{2} \rfloor}^{(n)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}. \quad (\text{A.27})$$

The matrices $M_{k,k+1}^{(n)}$ also have a recursive structure. To see this, rewrite (A.5) as,

$$A_k^{(n)} = \begin{bmatrix} A_{k-1}^{(n-1)} & \mathbf{1} \\ A_k^{(n-1)} & \mathbf{0} \end{bmatrix} \text{ and } A_{k+1}^{(n)} = \begin{bmatrix} A_k^{(n-1)} & \mathbf{1} \\ A_{k+1}^{(n-1)} & \mathbf{0} \end{bmatrix}, \quad 1 \leq k \leq n-1. \quad (\text{A.28})$$

Then $M_{k,k+1}^{(n)}$, the adjacency matrix between $A_k^{(n)}$ and $A_{k+1}^{(n)}$, can be divided into four submatrices. The top-left one is the adjacency matrix between $[A_{k-1}^{(n-1)} \ \mathbf{1}]$ and $[A_k^{(n-1)} \ \mathbf{1}]$, which equals to the adjacency matrix between $A_{k-1}^{(n-1)}$ and $A_k^{(n-1)}$, i.e. $M_{k-1,k}^{(n-1)}$. Similarly, the bottom-right one is $M_{k,k+1}^{(n-1)}$. Because there is no pair of Hamming distance one between $A_{k-1}^{(n-1)}$ and $A_{k+1}^{(n-1)}$, the top-right submatrix is an all-zero matrix. The bottom-left submatrix is the adjacency matrix between $[A_k^{(n-1)} \ \mathbf{0}]$ and $[A_k^{(n-1)} \ \mathbf{1}]$, which is an identity matrix $\mathbf{I}_{\binom{n-1}{k}}$ of dimension $\binom{n-1}{k}$. Namely,

$$M_{k,k+1}^{(n)} = \begin{bmatrix} M_{k-1,k}^{(n-1)} & \mathbf{0} \\ \mathbf{I}_{\binom{n-1}{k}} & M_{k,k+1}^{(n-1)} \end{bmatrix}, \quad 1 \leq k \leq n-1. \quad (\text{A.29})$$

Because $A_0^{(n)}$ is the all zero vector and $A_1^{(n)}$ contains n vectors of Hamming weight 1, we have $M_{0,1}^{(n)} = \mathbf{1}^T$.

Bibliography

- [1] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 79–423, 623–656, July, Oct. 1948.
- [2] S. B. Zahir Azami, P. Duhamel, and O. Rioul, “Joint source-channel coding: Panorama of methods,” *Proc. of CNES Workshop on Data Compression*, pp. 1232–1254, 1996.
- [3] Workshop Report, “NSF sponsored workshop on joint source-channel coding,” <http://www.ee2.caltech.edu/Faculty/effros/JSCC/index.html>, Oct. 1999.
- [4] K. Sayood, H. H. Otu, and N. Demir, “Joint source/channel coding for variable length codes,” *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 787–794, May 2000.
- [5] Y. Zhong, F. Alajaji, and L. L. Campbell, “On the joint source-channel coding error exponent for discrete memoryless systems,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1450–1468, 2006.
- [6] J. D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inf. Theory*, vol. 19, no. 7, pp. 471–480, July 1973.
- [7] A. D. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
- [8] I. F. Akyldiz, W. Su, Y. Sankarasubermanian, and E. Cayirici, “A survey on sensor networks,” *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

-
- [9] R. Puri and K. Ramchandran, "Prism: A 'reversed' multimedia coding paradigm," in *Proc. IEEE Int. Conf. on Image Processing*, Sept. 2003, pp. 617–620.
- [10] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.
- [11] M. R. Garey, D. S. Johnson, and H. S. Witsenhausen, "The complexity of the generalized Lloyd-Max problem," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 255–256, Mar. 1982.
- [12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
- [13] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.
- [14] X. Wang, X. Wu, and S. Dumitrescu, "On optimal index assignment for MAP decoding of Markov sequences," in *Proc. IEEE Inter. Symposium on Information Theory*, Seattle, Washington, July 2006, pp. 2314–2318.
- [15] X. Wang and X. Wu, "Index assignment optimization for joint source-channel MAP decoding," in *Proc. IEEE Information Theory Workshop*, Chengdu, China, Oct. 2006, pp. 308–312.
- [16] ———, "Index assignment optimization for joint source-channel MAP decoding," *IEEE Transactions on Communications*, submitted, 2008.
- [17] X. Wu, X. Wang, and J. Wang, "Joint source-channel decoding of multiple description quantized Markov sequences," in *Proc. Data Compression Conf.*, Snowbird, Utah, Mar. 2006, pp. 103–112.

