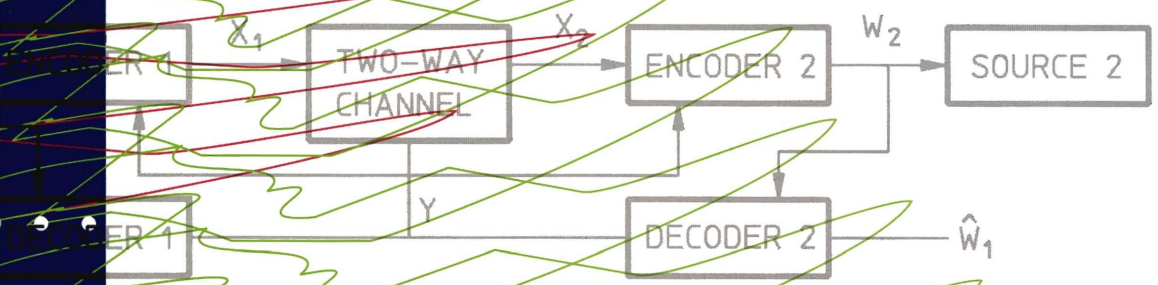
Capacity and coding in digital communications

Andries P. Hekstra

Capacity and Coding in Digital Communications



Capacity and Coding in Digital Communications

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hekstra, Andries Pieter

Capacity and coding in digital communications/ Andries
Pieter Hekstra. -Leidschendam: PTT Research. - Ill.
Proefschrift Eindhoven. -Met lit. opg., reg.
ISBN 90-72125-46-0
Trefw: Digitale communicatie.

Capacity and Coding in Digital Communications

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. J.H. van Lint
voor een commissie aangewezen door het College
van Dekanen in het openbaar te verdedigen
op donderdag 22 december 1994 te 14.00 uur

door
Andries Pieter Hekstra

geboren te Breda

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.P.M. Schalkwijk,

en

prof.dr. T. Berger.

Copromotor: dr.ir. F.M.J. Willems

To my Mother ...

The work presented in this dissertation was carried out at

- Eindhoven University of Technology, Eindhoven, the Netherlands,
- Cornell University, Ithaca, USA,
- PTT Research, Leidschendam, the Netherlands.

This dissertation was prepared while at PTT Research.

Abstract

In the year 1994, approximately half a century since Shannon's foundation of the field, information theory remains to a lively and multi disciplinary field of research. This dissertation presents contributions in the following arreas.

- an upper bound for the capacity region of the two way channel with single output;
- an analysis of the maximum processor density in programmable gate arrays, in which any n pairs of processors can communicate via disjunct routes of communication cells;
- definition of a timing jitter channel and determination of its capacity region;
- a new method for the implementation of the path or state metrics in the Viterbi algorithm using modulo arithmetic;
- an analysis of the numerical range of the path metrics in the Viterbi algorithm, as well as a technique for reduction of this numerical range with a factor of at most two.

Acknowledgements

First of all, I would like to thank my promotor Prof. Piet Schalkwijk of the Electrical Engineering Department of Eindhoven University of Technology. He provided many helpful suggestions and useful remarks.

I would also like to thank Dr. Frans M.J. Willems of Eindhoven University of Technology, who supervised my Master's Thesis work, for showing me the beauty of scientific discovery and for his friendship.

Then I would like to thank Prof. Toby Berger, who was my advisor during my study at Cornell University, for his stimulating enthusiasm, his guidance and his wit. I would also like to thank Prof. Chris Heegard and Talal Shamoon who helped me in many ways while at Cornell University. In addition, I would like to thank Dr. Alon Orlitsky of AT&T Bell Laboratories, Murray Hill, for his encouragement and many stimulating discussions on programmable gate arrays.

I am grateful to my supervisors at PTT Research for their encouragement to pursue this research, Dr. Derk v.d. Houwen, Theo Goumans and Fred Zelders. I would like to thank Dolf Schinkel for his guidance.

I would like to thank Dr. Stan Baggen of Philips Research who convinced me that the time was right to get my Ph.D. degree.

My colleagues Dr. Johan van Tilburg and Marjan Bolle were never tired of helping me with Latex and making a nice booklet out of it.

Finally, I want to express my gratitude to Ellie Aalbersberg for her support and inspiration.

November 1994 *Andries Hekstra*

Contents

0 Introduction **1**

0.1 The scope of information theory 1

0.1.1 Entropy and coding of random processes and fields 4

0.1.2 Rate distortion theory 5

0.1.3 Source coding and entropy theory 5

0.1.4 Source coding 6

0.1.5 Universal source coding 6

0.1.6 Image coding 6

0.1.7 Vector quantization 7

0.1.8 Single-user channels 7

0.1.9 Cyclic codes 8

0.1.10 Convolutional coding 8

0.1.11 Algebraic structure of convolutional codes 9

0.1.12 Concatenated codes 9

0.1.13 Group codes 9

0.1.14 Algebraic geometry codes 10

0.1.15 Properties of codes 10

0.1.16 Soft decision decoding 10

0.1.17 Data communication systems 10

0.1.18 Coding and modulation 10

0.1.19 Trellis coded modulation 11

0.1.20 Performance of trellis-coded diversity receivers over slowly fading channels 11

0.1.21 Trellis decoding for block codes 11

0.1.22 Multi-user channels 12

0.1.23 Multiple and random accessing 12

0.1.24 Code division multiple access sequences and techniques . . . 13
0.1.25 Sequences and arrays . . . 13
0.1.26 Coding for multiple access channels . . . 14
0.1.27 Recording channels . . . 14
0.1.28 Analysis of optical communication systems . . . 14
0.1.29 Coding with partly known errors . . . 15
0.1.30 Special channels . . . 15
0.1.31 Coding for special channels . . . 15
0.1.32 Synchronization . . . 15
0.1.33 Combinatoric designs and tilings . . . 16
0.1.34 Estimation . . . 16
0.1.35 Detection . . . 16
0.1.36 Pattern recognition and estimation . . . 16
0.1.37 Signal processing techniques . . . 16
0.1.38 Neural information processing . . . 17
0.1.39 Complexity of receivers . . . 17
0.1.40 Approximation of information sources . . . 17
0.1.41 Combinatorics . . . 17
0.1.42 Channel equalization . . . 18
0.1.43 Queueing analysis . . . 18
0.1.44 Automatic repeat request systems . . . 18
0.1.45 Data networks . . . 18
0.1.46 Statistical analysis of stationary sources . . . 19
0.1.47 Cryptosystems and wire-tap channels . . . 19
0.1.48 Secret sharing, authentication and key distribution 19
0.2 Introduction to the results . . . 20
0.2.1 The two-way channel . . . 20
0.2.2 Processor densities in programmable gate arrays . 22
0.2.3 Timing jitter channels . . . 23
0.2.4 The Viterbi algorithm . . . 24
0.3 Contributions of fellow authors . . . 27

1 Dependence Balance Bounds ... 31
1.1 Introduction . . . 31
1.2 Definitions and preliminaries . . . 32
1.2.1 The Two-Way Configuration . . . 32

1.2.2 Blackwell's Multiplying Channel 34
1.2.3 Fano's Result 34
1.2.4 K-Information 34
1.3 The Shannon Bounds 36
1.4 The dependence balance bound 37
1.5 The parallel channel extension 40
1.6 An adaptive parallel channel 43
1.7 Results for the multiple access channel with feedback . . 50
1.8 Conclusion 51
1.9 Acknowledgement 51

A The proof of the wringing lemma **53**

B The Zhang *et al.* bound is not tight **57**

2 Asymptotic component densities ... **61**
2.1 Introduction 62
2.2 Definitions 68
2.3 Embedding Degree-1 Graphs of Size 2 or 3 72
2.4 Embedding Degree-1 Graphs of Size 4 76
2.5 Edge-Disjoint Embeddings of Size-k Graphs 82
2.6 Vertex-Disjoint Embeddings of Size-k Graphs 86
2.7 Alternative Models 93

3 On the capacity of a binary channel ... **99**
3.1 Introduction 99
3.2 Definitions and preliminaries 105
3.3 Statement of result 106
3.4 Proof of result 107
3.5 Examples 107
3.6 Discussion 108
3.7 Conclusions 109
3.8 Acknowledgements 110

4 An alternative to metric rescaling in Viterbi decoders 113
4.1 Introduction 114
4.2 Description of the Viterbi Algorithm 115

4.3 Two Properties of the Vitervi Algorithm 116
4.4 The rescaling approach 118
4.5 Two's compl. arithmetic approach 118
4.6 Conclusions 119

5 On the numerical range of the path metrics ... 123

Excerpt (Abstract in Dutch) 161

Curriculum Vitae 165

Chapter 0

Introduction

0.1 The scope of information theory

Since Shannon's seminal papers, information theory has grown to be a subject with a wide range of topics. To demonstrate the width of the spectrum of subjects, consider the list of sessions of the 1994 IEEE International Symposium on Information Theory.

- entropy and coding of random processes and fields,
- rate distortion theory,
- source coding and entropy theory,
- source coding,
- universal source coding,
- vector quantization,
- image coding,
- single-user channels,
- multi-user channels,
- multiple and random accessing,

- universal source coding, stochastic complexity,
- cyclic codes,
- concatenated codes,
- group codes,
- algebraic geometry codes,
- algebraic structure of convolutional codes,
- properties of codes,
- soft decision decoding,
- convolutional coding techniques,
- data communication systems,
- trellis coding on fading channels,
- trellis coding for block codes,
- trellis coded modulation,
- coding for multiple access channels,
- code division multiple access sequences and techniques,
- sequences and arrays,
- recording channels,
- optical communication systems,
- coding with partly known errors,
- special channels,
- coding for special channels,
- synchronization,

- combinatoric designs and tilings,
- estimation,
- detection,
- pattern recognition and estimation,
- signal processing techniques,
- neural information processing,
- complexity of receivers,
- approximation of information sources,
- combinatorics
- coding and modulation,
- channel equalization,
- queueing analysis,
- automatic repeat request systems,
- data networks,
- random processes and random fields,
- statistical analysis of stationary sources,
- cryptosystems and wire-tap channels,
- secret sharing, authentication and key distribution,

In the following paragraphs we briefly introduce these topics to the non-informationtheorist and consider their relation to the central meaning of information theory, i.e. the fundamental study of information, information transmission and information processing. In our description of these topics we do not limit ourselves to the contents of the symposium mentioned. In the following section, the topics for which

this thesis presents contributions will be put into special focus. The common theme of the contributions is that they deal with capacity problems and coding problems in information theory. The three capacity problems that are addressed are that of

- the information rates in two-way channel with single output,
- the packing density of processor cells in a programmable gate array,
- the information rate of a class of timing jitter channels.

The two "coding problem" contributions concern the implementation of the Viterbi algorithm.

0.1.1 Entropy and coding of random processes and fields

The entropy of a random variable X with probability distribution $p(x)$ goes back to Shannon's landmark paper [1],

$$H(X) = -\sum_x p(x) \log_2 p(x). \tag{0.1}$$

A key theorem is that for any uniquely decodable (data compression) code for X, the expected length $E(L)$ of the code cannot be less than the entropy of the source $H(X)$. The difference

$$R \stackrel{def}{=} E(L) - H(X) \tag{0.2}$$

is called the redundancy of the code. Using Huffman codes [2] a redundancy of at most one bit can be achieved for any random variable X.

A random process is an (at least) one dimensional array of random variables. A random field is a two dimensional array of random variables. The interdependence between the random variables complicates the computation of the entropy rate of these random processes/random fields per source symbol as well as the coding methods.

0.1.2 Rate distortion theory

The entropy rate of a random process equals the amount of bits per source symbol necessary to exactly reproduce the source output. In rate distortion theory, one considers the number of R bits per source symbol necessary to reproduce the source output up to a certain accuracy, or distortion d. In many practical cases, e.g. coding of image or sound signals, it is sufficient to specify a signal up to a given small distortion, which remains unperceptible to the human receiver. The larger the distortion d that is allowed, the smaller the required rate R. The result is a convex $R(d)$ curve.

0.1.3 Source coding and entropy theory

Source coding, i.e. data compression problems can be considered involving more than one receiver and or transmitter. For instance, take the multiple description problem. Data compression is applied to a sequence of independent identically distributed (i.i.d.) random variables. The resulting message however is designed to consist of two pieces. Receiver 1 receives only piece 1 which requires a data rate of R_1 bits per source symbol. Similarly, receiver 2 receives only piece 2 which requires a data rate of R_2 bits per source symbol. Receiver 3 receives both constituents of the message at data rate equal to the sum of both rates $R_3 = R_1 + R_2$. This communication situation is intended to model the situation in which it is not sure whether both messages will arrive at the receiver. Receiver 3 can reconstruct the source sequence with a relatively high accuracy, i.e. a relatively low distortion d_3. Receivers 1 and 2 are only able to reconstruct the source sequence with some larger distortion d_1, d_2, respectively. The problem is to study the coding methods which minimize the rate triples (R_1, R_2, R_3) and minimize the distortions (d_1, d_2, d_3). This leads to a six dimensional shape of all achievable rate-distortion tuples $(R_1, R_2, R_3, d_1, d_2, d_3)$, which fully characterize which rate-distortion tuples can occur in practise and which are impossible.

Many other source coding problems exist. E.g. two persons wish to exchange messages, but the messages they have are not statistically independent. Again, the interdependence complicates the problem. The

minimum number of bits necessary to exchange the messages can be studied as a function of the number of rounds of communication that are allowed, etc.

The aforementioned problems are both instances of multi-user source coding, because they involve multiple terminals (transmitters, receivers).

0.1.4 Source coding

Classical source coding problems consider data compression with only a single transmitting and a single receiving user. When an in principle infinite length data stream, e.g. bit stream, is to be compressed, two common approaches exist. One is to chop the data stream into fixed length messages words which are translated into variable length codewords. Relatively more likely message words are translated into relatively shorter codewords and vice versa. This is called fixed-to-variable length coding. Also the oppossite approach exists, variable-to-fixed length coding. Then, the data sequence is chopped into message words of about equal probability of occurrence. Various methods can be considered to achieve fixed-to-variable and variable-to-fixed length coding.

0.1.5 Universal source coding

In classical source coding, one generally assumes that the receiver and the transmitter know the statistics of the source. For instance, the source can produce ASCII symbols according to some given distribution function in a memoryless way. In many practical cases, however, the statistics of the source are a priori unknown and must be estimated in some way, before they can be used in a source coding scheme. Universal source coding concerns itself with this problem: compression of data sequences with unknown statistical properties.

0.1.6 Image coding

Image coding is a special branch of source coding which concerns itself with data compression of image information. Each image is represented by three arrays with colour information for each image point (pel). A

distinction can be made between single image data compression and compression of sequences of images (video coding). Two principles can be exploited in order to obtain data compression. First of all, image information is statistically correlated. Decorrelation techniques result in data compression. Secondly, the human eye has a particular sensitivity. For instance, the human eye is less sensitive to fine grain inaccuracies than to distortions that affect entire blocks of the image. By shaping the distortions in the frequency domain, relatively large distortions can be allowed where the human eye is least sensitive. Image coding thus corresponds to a complex example of rate distortion theory: The signal has a complicated stochastic structure, and the distortion function involves multiple masking effects inherent to the human psychovisual system.

0.1.7 Vector quantization

Quantization amounts to a form of truncation. For instance, "round off to the nearest multiple of 8", or "round off to the nearest power of 2". Thus, the essence of quantizer is to replace the input value by an output value from a limited alphabet. Vector quantization involves the same principle, applied to n-tuples, i.e. vectors. Vector quantization makes use of a codebook, i.e. a set of output vectors that the quantizer can produce. This codebook is designed with the aid of a so called training set of experimental data that is to be quantized. During the training phase an analysis is made of the frequency of different input vectors. The codebook vectors are chosen such that they maximally look alike the occuring input vectors.

0.1.8 Single-user channels

In information theory, a channel is a device by means of which information can be transmitted. Single user channels involve only a single transmitter and a single receiver. Storage of information is considered as a special case, viz. transmission in the time dimension. In all interesting cases, the channel is imperfect, in that (in some cases) it leaves the receiver with some (statistical) uncertainty about which signal was input to the channel by the transmitter. Consider for in-

stance a telephone line, or a satellite link that limits the bandwidth of the transmitted signal, which is, therefore, distorted. In addition, the signal is received with noise added. The receiver's problem is to estimate the transmitted signal sequence and, thereby, the message with minimal (average) error probability.

0.1.9 Cyclic codes

A decoding error occurs in channel coding if the receiver cannot decide what signal sequence (codeword) was transmitted by the sender. To minimize the decoding error, the codewords must be distinctly different, so that they remain discernable when the channel has introduced a limited amount of errors (substitutions). That is, any two codewords must have some prespecified number of indices in which they differ. The number of indices in which two sequences differ is called the Hamming distance. The set of all codewords, binary sequences that the user may send, is called the codebook, or the code. A binary error correction code has as its parameters: the codeword length n, the logarithm k of the number of codewords, and the minimum Hamming distance d between any pair of codewords.

A special branch of applied discrete mathemetics concerns itself with the design of codes that have maximum minimum Hamming distance. Cyclic codes are codes for which one can think of the first index following the last index. That is, the codewords can be thought of being placed on a circle. With a cyclic code, each cyclic shift of a codeword is also a codeword.

0.1.10 Convolutional coding

With a block code, the encoder accepts k message bits, e.g. $k = 57$, and outputs codewords of length n, e.g. $n = 63$. The decoder accepts blocks of n bits and outputs an estimate of the message sequence of k bits. With a convolutional code, the encoder consists of k shift registers. The message sequence is thought of as an in principle infinite length bit stream. The bit stream is chopped into k-tuples. Out of each k-tuple, one bit is fed into each shift register. There are n, $n > k$, linear output functions. Each output is a linear function of any of the bits in the

k shift registers. The codeword bit stream consists of a sequence of n-tuples. The decoder of a convolutional code has to perform sequence estimation. In many practical cases, $k = 1$. Then there is only one shift register. The convolutional encoder outputs n output bits for each message bit entered into the shift register. The n output functions are in fact finite impulse response (FIR) filters.

0.1.11 Algebraic structure of convolutional codes

A convolutional code may be generated by more than one encoder. It is useful to find the minimal, or simplest encoder. Also special properties of a certain convolutional code can be studied in relation to simplified decoding algorithms.

0.1.12 Concatenated codes

Concatenation of two codes amounts to placing two encoders in sequence. That is the message sequence is encoded with the first code. The result is encoded with the second code. Thus, the message is doubly protected against errors. At the receiver's side, the received sequence is first decoded with the second (so called inner) code. Thus, many errors are corrected. The result is decoded using the first (outer) code and remaining errors are corrected. Note that the encoder-decoder pairs are placed as brackets around the channel. In order for the concatenation to be effective, the two codes must be matched in a certain sense. Errors which are not correctable with the inner code must be correctable with the outer code. Also the error rate performance of the concatenated code is an issue of study.

0.1.13 Group codes

Classically, code design concerns itself with binary codes. However, not all channels have binary inputs. E.g. with phase modulation, higher cardinality signal sets can be used. Under some specific conditions, the error correcting codes in such cases are codes over mathematical groups.