- [18] X. Wu, X. Wang, and Z. Wang, "Resource-aware joint source-channel multiple description estimation and compression in networks," in *Proc. IEEE Information Theory Workshop*, Lake Tahoe, California, Sept. 2007, pp. 271–276.
- [19] —, "Resource-scalable joint source-channel MAP and MMSE estimation of multiple descriptions," *IEEE Transactions on Signal Processing*, *accepted*, 2008.
- [20] X. Wang and X. Wu, "Joint source-channel decoding of multiple description quantized and variable length coded Markov sequences," in *Proc. IEEE Inter. Conf. on Multimedia & Expo*, Toronto, Canada, July 2006, pp. 1429–1432.
- [21] X. Wang, X. Wu, and S. Dumitrescu, "Recursive structure and bandwidth of Hales-numbered hypercube," *Discrete Applied Mathematics*, *under revision*, 2008.
- [22] S. Emami and S. L. Miller, "DPCM picture transmission over noisy channels with the aid of a Markov model," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1473 – 1481, Nov. 1995.
- [23] A. W. Drake, "Observation of a Markov source through a noisy channel," Ph.D. dissertation, M.I.T., Cambridge, MA, June 1966.
- [24] J. L. Devore, "A note on the observation of a Markov source through a noisy channel," *IEEE Trans. Inf. Theory*, vol. 20, no. 6, pp. 762–764, Nov. 1974.
- [25] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 838–846, June 1991.
- [26] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov sources over discrete memoryless channels – applications to combined source-channel coding," *IEEE Trans. Inf. Theory*, vol. 40, no. 1, pp. 186–193, Jan. 1994.

- [27] K. P. Subbalakshmi and J. Vaisey, "On the joint source-channel decoding of variable-length encoded sources: The BSC case," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2052–2055, Dec. 2001.
- [28] —, "On the joint source-channel decoding of variable-length encoded sources: the additive-Markov case," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1420–1425, Sept. 2003.
- [29] X. Wu, S. Dumitrescu, and Z. Wang, "Monotonicity-based fast algorithms for MAP estimation of Markov sequences over noisy channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1539–1544, July 2004.
- [30] M. Park and D. J. Miller, "Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs," in *Proc. Conf. on Info. Sciences and Systems*, Princeton, NJ, Mar. 1998, pp. 477–482.
- [31] —, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.
- [32] Z. Wang and X. Wu, "Length-constrained MAP decoding of variable length encoded Markov sequences," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1259–1266, July 2006.
- [33] Q. Chen and K. Subbalakshmi, "Joint source-channel decoding for MPEG-4 video transmission over wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1780–1789, Dec. 2003.
- [34] B. Pettijohn, M. Hoffman, and K. Sayood, "Joint source/channel coding using arithmetic codes," *IEEE Trans. Commun.*, vol. 49, no. 5, pp. 826–836, May 2001.
- [35] M. Grangetto, P. Cosman, and G. Olmo, "Joint source/channel coding and MAP decoding of arithmetic codes," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 1007–1016, 2005.