0.1.14 Algebraic geometry codes

A relatively new branch of code design that makes use of the subfield of algrebraic geometry of mathematics. This subfield has yielded some very good codes. Recently, effective decoding algorithms for algebraic geometry codes have been found.

0.1.15 Properties of codes

Symmetries and other invariances of codes can be used to attain simplifications during encoding or decoding. Also properties of codes can be of interest when coding systems or modulation systems are concatenated. An important property is cyclicity of a code. A code does not have to be evidently cyclic by construction, whereas it is still cyclic, or the indices can be permuted to obtain a cyclic code.

0.1.16 Soft decision decoding

Assume that the codewords transmitted over a channel are binary. Then, it is not necessarily the case that what is received is again a sequence of binary values. For instance what is received can be a real-valued voltage. Of course, this voltage can be truncated into a binary signal, but then information is thrown away. With soft decision decoding, the voltage is e.g. truncated into one out of 8 levels instead of only 2 levels (the latter would be called hard decision). This 8-valued signal is then used in the decoder. Whenever applicable, the use of soft decision gives a marked performance improvement.

0.1.17 Data communication systems

System aspects of communication systems involving error correction codes, modulation schemes, multiplexing schemes, in the presence of synchronization errors, fading channels, etc.

0.1.18 Coding and modulation

In the recent history of information theory, the combination of error correction and modulation has yielded an improvement in the perfor-

mance of modems. The classical approach to modulation would be to choose input signals to a channel such that the resulting probability of error would be small. With coded modulation the number of input signals is increased such that the resulting channel error probability is relatively large. In a clever way, only the detail signal that is added to the input signal is protected with forward error correcting codes. The coarse grain of the input signal is not protected with forward error correction, or with a much less powerful error correcting code. The introduction of the extra detail in the input signal minus the cost of protecting it with forward error correction gives a substantial gain in information rate. Because only the detail signal is heavily protected with forward error correction, the resulting increase in coder complexity and redundancy of the code is minimal.

0.1.19 Trellis coded modulation

Subject of research are the choice of the constellation of input signals, the forward error correction that is added, how to apply trellis coded modulation to channels that employ phase modulation, performance of trellis coded modulation over intersymbol interference channels, etc.

0.1.20 Performance of trellis-coded diversity receivers over slowly fading channels

Fading radio channels occur e.g. in mobile communications. In certain circumstances, the reception of a signal may drop to zero, e.g. because of the blockage of a radio signal by a building. When the receiver moves out of the shadow of the building, the reception of the signal can be continued. Special coding techniques are used to combat such adverse transmission conditions.

0.1.21 Trellis decoding for block codes

Consider a rate $R = 1/n$ convolutional encoder. The encoder consists of one length m shift register. Initially, at time index zero, the encoder state is assumed all zeroes. For each of the two possible input symbol values to the shift register, the all zeroes state has a successor states at

time index one. At time index two, there are four states, etc. At time index m and beyond, the number of states settles at a fixed number of 2^m states. The connections between states and each of its two successor states can be depicted in a graph called a trellis. The trellis is the state transition diagram of the convolutional encoder. It serves as a tool in the Viterbi decoder algorithm.

Since the Viterbi algorithm is a handy tool for soft decision decoding, and soft decision decoding of block codes rather than standard hard decision decoding brings a substantial performance gain, a method has been devised to give block codes a trellis description. Then, the Viterbi algorithm can be used to decode such codes.

0.1.22 Multi-user channels

Multi-user channel theory involves more than one transmitter and/or more than one receiver. An important example is that of the multiple access channel and the two-way channel (the two-way channel is introduced, below). The multiple access channel has two (or more) inputs, one for each transmitting user, and one output, connected to the receiver. The users may only communicate with each other via the channel. The messages of the transmitters are statistically independent. In this communication situation no noise source needs to be present in the channel. The communication of one message to the receiver hampers the reception of the other messages. An simple approach is to divide the time equally among the transmitters, and let only one transmitter send information at a time. However, in general, such a time-sharing approach does not yield the maximum average communication rates that can be achieved in such a configuration.

0.1.23 Multiple and random accessing

In mobile communications and computer communications, it is often the case that there is a single transmission medium, e.g. a LAN cable or a certain radio frequency band, that can be seized by only one transmitter at a time. All users involved can receive the signal transmitted by a sending user. Time is divided into time slots, and all users have synchronous clocks. Whenever, a user stops sending, immediately the

channel is open for transmission by other users. When two or more users start transmission simultaneously, a collision occurs, and the pertaining time slot is wasted in the sense that the users only know that more than one user want to transmit, but not which user nor what was transmitted. Strategies can be devised that minimize the number of collisions that occur, and maximize the utilization of the transmission medium. Upper bounds can be derived on the maximum utilization rate of the channel.

0.1.24 Code division multiple access sequences and techniques

Code division multiple access (CDMA) uses noise-like waveforms for communication. The encoder and decoder are assumed to be in time synchronization. The encoder sends "+ noise like waveform" or " - noise like waveform". The decoder correlates the received signal with the noise like waveform and receives a "+" or a "-". The trick is that because of the use of the noise-like waveform, this communication is difficult to intercept by someone who does not know what waveform is used. In the frequency spectrum, the noise-like signal has its energy smeared out over a very wide band. Thus, the signal energy can remain well below the noise level. Apart from secret communications, CDMA can be used for multiple access. Different senders each have their own noise-like sequence. Because the cross correlation between different noise sequences is almost zero, different users do not interfere with each other. Each additional user just adds slightly to the overall noise level.

0.1.25 Sequences and arrays

The sequences used in CDMA must be noise like. To define what "noise like" means and to find sequences that have these properties is a special topic. Maximum length shift register sequences can do the job, but there are not very many of them. With secret communications, one wants to have a large supply of "good" sequences to choose from. Noise-like arrays are sometimes used in radar applications.

0.1.26 Coding for multiple access channels

Alternatives to CDMA are time division multiple access (TDMA) and frequency division multiple access (FDMA). With TDMA, each user has a different time slot. With FDMA each user has a different frequency band. With a large number of users that seldomly use the channel both TDMA and FDMA are inefficient, because a large portion of the time period (frequency band) is alloted to inactive users. With CDMA, only those users that are sending add slightly to the overall noise level.

0.1.27 Recording channels

A channel which is of especial practical importance is the recording channel. The optical recording channel (CD) and the magnetical recording channel can be distinguished. With both recording channels, assume that a digital signal is recorded. During play off, transitions in the input signal are used to derive a clock signal. If transitions occur too far spaced apart in time, it becomes a problem to distinguish exactly how many bits are in between the transitions. Therefore, the maximum distance between transitions must be limited. Similarly, if transitions occur with too high a frequency, due to the bandlimitedness of the detector, these transitions lead to inter symbol interference. In short, a minimum spacing $d+1$ and a maximum spacing $k+1$ between the transitions is prescribed.

A branch of information theory concerns itself with the design of optimal (d, k) codes, that have maximal information density and yet have simple encoders and decoders. Also joint (d, k) constrained and error correction codes are studied.

0.1.28 Analysis of optical communication systems

Laser communication through optical fibers is a special branch of channel coding/modulation that requires its own analysis. Typically, the detector is assumed to count the received photons. Photon noise, bandwidth restrictions and intersymbol interference can limit the possible communication rate.

0.1.29 Coding with partly known errors

Classically, error correction codes are designed to cope with one of two types of errors. With so called random errors at most e errors can occur anywhere in the codeword. With burst errors, a number of errors can occur at successive positions over an index range of at most b. However, other situations can occur in practise. A priori knowledge can be available about the set of error vectors. Or, with nonbinary codes, the values of the errors can be known and only their locations need to be resolved. For each situation, special error correction codes need to be designed.

0.1.30 Special channels

Several topics can be classified under this subject. For instance, multi-user channels such as the two-way binary multiplying channel, bandlimited additive Gaussian noise channels in the presence of sampling jitter (see also 'Synchronization'), and channels with insertions and deletions.

0.1.31 Coding for special channels

Classically with error correction, it is assumed that 0's can go over into 1's and vice versa. If only 0's can turn into 1's but not the other way around, one has so called asymmetrical errors. Asymmetrical error detection and error correction codes are a special subject.

0.1.32 Synchronization

In classical information theory, one assumes that sender and receiver are in time synchronization, and that only the noise in the channel limits the communication capacity between sender and receiver. In practise, sender and receiver do not have perfect clocks available, and the timing uncertainty can limit the communication capacity, too. For example, consider the replay of a disc. Inaccuracies in the replay velocity incur a timing discrepancy at the "receiver".

Differences in timing velocity may the receiver to miss certain bits, or to insert spurious bits. This is referred to as the channel with inser-

tions and deletions. Coding for this channel is a relatively new branch of coding theory.

0.1.33 Combinatoric designs and tilings

A subject which generalizes upon the design of good error correction codes.

0.1.34 Estimation

Estimation of stock performance turns out to be a problem closely related to universal data compression. Estimation of probability densities and entropies is another subject of interest.

0.1.35 Detection

Detection of signals or the presence of signals such as is the case with radar, and the computation of false alarm probabilities is a discipline by itself. Distributed detection entails detection using more than one receiver and combining the results before making a decision. Maximum likelihood rules and other rules can be used to make decisions.

0.1.36 Pattern recognition and estimation

E.g. recognition of letters in a digitized camera output signal can be an important application of patter recognition. Artificial neural networks can be trained to recognize particular shapes, e.g. bomb craters. In certain circumstances, bounds can be derived on the error probabilities of pattern recognition and estimation. See also 'Neural information processing'.

0.1.37 Signal processing techniques

Signal processing techniques are e.g. used in image coding and in image restoration. Also signal analysis, e.g. frequency and wavelet analysis are of interest to the informationtheorist.

0.1.38 Neural information processing

Artificial neural networks mimic certain brain functions and can be used to realize associative memories. These networks contain a large number of connection weights that, during the learning phase, are positively reinforced to produce certain given outputs in response to certain given inputs. After the learning phase, the neural networks can be used to produce the outputs in response to the inputs. Also, these networks have a certain generalization capability.

0.1.39 Complexity of receivers

Shannon proved that good error correcting codes can be found by simple coin tossing (in case of a binary channel when a uniform input distribution realizes channel capacity). The problem is that for such a random code, the encoder and decoder are extremely complex. Thus, a good code is only really valuable if simple encoder and decoders exist for this code. The decoding problem amounts to finding the codeword closest to the received word. Because of this search aspect, decoding is inherently more complex than encoding. Given an error correcting code, the encoder and decoder are not at all uniquely determined. The degrees of freedom can be used to achieve simplifications. In many broadcast situations, only one encoder has to be realized, but many decoders. Then, decoder complexity is especially important.

0.1.40 Approximation of information sources

Approximation of information sources can be of interest for theoretical analysis and for practical simulations. E.g. also in simulations of particular information sources, can approximation be of interest.

0.1.41 Combinatorics

Zero error communication leads to graph problems. Therefore, for instance graph problems are relevant to information theory.

0.1.42 Channel equalization

A transmission channel can have a certain non-ideal frequency transfer function. E.g. the channel can have a low pass behavior. Then, high frequencies are blocked. Frequencies that lie in the transition range of the frequency transfer function can suffer attenuation and phase distortion. Such distortion can work out to a "smearing out effect" on pulses transmitted accross that channel. Equalization means compensation of the low pass characteristic, so that within the frequency range of the channel that is actually used, it has a flat amplitude characteristic and linear phase.

How to control the equalization filter such that this is achieved can be approached in different ways. One approach is to sometimes send a signal with a known frequency characteristic. Then, the frequency transfer function of the channel can be calculated. Another approach is to calculate the frequency transfer function of the channel "on the fly".

0.1.43 Queueing analysis

Oftentimes information is transmitted in the form of packets. Packets can await transmission in queues, as is e.g. the case in broadband ISDN network nodes. Therefore the analysis of queueing disciplines is of special interest to information theory.

0.1.44 Automatic repeat request systems

If the receiver has a feedback link to the sender, the sender need only use a code for error detection rather than error correction, and the receiver can request for all data packets received in error to be repeated. This approach to error correction does impose extra transmission delay before the retransmitted data packets are available.

0.1.45 Data networks

Important issues with data networks are the delay a data packet encounters when transmitted through the network, the maximum through-

put in terms of number of packets per second that can be transmitted between the various points in a network and stability issues. Many data networks contain queues and therefore, require queueing analysis. Topology issues can be addresses of how to serve a large number of users most effectively. Protocols are used to modularize the functional entity of a network into several layers. In short, there are many informationtheoretical and statistical problems related to data networks.

0.1.46 Statistical analysis of stationary sources

Stationary sources have a certain probability density function that can be analyzed. The probability density function provides a lot of insight about the stochastic structure of the source. Stationary sources have a spectrum. Spectral analysis is one of the corner stones of linear system theory. Spectral analysis is a topic by itself.

0.1.47 Cryptosystems and wire-tap channels

Cryptography is an important part of information theory. The purpose of secure communications is to make data unintelligible to an unauthorized user, who is not in possession of the secret key information. To break in on such a scheme is not theoretically impossible, it just takes a very large computation time. These computation times depend on the creativity of the attacker, so a cryptosystem is as secure as the lowest complexity attack for that scheme.

0.1.48 Secret sharing, authentication and key distribution

Protection of confidentiality against a wire-tapper is not the only subject of cryptography. Digital signatures by means a user can unmistakenly prove his identity are another important topic. Also the distribution of secret key information among different users is topic of interest.

0.2 Introduction to the results

Results have been obtained on the following topics:

- upper bounds on the capacity of the single output two-way channel (multi-user channels),
- the density of processor cells in a programmable gate array network (data networks),
- the capacity of a timing jitter channel (synchronization, special channels),
- implementation of the Viterbi algorithm (convolutional coding).

We now present introductions to each of these topics.

0.2.1 The two-way channel

In the simplest situation a two-way channel consists of two one-way channels, one from user A to user B and from user B to user A. The set of all rate pairs (R_1, R_2) that are achievable for such a constellation, i.e. the capacity region, consists of a rectangle, as follows:

$$\{(R_1, R_2) \mid 0 \leq R_1 \leq C_{AB}, 0 \leq R_2 \leq C_{BA}\}, \tag{0.3}$$

where C_{AB}, C_{BA} denote the capacity of the channel from A to B and v.v., resp. As evidence that the communications in both directions have nothing to do with each other, the inputs to both channels $A \rightarrow B$ and $B \rightarrow A$ are statistically independent.

Theoretically more interesting are those channels for which the communication in the first $A \rightarrow B$ direction interferes with the communication in the second, $B \rightarrow A$ direction. The capacity region of such channels is not rectangular. Also the inputs to the two-way channel are in general not statistically independent. However, because the inputs reflect message information and the messages are statistically independent, any dependence between the inputs of the channel must have been created during previous transmission. Considerations like this led to the dependence balance bound.

Chapter 1 presents an upper bound for the capacity region of the two way channel with single output on the basis of a so called dependence balance bound. In case a two way channel consists of two separate one way channels in both directions, the input signals to both channels in the optimal case are independent. For the general two way channel, statistical dependence between the inputs can increase the achievable data rates. The inputs are derivates of the messages, and the messages are independent. Therefore, the first input signals in any strategy are necessarily statistically independent. This does not hold for the subsequent input signals, but what can be proved is that on average the amount of dependence between the input signals cannot be larger than the amount of a posteriori conditional dependence of the input signals given the output signal. The amount of dependence is measured with the (conditional) mutual information. In essence, any dependence between the input signals first has to be created. In addition, the introduction of an additional parallel channel can only increase the capacity of the combined channel. Such a parallel channel can be chosen such that the a posteriori dependence of the input signals given the output signal is reduced, and that the amount of information that is given away by the parallel channel is as small as possible.

The new outer bound to the capacity region that has been obtained is the strongest bound yet known. It allows the determination of upper bound results for new two way channels of interest, and provides a lot of insight to the role of dependence in the construction of good codes for the two way channel.

The key ingredient in this contribution was to measure the dependence between the terminals with the conditional mutual information and require that *on average* as much dependence is produced as is consumed. Further research could aim other relationships between the a priori and the a posteriori distribution of the inputs given the outputs to further tighten the bound obtained.

0.2.2 Processor densities in programmable gate arrays

The design of error correction codes amounts to a packing problem. How to pack as many codewords in the space of all sequences of a certain length as possible without that the spheres around these codewords with a radius of $\lfloor (d-1)/2 \rfloor$ intersect?

A more complex packing problem is the following: How can as many processor cells as possible by packed in, say, a two-dimensional grid such that pairs of of processors satisfy the constraint of being mutually connectable?

The simplest case has an appealing analog to the configuration of parking lots. How can as many cars as possible be parked on a two dimensional grid of cells such that any car can be moved to an exit cell without having to move other cars? When the movements of the car are bidirectional, this means that each car can move to the exit cell and from there to any other car. That is, any two cars can be connected via a route over empty cells.

Chapter 2 discusses what with some feeling for analogies can be called parking lot theory. Given a rectangle that is partitioned into small rectangular cells, all of equal size, the cells can be used as processor cells or as communication cells. The target is to achieve the highest possible density of processor cells under the constraint that every pair of processor cells can communicate via a path of communication cells. The largest achievable density turns out to be 2/3. If the processor cells are considered as cars and the communication cells are considered as empty cells, a single processor cell can be coined exit cell, and then all cars can move to the exit. Vice versa, if in a parking lot all cars can move to an exit cell without that other cars have to be moved away, if the cars can move bidirectionally, any car can move to the exit cell and from there to any other car. Thus, any two pair of cars are connected. More generally, we seek the maximum processor density given that n pairs of processors can communicate via n disjoint paths. The optimal density turns out to be $O(n^{-2})$.

It should be noted that this contribution also amounts to a capacity problem related to the packing of points in a space. "Packing points in a space subject to a condition" is also the form of the problem of

finding good error correcting codes. Only in this case the condition is a connectivity condition which is more complex than the usual minimum distance requirement.

Further research could aim at the analysis of more realistic programmable gate arrays with multiple layers and more general networks to be embedded.

0.2.3 Timing jitter channels

In classical Shannon channel theory, each channel input symbol gives rise to one channel input symbol. Implicitly this means, that sender and receiver have a perfect clock. In practise, this often is not the case. For instance, the channel can be a storage device which is replayed. If the replay velocity is not perfectly equal to the recording velocity, timing errors occur. The output signal is resampled. It is evident that timing errors can affect the correctness of the message that is received and can reduce the information capacity of the channel. This problem is also addressed by Heegard, et. al. [3]. The binary input signal of a timing jitter channel consists of successive runs of zeroes and ones. We assume that the timing jitter is not synchronous, but run-synchronous. That is, the lengths of runs of zeroes and ones can be changed, but not entire runs are omitted. Thus, the problem can be reformulated in terms of a channel that has run variables as input and output. The capacity of this channel can be analyzed using a result of Verdu about capacity per unit cost. Here, cost is the length of the input run. In this way, the capacity of the channel is renormalized to the information capacity per input bit.