- [36] F. Lahouti and A. Khandani, "Efficient source decoding over memoryless noisy channels using higher order Markov models," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2103–2118, Sept. 2004.
- [37] K. Sayood, F. Liu, and J. D. Gibson, "A constrained joint source/channel coder design," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 9, pp. 1584 – 1593, Dec. 1994.
- [38] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, no. 9, pp. 2449–2457, 1995.
- [39] A. H. Murad and T. E. Fuja, "Joint source-channel decoding of variable-length encoded sources," in *Proc. Information Theory Workshop*, Killarney, Ireland, June 1998, pp. 94–95.
- [40] M. Jeanne, J. C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 10–15, Jan. 2005.
- [41] F. Gray, "Pulse code communication." United States Patent Number 2632058, Mar. 1953.
- [42] D. R. Smith, *Digital Transmission Systems*, 3rd ed. Kluwer Academic Publishers, 2004.
- [43] Y. Yamaguchi and T. S. Huang, "Optimum binary fixed-length block codes," Quarterly Progress Report 78, M.I.T. Research Lab. of Electronics, Cambridge, MA, pp. 231-233, July 1965.
- [44] T. S. Huang, "Optimum binary code," Quarterly Progress Report 82, M.I.T. Research Lab. of Electronics, Cambridge, MA, pp. 223-225, July 1966.
- [45] T. Crimmins, H. Horwitz, C. Palermo, and R. Palermo, "Minimization of mean-square error for data transmitted via group codes," *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 72–78, Jan. 1969.

- [46] N. Rydbeck and C. Sundberg, "Analysis of digital errors in nonlinear PCM systems," *IEEE Trans. Commun.*, vol. 24, no. 1, pp. 59–65, Jan. 1976.
- [47] S. Bingöl and B. Sankur, "Joint source channel coding in vector quantization," in *IEEE Mediterranean Elec. Eng. Conf.*, Madrid, Spain, Sept. 1985, pp. 137–141.
- [48] K. Zeger and A. Gersho, "Pseudo-Gray coding," *IEEE Trans. Commun.*, vol. 38, pp. 2147–2158, Dec. 1990.
- [49] S. W. McLaughlin, D. L. Neuhoff, and J. J. Ashley, "Optimal binary index assignments for a class of equiprobable scalar and vector quantizers," *IEEE Trans. Inf. Theory*, vol. 41, no. 6 Part 2, pp. 2031–2037, Nov. 1995.
- [50] L. H. Harper, "Optimal assignments of numbers to vertices," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, pp. 131–135, Mar. 1964.
- [51] A. J. Bernstein, K. Steiglitz, and J. E. Hopcroft, "Encoding of analog signals for binary symmetric channels," *IEEE Trans. Inf. Theory*, vol. IT-12, no. 4, pp. 425–430, Oct. 1966.
- [52] L. H. Harper, "Optimal numbering and isoperimetric problems on graphs," *Journal of Combinatorial Theory*, vol. 1, pp. 385–393, 1966.
- [53] B. Farber and K. Zeger, "Quantizers with uniform encoders and channel optimized decoders," *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 62–77, Jan. 2004.
- [54] W. H. Kautz, "Optimized data encoding for digital computers," in *Convention Record of the I.R.E.*, vol. 2, pt. 4, 1954, pp. 47–57.
- [55] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 799–809, July 1990.
- [56] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

- [57] P. J. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Dordrecht, Holland: D. Reidel Publishing Company, 1987.
- [58] A. A. E. Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 1, pp. 116–123, Jan. 1987.
- [59] P. Knagenhjelm and E. Agrell, "The Hadamard transform—a tool for index assignment," *IEEE Trans. Inf. Theory*, vol. 42, no. 4, pp. 1139–1151, July 1996.
- [60] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2355–2383, Oct. 1998.
- [61] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 821–834, May 1993.
- [62] J. Barros, J. Hagenauer, and N. Goertz, "Turbo cross decoding of multiple descriptions," in *Proc. IEEE Int. Conf. on Communications*, vol. 3, Apr. 28-May 2 2002, pp. 1398 – 1402.
- [63] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sept. 2001.
- [64] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill Science / Engineering / Math, Aug. 2000, ch. 8, p. 479.
- [65] D.-Z. Du and P. M. Pardalos, Eds., *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, 1998, vol. 3, pp. 241–338.
- [66] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms (Combinatorial Optimization)*, 1st ed. Kluwer Academic Publishers, Jan. 1998.
- [67] X. Wang, X. Wu, and S. Dumitrescu, "On optimal index assignment for MAP decoding of Markov sequences," in *Proc. IEEE Int. Symp. on Information Theory*, Seattle, Washington, July 2006, pp. 2314–2318.