Chapter 3 introduces channels with timing jitter. Shannon's well known result for the capacity of the one way channel presumes that sender and receiver have a perfect clock, and thus are capable of submitting and sampling the input and output signals at the correct time. In general, channels transport signals not only in the spatial dimensions but also in the time dimension. Following Baggen and Wolf, we consider a class of channels that occur, e.g., in storage media such as compact disc and optical or magnetic tape. If the velocity at which such a medium is played is not exactly equal to the velocity at which the information was recorded, uncertainty is created in the time relation

between sender and receiver. Assume that the signal has binary input and output signals. The length of a pulse train of 1's or 0's, also called a run of 1's or 0's, can be affected by an incorrect sampling speed. If it is assumed that only the runlengths are affected, as opposed to entire runs being missed or inserted, sender and receiver agree on the run index number which, in turn, can serve as an alternative synchronization index. Reformulation of the problem in runlengths allows for exact determination of the capacity region of this channel using a theorem of Verdu about capacity per unit of cost.

The results obtained allow for an easy determination of the capacity of an important class of timing jitter channels. For instance in the case of compact disc, the information written on the disc is coded such that it has a hole around zero in the frequency spectrum. Thus, a low frequency pilot tone written on the disc can be recognized in the frequency domain. The result is that for the data written on the disc - a timing jitter channel - side information is available about the transmitter's time. This led Baggen and Wolf to introduce a different timing jitter channel in which the running sum of the timing disparities is bounded. This is motivated by the boundedness of the timing uncertainty because of the side information. In our approach, however, the creation of a hole in the frequency spectrum is considered as a loss of channel capacity. We do not consider side information. Instead e.g. the timing uncertainty can be bounded by letting the transitions in the input to the channel happen at a prescribed rate. The estimate of the transmitter time then equals the transition rate times the transition index. Further research could analyze the capacity of this timing jitter channel with a restriction on the code such that the timing uncertainty is bounded. This research could be motivated by a promising improvement in efficiency over the pilot tone approach.

0.2.4 The Viterbi algorithm

The Viterbi algorithm is a very widely used tool for maximum likelihood sequence estimation. Assume e.g. that symbols $X_i \in \{-1, 1\}$ are sent, and that symbols $R_i \in \mathcal{R}$ are received, $i = 1, 2, \ldots$. Furthermore,

assume that the differences $R_i - X_i$ are normally distributed. Then,

$$p(R_i|X_i) = \mathcal{N}((R_i - X_i)/\sigma^2) \tag{0.4a}$$

$$p(R|X) = \prod_{i=1}^{N} p(R_i), \tag{0.4b}$$

where R, X denote sequences over the index range $\{1, \ldots, N\}$. Maximization of $p(R|X)$ is tantamount to minimization of $\| R - X \|_2$.

Given a state space flow diagram or trellis, the Viterbi algorithm determines that path through the trellis for which the aforementioned quantity is minimal. With hard decision, if the R_i take on values in $\{-1, 1\}$, the Viterbi algorithm find that path (codeword) for which the Hamming distance to the received sequence is minimal.

The Viterbi algorithm has variables which represent a running sum of the distance between the R and X sequences, so far, for each of the encoder states in which a sequence X can end. As these so called metric variables accumulate the difference between these sequences, the problem arises of how to arrive at an implementation using a finite number of bits.

Presented are a new technique to implement the path or state metric variables (modular arithmetic), an analysis technique for exact determination of the numerical range of these variables, and a new technique to actually reduce the numerical range of the metric variables.

The subject of Chapter 4 is the implementation of the path or state metrics in the Viterbi algorithm (VA). The results holds for (almost) any application of the VA, not just for decoding of linear convolutional codes. The path metrics in the VA accumulate the distance between the received channel symbol sequence and the survivor paths in the trellis for the respective states. As the number of channel errors increases, so do the path metrics. In an implementation it is preferred that the path metrics can be represented with a finite number of bits. To this end the following properties of the VA can be exploited. 1. The selection of survivor paths depends only on differences of path metrics. 2. The maximum difference between any two path metrics can be bounded using structure properties of the trellis. A well known implementation method which makes use of these properties is the rescaling method. After each iteration of the VA the minimum path metric is

deducted from all path metrics. This costs hardware and computation time. Presented is an alternative method which obliterates the need for subtraction inside the VA loop. The modulo reduced difference of two path metrics equals the true difference provided that the true difference lies in the range of the modulo operator, which can be chosen approximately symmetric around zero. This corresponds to two's complement arithmetic. Overflows do occur, but do not cause problems.

As a sequel to Chapter 4, in Chapter 5 the maximum difference between path metrics is analyzed for the case of a decoder of linear convolutional codes. The results hold for hard decision decoders and a class of soft decision metrics. It can be proven that the reception of the all zeroes codeword represents the worst case for the maximum difference between path metrics. In addition, it can be proven that, as a function of the depth in the trellis, this difference increases to a maximum value and then (slightly) decreases. What matters for an implementation, is the maximum difference between candidate values for the path metrics inside the VA metric update loop. These difference can be larger than the differences between path metrics. With respect to the maximum difference between candidate path metrics, a distinction must be made between a rescaling implementation and modulo arithmetic. With modulo arithmetic only the difference between candidate path metric values for the same state have to be considered. Because of this, with modulo arithmetic the numerical range of the candidate path metrics is somewhat smaller than with the rescaling method. Furthermore, a selection rule is introduced that prunes nodes from the trellis for which the path metric is relatively large at the given depth in the trellis. Large path metric differences correspond to unlikely survivor paths. It can be proven that some of these paths never can have the overall minimal metric, i.e., cannot be the output path of the VA, because there always is another path that has smaller path metric. The resulting reduction in the numerical range of the path metrics is at most a factor of two.

Chapter 4 presents a very practical method of how to simplify the implementation of the Viterbi algorithm. The results of Chapter 5 can be used to determine the numerical range of the path metrics of a decoder of linear codes exactly. In addition it describes a method of how to reduce this numerical range to achieve a further simplification

of the Viterbi algorithm. Further research could aim at combination of the selection rule of Chapter 5 with the two's complement arithmetic method of Chapter 4. Also Chapter 4 could be extended to nonlinear trellis codes.

0.3 Contributions of fellow authors

The dependence balance bound was the author's Master's thesis. A large part of the results are due to my Master's thesis advisor, Dr. Frans M.J. Willems. The dependence balance bound was obtained as a generalization of Willems' dissertation. The observation that Willems' results had an avoidable asymmetry which led to generalizations is by the author. Also, the observation that the dependence increase $I(A;B) - I(A;B|C)$ is in fact a special case of Csiszar and Körner's mutual information in three variables $I(A;B;C)$ is by the author.

The author independently proved an upper bound $O(1/n)$ for the density of Programmable Gate Arrays (PGA's), in which n pairs of processors are connectable via disjoint paths. Independently, the author conjectured that the true density was $O(1/n^2)$ and proved that the achievability of $O(1/n^2)$. The simpler proof of achievability and the converse result were obtained by Berger and Orlitsky.

Bibliography

[1] C.E. Shannon, A Mathematical theory of communication, *Bell Syst. Techn. Journ.*, Vol. 27, pp. 379-423, pp. 623-656, 1948.

[2] T.M. Cover, J.A. Thomas, *Elements of Information Theory,* Wiley, New York, 1991.

[3] C. Heegard, A. Duel-Hallen, R. Krishnamoorty, "On the capacity of the noisy runlength channel", *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 712-720, May 1991.

Chapter 1

Dependence balance bounds for single output two-way channels

Abstract

If in a transmission the inputs of a single-output two-way channel exhibit some interdependence, this dependence must have been created during earlier transmissions. The idea that no more dependence can be consumed than produced is used to obtain new upper bounds to the capacity region of the discrete memoryless single-output two-way channel. With these upper bounds we can show that Shannon's inner bound region is the capacity region for channels in a certain class and improve upon the Zhang-Berger-Schalkwijk upper bound for Blackwell's multiplying channel.

1.1 Introduction

A challenging problem in multiuser information theory is that of determining the capacity region of the *two-way channel* (TWC). In 1961

[0]Co-authored with F.M.J. Willems from Eindhoven Technical University, Eindhoven, The Netherlands. Published in IEEE Transactions on Inform. Theory, vol. IT-35, nr. 1, pp.44-53, Jan. 1989.

Shannon [1] proposed this problem and found an inner and an outer bound for this capacity region. In general these bounds are different, and we do not know the capacity region. A well-known TWC for which inner and outer bounds do not coincide is Blackwell's multiplying channel (BMC). By means of an example, Dueck [2] showed 18 years after the introduction of the TWC that channels exist for which the capacity region is strictly greater than Shannon's inner bound region. Schalkwijk [3], [4] constructed coding strategies for the BMC and rates exceeding Shannon's inner bound. For general TWC's Han [5] determined an achievable rate region that improves upon Shannon's inner bound region. The first improvements of Shannon's outer bound in the general case were recently obtained by Zhang et al. [6].

We determine a new upper bound for the capacity region of *single-output* TWC's. This upper bound improves upon the bound of Zhang et al. For a certain class of TWC's, our upper bound is equal to Shannon's inner bound and therefore establishes the capacity region. For the multiple access channel (MAC) *with feedback*, analogous results are obtained.

1.2 Definitions and preliminaries

1.2.1 The Two-Way Configuration

A discrete memoryless (DM) single-output TWC denoted by $(\mathcal{X}_1 \times \mathcal{X}_2, P^*(y \mid x_1, x_2), \mathcal{Y})$ consists of three finite alphabets $\mathcal{X}_1, \mathcal{X}_2$, and $\mathcal{Y}$, and a probability matrix $P^*(y \mid x_1, x_2)$. The inputs to the channel are X_1 and X_2 and the output is Y. This TWC is the basic element of the two-way communication system shown in Fig. 1.

Sources 1 and 2 generate messages $W_1 \in \{1, 2, \cdots, M_1\}$ and $W_2 \in \{1, 2, \cdots, M_2\}$, respectively. The messages W_1 and W_2 are statistically independent and uniformly distributed. These messages are to be transmitted to the other terminal via N transmissions.

Each *encoder* is completely described by a set of N encoding functions. These functions map the message and the sequence of received channel outputs into the next channel input. Letting a^t denote $(a_1, a_2, \cdots, a_t)$ for $t = 1, 2, \cdots, N$ and a^0 "empty" we may describe the encoders

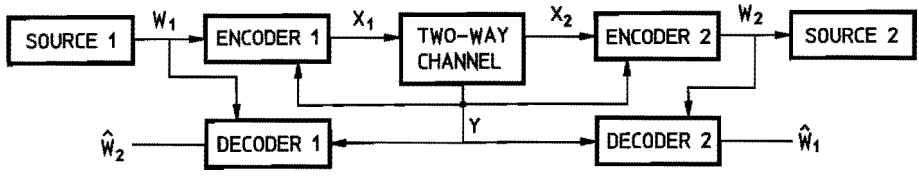


Figure 1.1: Single-output two-way configuration.

as follows:

$$X_{1n} = f_{1n}(W_1, Y^{n-1}) \tag{1.1a}$$

$$X_{2n} = f_{2n}(W_2, Y^{n-1}), \qquad n = 1, \ldots, N. \tag{1.1b}$$

Decoders 1 and 2 produce estimates $\hat{W}_2$ and $\hat{W}_1$, based on their knowledge of their own messages W_1 and W_2, respectively, and the sequence of received channel outputs Y^N. Hence

$$\hat{W}_2 = g_1(W_1, Y^N) \tag{1.2a}$$

$$\hat{W}_1 = g_2(W_2, Y^N). \tag{1.2b}$$

An $(N, M_1, M_2, P_{e1}, P_{e2})$ code for the DM single-output TWC consists of two sets of N encoding functions and two decoding functions such that

$$Pr\{\hat{W}_1 \neq W_1\} = P_{e1} \tag{1.3a}$$

$$Pr\{\hat{W}_2 \neq W_2\} = P_{e2}. \tag{1.3b}$$

A *rate pair* (R_1, R_2) is *achievable* for the DM single-output TWC if and only if for $\delta > 0$ there exists an $(N, M_1, M_2, P_{e1}, P_{e2})$ code with

$$\frac{1}{N} \cdot \log(M_1) \geq R_1 - \delta \tag{1.4a}$$

$$\frac{1}{N} \cdot \log(M_2) \geq R_2 - \delta \tag{1.4b}$$

$$P_{e1} \leq \delta \tag{1.4c}$$

$$P_{e2} \leq \delta \tag{1.4d}$$

The capacity region $\mathcal{C}_{TWC}$ of the DM single-output TWC is the set of all achievable rate pairs (R_1, R_2) with $R_1 \geq 0$ and $R_2 \geq 0$.

1.2.2 Blackwell's Multiplying Channel

The BMC is a deterministic single-output TWC with $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y} = \{0, 1\}$ and $P^*(y \mid x_1 x_2) = 1$ if $y = x_1 \cdot x_2$ and 0 otherwise.

1.2.3 Fano's Result

Fano's result (see, e.g., Csiszár and Körner [7, p. 53]) may be applied to obtain

$$\begin{aligned} H(W_1 \mid Y^N, W_2) &= H(W_1 \mid Y^N, W_2, \hat{W}_1 = g_2(W_2, Y^N)) \\ &\leq H(W_1 \mid \hat{W}_1) \\ &\leq h(P_{e1}) + P_{e1} \cdot \log(M_1), \end{aligned} \tag{1.5a}$$

and

$$H(W_2 \mid Y^N, W_1) \leq h(P_{e2}) + P_{e2} \cdot \log(M_2). \tag{1.5b}$$

Here $h(\gamma) \triangleq \gamma \cdot \log(\gamma) - (1 - \gamma) \cdot \log(1 - \gamma)$ for $0 \leq \gamma \leq 1$ denotes the binary entropy function.

1.2.4 K-Information

Multiple mutual information, or K-information as we shall call it, was first defined by McGill [8]. Fano [9] devoted a subsection of his book [10] to this subject, and more recently, Han [11] used K-information to describe multiple interactions in frequency data. The present subsection is an introduction to K-information.

Let $\{V_0, V_1, V_2, \cdots V_K\}$ be a set of $K + 1$ random variables ($K = 1, 2, \cdots$), and assume that each of these variables takes values in some finite alphabet. Let the random variables S_1, S_2, and S_3 be subsets of $\{V_1, V_2, \cdots V_K\}$. We define the entropy of S_1 as

$$H(S_1) \triangleq \sum_{s_1} -Pr\{S_1 = s_1\} \cdot \log(Pr\{S_1 = s_1\}) \tag{1.6a}$$

and the conditional entropy of S_1 given S_2 as

$$\begin{aligned} H(S_1 \mid S_2) = \textstyle\sum_{s_1 s_2} - \; & Pr\{S_1 = s_1, S_2 = s_2\} \\ & \cdot \log(Pr\{S_1 = s_1 \mid S_2 = s_2\}). \end{aligned} \tag{1.6b}$$

An important property of entropy follows from the chain rule of probability:

$$H(S_1, S_2 \mid S_3) = H(S_1 \mid S_3) + H(S_2 \mid S_1, S_3). \tag{1.7}$$

We are now ready to define the *K-information* I_K

$$I_K(V_1; V_2; \cdots; V_K) = \sum_{k=1,K} (-1)^{k-1} \sum_{\substack{S \subset \{V_1, V_2, \cdots V_K\} \\ \mid S \mid = k}} H(S). \tag{1.8a}$$

and the *conditional K-information*

$$I_K(V_1, V_2; \cdots; V_K \mid V_0) = \sum_{k=1.K} (-1)^{k-1} \sum_{\substack{S \subset \{V_1, V_2, \cdots, V_K\} \\ |S| = k}} H(S \mid V_0) \tag{1.8b}$$

Hence,

$$\begin{aligned}
I_1(A) &= H(A) \\
I_2(A; B) &= H(A) + H(B) - H(A, B) \\
I_3(A; B; C) &= H(A) + H(B) + H(C) - H(A, B) \\
&\quad -H(A, C) - H(B, C) + H(A, B, C,) \\
I_4(A; B; C; D) &= H(A) + H(B) + H(C) + H(D) \\
&\quad -H(A, B) - H(A, C) - H(A, D) \\
&\quad -H(B, C) - H(B, D) \\
&\quad -H(C, D) + H(A, B, C) \\
&\quad +H(A, B, D) + H(A, C, D) \\
&\quad +H(B, C, D) - H(A, B, C, D)
\end{aligned} \tag{1.9}$$

and

$$\begin{aligned}
I_2(A; B \mid C) &= H(A \mid C) + H(B \mid C) - H(A, B \mid C) \\
I_3(A; B; C \mid D) &= H(A \mid D) + H(B \mid D) + H(C \mid D) \\
&\quad -H(A, B \mid D) - H(A, C \mid D) \\
&\quad -H(B, C \mid D) + H(A, B, C \mid D),
\end{aligned}$$

etc. We state without proof two properties of K-information as follows. *The chaining property*:

$$\begin{aligned}
I_K((V_1, V_2); V_3; \cdots ; V_{K+1} \mid V_0) &= I_K(V_1; V_3; \cdots; V_{K+1} \mid V_0) \\
&\quad + I_K(V_2; V_3; \cdots; V_{K+1} \mid V_1, V_0).
\end{aligned} \tag{1.10a}$$

The recursive property ($K \neq 1$) :

$$I_K(V_1; V_2, \cdots, V_K \mid V_0) = I_{K-1}(V_1; V_2; \cdots; V_{K-1} \mid V_0) - I_{K-1}(V_1; V_2; \cdots; V_{K-1} \mid V_K, V_0). \tag{1.10b}$$

Note that the K-information is completely symmetrical in all its K "arguments." This means with the recursive property we have K possible ways of writing I_K, e.g., $I_2(A; B) = I_1(A) - I_1(A \mid B) = I_1(B) - I_1(B \mid A)$.
Clearly I_1 is equal to the entropy and I_2 to the mutual information. Both I_1 (entropy) and I_2 (mutual information) have an intuitive meaning. Until now, I_3 had no natural intuitive meaning (see, e.g., Csiszár and Körner [7, p. 53]). In the proofs given in this paper it turns out that I_3 plays a crucial role. We can think of I_3 as the dependence reduction.

1.3 The Shannon Bounds

Shannon's inner bound region $\mathcal{B}_{si}$ for the single-output TWC is defined in the following way:

$$\begin{aligned} \mathcal{B}_{si} \triangleq \; & \mathrm{co}\,(\{(R_1, R_2) : 0 \leq R_1 \leq I(X_1; Y \mid X_2), \\ & 0 \leq R_2 \leq I(X_2; Y \mid X_1), \\ & \text{for some} \quad P(x_1, x_2, y) \\ & = P(x_1)P(x_2)P^*(y \mid x_1, x_2)\}) \,. \end{aligned} \tag{1.11}$$

Here "co" denotes convex hull.
Shannon's outer bound is defined as

$$\begin{aligned} \mathcal{B}_{so} \triangleq \; & \{(R_1, R_2) : 0 \leq R_1 \leq I(X_1; Y \mid X_2), \\ & 0 \leq R_2 \leq I(X_2; Y \mid X_1), \\ & \text{for some} \quad P\,(x_1, x_2, y) = P(x_1, x_2)P^*(y \mid x_1, x_2)\} \,. \end{aligned} \tag{1.12}$$

Note that $\mathcal{B}_{so}$ is a convex region.

Shannon [1] proved that $\mathrm{B}_{si} \subset \mathcal{C}_{TWC} \subset \mathcal{B}_{so}$. For the BMC it follows from Shannon's outer bound that, for the maximal equal rate point, $R_1 = R_2 \leq 0.69424$ bit/transmission.