- [68] Z. Miller and D. Pritikin, "On the separation number of a graph," *Networks*, vol. 19, pp. 651–666, 1989.
- [69] A. Raspaud, H. Schröder, O. Sýkora, L. Torok, and I. Vrt'o, "Antibandwidth and cyclic antibandwidth of meshes and hypercubes," *Discrete Mathematics, In Press*, p. doi:10.1016/j.disc.2007.12.058, 2008.
- [70] X. Wang and X. Wu, "Recursive structure and bandwidth of Hales-numbered hypercube," *Discrete Applied Mathematics*, 2007, submitted.
- [71] J. Diaz, J. Petit, and M. Serna, "A Survey of Graph Layout Problems," *ACM Computing Surveys*, vol. 34, no. 3, pp. 313–356, 2002.
- [72] P. Z. Chinn, J. Chvátalová, A. K. Dewdney, and N. E. Gibbs, "The bandwidth problem for graphs and matrices - a survey," *Journal of Graph Theory*, vol. 6, pp. 223–254, 1982.
- [73] L. H. Harper, *Global Methods for Combinatorial Isoperimetric Problems*. Cambridge University Press, 2004.
- [74] J. A. Gubner, "Distributed estimation and quantization," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1456–1459, July 1993.
- [75] P. Willett, P. F. Swaszek, and R. S. Blum, "The good, bad and ugly: distributed detection of a known signal in dependent Gaussian noise," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3266–3279, Dec. 2000.
- [76] J.-F. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 407–416, Feb. 2003.
- [77] M. Srinivasan, "Iterative decoding of multiple descriptions," in *Proc. IEEE Data Compression Conf.*, Mar. 1999, pp. 463–472.

- [78] T. Guionnet, C. Guillemot, and E. Fabre, "Soft decoding of multiple descriptions," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, vol. 2, Lausanne, Switzerland, Aug. 26-29 2002, pp. 601–604.
- [79] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane, "Multiple description lattice vector quantization," in *Proc. IEEE Data Compression Conf.*, Mar. 1999, pp. 13–22.
- [80] A. Aggarwal, M. Klave, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix-searching algorithm," *Algorithmica*, vol. 2, pp. 195–208, 1987.
- [81] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [82] J. C. Maxted and J. P. Robinson, "Error recovery for variable length codes," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 6, pp. 794–801, Nov. 1985.
- [83] M. E. Monaco and J. M. Lawler, "Corrections and additions to "error recovery for variable length codes"," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 6, pp. 794–801, Nov. 1985.
- [84] Y. Takishima, M. Wada, and H. Murakami, "Error states and synchronization recovery for variable length codes," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 783–792, Feb./Mar./Apr. 1994.
- [85] P. F. Swaszek and P. DiCicco, "More on the error recovery for variable-length codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 2064–2071, Nov. 1995.
- [86] G. Zhou and Z. Zhang, "Synchronization recovery of variable-length codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 219–227, Jan. 2002.
- [87] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. of the I.R.E.*, Sept. 1952, pp. 1098–1102.

- [88] U. Manber, *Introduction to Algorithms: A Creative Approach*. Reading, Massachusetts: Addison-Wesley Publishing Company, Oct. 1989.
- [89] A. Rosa, "On certain valuations of the vertices of a graph," *Theory of Graphs (Proc. Inter. Symp., Rome, July 1966)*, pp. 349–355, 1967.
- [90] J. A. Gallian, "A dynamic survey of graph labeling," *The Electronic Journal of Combinatorics*, vol. 15, no. DS6, 1998.
- [91] Y.-L. Lai and K. Williams, "A survey of solved problems and applications on bandwidth, edgsum, and profile of graphs," *Journal of Graph Theory*, vol. 31, no. 2, pp. 75–94, June 1999.
- [92] J. Y. T. Leung, O. Vornberger, and J. D. Witthoff, "On Some Variants of the Bandwidth Minimization Problem," *SIAM Journal on Computing*, vol. 13, p. 650, 1984.
- [93] T. Calamoneri, A. Massini, and I. Vrt'ò, "New results on edge-bandwidth," *Theoretical Computer Science*, vol. 307, no. 3, pp. 503–513, 2003.