1.4 The dependence balance bound

In this section we prove the main result of this paper. We start by observing that for an $(N, M_1, M_2, P_{e1}, P_{e2})$ code

$$\begin{aligned}
\log(M_1) &= H(W_1 \mid W_2) \\
&\overset{(a)}{\leq} I(W_1; Y^N \mid W_2) + h(P_{e1}) + P_{e1} \cdot \log(M_1) \\
&\overset{(b)}{=} I(W_1; Y^N \mid W_2) + N\psi_1(N, M_1, P_{e1}) \\
&= \textstyle\sum_{n=1,N} I(W_1; Y_n \mid W_2, Y^{n-1}) + N\psi_1(N, M_1, P_{e1}) \\
&\overset{(c)}{=} \textstyle\sum_{n=1,N} I(W_1; X_{1n}; Y_n \mid W_2, X_{2n}, Y^{n-1}) \\
&\quad + N\psi_1(N, M_1, P_{e1}) \\
&\overset{(d)}{\leq} \textstyle\sum_{n=1,N} I(X_{1n}; Y_n \mid X_{2n}, Y^{n-1}) \\
&\quad + N\psi_1(N, M_1, P_{e1}),
\end{aligned} \tag{1.13a}$$

and

$$\log(M_2) \overset{(e)}{\leq} \textstyle\sum_{n=1,N} I(X_{2n}; Y_n \mid X_{1n}, Y^{n-1}) + N\psi_2(N, M_2, P_{e2}). \tag{1.13b}$$

Here (a) follows from Fano's result (see (5)); (b) if we define

$$\psi(N, M_1, P_{e1}) \triangleq (h(P_{e1}) + P_{e1} \cdot \log(M_1))/N;$$

(c) from the encoding functions (see (1)); (d) from the Markov relation $(W_1, W_2, Y^{n-1}) - (X_{1n}, X_{2n}) - Y_n$; and (e) if we define ψ_2 in an analogous way to ψ_1. Note that $A - B - C$ is used to express the fact that the random variables A, B, and C form a Markov chain in that order.

So far our derivation is rather standard. In fact, Shannon's outer bound follows almost immediately from (13). However, in Shannon's outer bound (see (12)) the input distribution may be arbitrary. We will show here that we can impose a restriction on this input distribution.

To see this, consider

$$
\begin{aligned}
0 &\leq I(W_1; W_2 \mid Y^n) \\
&\overset{(a)}{=} I(W_1; W_2 Y^N) - I(W_1; W_2) \\
&\overset{(b)}{=} -I_3(W_1; W_2; Y^N) \\
&\overset{(c)}{=} \textstyle\sum_{n=1,N} -I_3(W_1; W_2; Y_n \mid Y^{n-1}) \\
&\overset{(d)}{=} \textstyle\sum_{n=1,N}(-H(Y_n \mid Y^{n-1}) + H(Y_n \mid W_1, Y^{n-1}) \\
&\quad +H(Y_n \mid W_2, Y^{n-1}) - H(Y_n \mid W_1, W_2, Y^{n-1})) \\
&\overset{(e)}{=} \textstyle\sum_{n=1,N}(-H(Y_n \mid Y^{n-1}) + H(Y_n \mid W_1, Y^{n-1}, X_{1n} \\
&\quad +H(Y_n \mid W_2, Y^{n-1}, X_{2n}) \\
&\quad -H(Y_n \mid W_1, W_2, Y^{n-1}, X_{1n}, X_{2n})) \\
&\overset{(f)}{\leq} \textstyle\sum_{n=1,N}(-H(Y_n \mid Y^{n-1}) + H(Y_n \mid X_{1n}, X_{2n}, Y^{n-1})) \\
&\overset{(d)}{=} \textstyle\sum_{n=1,N} -I_3(X_{1n}; X_{2n}; Y_n \mid Y^{n-1} \\
&\overset{(b)}{=} \textstyle\sum_{n=1,N}(I(X_{1n}; X_{2n} \mid Y^{n-1})).
\end{aligned}
\tag{1.14}
$$

Now (a) follows from the independence of W_1 and W_2, (b) from the recursive property of K-information (see (10b)), (c) from the chaining property of K-information (see (10a)); (d) from the definition of K-information (see (8b) and (7)); (e) from (1); and (f) from the Markov relation $(W_1, W_2, Y^{n-1}) - (X_{1n}, X_{2n}) - Y_n$.

We can interpret inequality (14) as follows. If $I(X_{1n}; X_{2n} \mid Y^{n-1})$ is the dependence that is consumed in transmission n, then we can think of $I(X_{1n}; X_{2n} \mid Y_n, Y^{n-1})$ as the dependence that is produced in that transmission. Inequality (14) tells us that each code must produce the dependence it consumes or, in other words, must satisfy the *dependence balance*.

To see what input distributions are possible, first note that

$$
\begin{aligned}
P(y^{n-1} &, x_{1n}, x_{2n}, y_n) \\
&= P(y^{n-1}) P(x_{1n}, x_{2n} \mid y^{n-1}) P^*(y_n \mid x_{1n}, x_{2n}).
\end{aligned}
\tag{1.15}
$$

Define

$$
T \triangleq (S, Y^{S-1}) \quad X_1 \triangleq X_{1S} \quad X_2 \triangleq X_{2S} \quad Y \triangleq Y_S \tag{1.16}
$$

where S is a random variable that is uniformly distributed over $\{1, 2, \cdots, N\}$ and independent of $(W_1, W_2, X_1^N, X_2^N, Y^N)$. From (13)-(16) we

can conclude that, for each $(N, M_1, M_2, P_{e1}, P_{e2})$ code, there exists a distribution

$$P(t, s_1, x_2, y) = P(t, x_1, x_2)P^*(y \mid x_1, x_2) \tag{1.17a}$$

with

$$I(X_1; X_2 \mid T) \leq I(X_1; X_2 \mid Y, T) \tag{1.17b}$$

and such that

$$\frac{1}{N}.\log(M_1) \leq I(X_1; Y \mid X_2, T) + \psi_1(N, M_1, P_{e1}) \tag{1.17c}$$

$$\frac{1}{N} \cdot \log(M_2) \leq (X_2; Y \mid X_1, T) + \psi_2(N, M_2, P_{e2}). \tag{1.17d}$$

From the definition of the capacity region we know that for every achievable rate pair (R_1, R_2) and $\delta > 0$ there is an $(N, M_1, M_2, P_{e1}, P_{e2})$ code that satisfies (4). We now combine (4) with (17) and let $\delta \downarrow 0$. As a result $\psi_1 \downarrow 0$ and $\psi_2 \downarrow 0$, and we obtain the following theorem.

Theorem 1: For each single-output TWC $(\mathcal{X}_1 \times \mathcal{X}_2, P^*(y \mid x_1, x_2), \mathcal{Y})$ we have $\mathcal{C}_{TWC} \subset B_I$ where

$$\begin{aligned} B_I \triangleq \; & \{(R_1, R_2) : 0 \leq R_1 \leq I(X_1; Y \mid X_2, T), \\ & 0 \leq R_2 \leq I(X_2; Y \mid X_1, T), \\ & \text{for some} \quad P(t, x_1, x_2, y) = P(t, x_1, x_2)P^*(y \mid x_1, x_2) \\ & \text{such that} \quad I(X_1; X_2 \mid T) \leq I(X_1; X_2 \mid Y, T) \\ & \text{with} \quad \mid \mathcal{T} \mid \leq 3\}. \end{aligned} \tag{1.18}$$

The cardinality constraint on the time-sharing variable T follows from the support lemma (see Csiszár and Körner [7. p. 310]. Because of this constraint $\mathcal{B}_I$ is closed.

Note that $\mathcal{B}_I$ differs from Shannon's outer bound $\mathcal{B}_{so}$ only because of the restriction on the input distributions. The only (sets of) input distributions that are possible are those that do not consume more dependence (on the average) than they produce.

Applications:

a) An example: Consider the deterministic channel with $\mathcal{X}_1 = \mathcal{X}_2 = \{0,1,2\}$ and $\mathcal{Y} = \{0,1,2,3,4,5,6\}$. The relation between the inputs and the output is shown in Table 1. Clearly, for this channel, $I(X_1;X_2 \mid Y,T) = 0$ always. Therefore, $I(X_1;X_2 \mid T) = 0$, and thus the capacity region of this channel is Shannon's inner bound region. Note that the random variable T performs the convexification. Now $|\mathcal{T}| = 2$ suffices.

b) The BMC: For the symmetrical rate point of the BMC our new upper bound is useless. The reason is that the corresponding outer bound probability distribution $P(x_1,x_2,y)$ satisfies $I(X_1;X_2) \leq I(X_1;X_2 \mid Y)$.

Table 1.1: The output of the channel as a function of its inputs

X_1	0	1	2	X_2
0	0	1	1	
1	2	3	4	
2	2	5	6	Y

1.5 The parallel channel extension

A generalization of Theorem I can be obtained if we use a *parallel channel.* This parallel channel is assumed memoryless and has a finite output alphabet Z, inputs x_1, and x_2, and y, and a transition probability matrix $P^+(z \mid x_1,x_2,y)$. The idea behind introducing the parallel channel is to reduce the amount of dependence produced. For example, if we take $Z \equiv X_1$, the term $I(X_1,X_2 \mid Y,Z,T)$ is 0 and, consequently, only product input distributions are allowed. This can yield a better upper bound as we shall see. Before going into more detail, we shall derive our second bound.

Noting that $H(W_1 \mid Y^N,B^N,W_2) \leq H(W_1 \mid Y^N,W_2)$ and $H(W_2 \mid Y^N,B^N,W_1) \leq H(W_2 \mid Y^N,W_1)$ and proceeding along the lines of (13) and (14), we find that

$$\log(M_1) \leq \sum_{n=1,N} I(X_{1n}; Y_n, Z_n \mid X_{2n}, (Y^{n-1}, Z^{n-1})) + N\psi_1(N, M_1, P_{e1}), \tag{1.19a}$$

$$\log(M_2) \leq \sum_{n=1,N} I(X_{1n}; Y_n, Z_n \mid X_{2n}, (Y^{n-1}, Z^{n-1})) + N\psi_1(N, M_1, P_{e2}), \tag{1.19b}$$

and

$$0 \leq \sum_{n=1,N} (I(X_{1n}; X_{2n} \mid Y_n, Z_n, (Y^{n-1}, Z^{n-1})) - I(X_{1n}; X_{2n} \mid (Y^{n-1}, Z^{n-1}))). \tag{1.20}$$

From (13) and the Markov relation $Y^{n-1} - (X_{1n}, X_{2n}) - Y_n$, we obtain the following (Shannon outer bound) constraints:

$$\log(M_1) \leq \sum_{n=1,N} I(X_{1n}; Y_n \mid X_{2n}) + N\psi_1(N, M_1, P_{e1}) \tag{1.21a}$$

$$\log(M_2) \leq \sum_{n=1,N} I(X_{2n}; Y_n \mid X_{1n}) + N\psi_2(N, M_2, P_{e2}). \tag{1.21b}$$

Note that

$$\begin{aligned} P(& (y^{n-1}, z^{n-1}), x_{1n}, x_{2n}, y_n, z_n) \\ = & P(y^{n-1}, z^{n-1}) P(x_{1n}, x_{2n} \mid (y^{n-1}, z^{n-1}) \\ & \cdot P^*(y_n \mid x_{1n}, x_{2n}) P^+(z_n \mid x_{1n}, x_{2n}, y_n). \end{aligned} \tag{1.22}$$

Defining

$$T \triangleq (S, Y^{S-1}, Z^{S-1}) \; X_1 \triangleq X_{1S} \; X_2 \triangleq X_{2S} \; Y \triangleq Y_S \; Z \triangleq Z_S \tag{1.23}$$

where S is a random variable uniformly distributed over $\{1, 2, \cdots, N\}$ and independent of $(W_1, W_2, X_1^N, X_2^N, Y^N, Z^N)$, we obtain our next result.

Theorem 2: For the single-output TWC $(\mathcal{X}_1 \times \mathcal{X}_2, P^*(y \mid x_1, x_2), \mathcal{Y})$ and any DM parallel channel $(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}, P^+(z \mid x_1, x_2, y), \mathcal{Z})$ we have that $\mathcal{C}_{TWC} \subset \mathcal{B}_{II}$, where

$$\begin{aligned}\mathcal{B}_{II} = \{&(R_1, R_2) : 0 \le R_1 \le I(X_1; Y, Z \mid X_2, T)\\ &0 \le R_2 \le I(X_2; Y, Z \mid X_1, T)\\ &0 \le R_1 \le I(X_1; Y \mid X_2),\\ &0 \le R_2 \le I(X_2; Y \mid X_1),\\ &\text{for some} \quad P(t, x_1, x_2, y, z)\\ &= P(t, x_1, x_2) P^*(y \mid x_1, x_2) P^+(z \mid x_1, x_2, y)\\ &\text{such that} \quad I(X_1; X_2 \mid T) \le I(X_1; X_2 \mid Y, Z, T,)\\ &\text{with} \quad |\mathcal{T}| \le |\mathcal{X}_1| \cdot |\mathcal{X}_2| + 2\}.\end{aligned} \tag{1.24}$$

Again the cardinality constraint follows from the support lemma (see [7, p. 310]), and $\mathcal{B}_{II}$ is closed.

It follows from the definition of $\mathcal{B}_{II}$ that, to get a good upper bound, a parallel channel has to be chosen to reduce the dependence production $I(X_1; X_2 \mid Y, Z, T)$ without increasing the information rates $I(X_1; Y, Z \mid X_2, T)$ and $I(X_2; Y, Z \mid X_1 Y)$ too much.

Applications:
c) A corollary: Let the output Z of the parallel channel be equal to its input X_1. First we obtain that $I(X_1; X_2; \mid Y, Z, T) = 0$, and thus the only input distributions allowed are the ones for which $X_1 - T - X_2$. Now since $I(X_1; Y, Z \mid X_2, T) = H(X_1 \mid T)$ and $I(X_2; Y, Z \mid X_1, T = I(X_2; Y \mid X_1, T) \le I(X_2; Y \mid X_1)$, we obtain from Theorem 2 the following corollary.

Corollary: For the single-output TWC $(\mathcal{X}_1 \times \mathcal{X}_2, P^*(y \mid x_1, x_2), \mathcal{Y})$ we have that $\mathcal{C}_{TWC} \subset \mathcal{B}_{II}^{zbs}$ where

$$\begin{aligned}\mathcal{B}_{II}^{zbs} \triangleq \{&(R_1, R_2) : 0 \le R_1 \le H(X_1 \mid T),\\ &0 \le R_2 \le I(X_2; Y \mid X_1, T),\\ &0 \le R_1 \le I(X_1; Y \mid X_2),\\ &\text{for some} \quad P(t, x_1, x_2, y)\\ &= P(t) P(x_1 \mid t) P(x_2 \mid t) P^*(y \mid x_1, x_2)\\ &\text{with} \quad |\mathcal{T}| \le |\mathcal{X}_1| \cdot |\mathcal{X}_2| + 1\}.\end{aligned} \tag{1.25}$$

This bound could be called the Zhang *et al.* bound for single-output TWC's. Although it is obtained in an entirely different way, it is the same bound as Zhang *et al.* give in [6, theorem 3]. Numerical computation based on this Zhang *et al.* bound shows that 0.64891 bit/transmission is an upper bound for $R_1 = R_2$ of the maximal equal rate point of the BMC.

d) Another corollary: A single-output TWC for which there exists a mapping f from $\mathcal{X}_2 \times \mathcal{Y}$ into $\mathcal{X}_1$ such that $P^*(y \mid x_1, x_2) = 0$ if $x_1 \neq f(y, x_2)$, is said to be in *class* $\mathcal{D}_1$. For such a channel,

$$\begin{aligned} I(X_1; Y \mid X_2) &= H(X_1 \mid X_2) - H(X_1 \mid X_2, Y) \\ &= H(X_1 \mid X_2) - H(X_1 \mid X_2, Y, X_1 = f(Y, X_2)) \\ &= H(X_1 \mid X_2) \\ &\geq H(X_1 \mid X_2, T) \\ &= H(X_1 \mid T) \end{aligned} \tag{1.26}$$

where the last step follows of the Markov relation $X_1 - T - X_2$. From Corollary 1 and (26) we obtain the following upper bound for the capacity region of a TWC in class $\mathcal{D}_1$:

$$\begin{aligned} \mathcal{B}_{II}^d \triangleq \; & \{(R_1, R_2) : 0 \leq R_1 \leq H(X_1 \mid T), \\ & 0 \leq R_2 \leq I(X_2; Y \mid X_1, T), \\ & \text{for some} \quad P(t, x_1, x_2, y) \\ & = P(t)P(x_1 \mid t)P(x_2 \mid t)P^*(y \mid x_1, x_2) \\ & \text{with} \quad |\mathcal{T}| \leq 2\} . \end{aligned} \tag{1.27}$$

Next observe that for channels in class $\mathcal{D}_1$ the upper bound region $\mathcal{B}_{II}^d$ is equal to Shannon's inner bound region $\mathcal{B}_{si}$. Hence we have the next corollary.

Corollary 2: For a single-output TWC belonging to class $\mathcal{D}_1$ the capacity region $\mathcal{C}_{TWC} = \mathcal{B}_{si}$.

1.6 An adaptive parallel channel

The parallel channel in the previous section was a fixed channel. A good parallel channel simultaneously achieves low values of $I(X_1; Z \mid$

$Y, X_2, T)$, and $I(X_2; Z \mid Y, X_1, T)$, and of $I(X_1; X_2 \mid Y, Z, T)$. It is easy to see that even lower values of these mutual informations might be obtained if for each t the parameters of the parallel channel are allowed to depend on $P(x_1, x_2 \mid t)$ instead of being fixed as in the previous section. These lower values can then lead to an upper bound that is better than $\mathcal{B}_{II}$. This is shown in the present section.

First assume that the transition probabilities of the *adaptive* parallel channel are completely determined by the past outputs (y^{n-1}, z^{n-1}) of both the channel and its parallel channel. Then, clearly,

$$\begin{aligned} P(w_1, & \; w_2, x_1^n, x_2^n, y^n, z^n) \\ = & \; P(w_1)P(w_2)P(x_{11} \mid w_1)P(x_{21} \mid w_2) \\ & \cdot P^*(y_1 \mid x_{11}, x_{21})P^+(z_1 \mid x_{11}, x_{21}, y_1, \phi), \\ & \text{for } n = 1 \text{ where } \phi \text{ denotes "empty"} \\ = & \; P(w_1, w_2, x_1^{n-1}, x_2^{n-1}, (y^{n-1}, z^{n-1})) \\ & \cdot P(x_{1n} \mid w_1, y^{n-1})P(x_{2n} \mid w_2, y^{n-1}) \\ & \cdot P^*(y_n \mid x_{1n}, x_{2n})P^+(z_n \mid x_{1n}, x_{2n}, y_n, (y^{n-1}, z^{n-1})), \\ & \text{for } n = 2, \cdots, N. \end{aligned} \tag{1.28}$$

From (28) we obtain the Markov relation

$$(W_1, W_2) - \left((Y^{n-1}, Z^{n-1}), X_{1n}, X_{2n}\right) - (Y_n, Z_n). \tag{1.29}$$

We now specify how the past outputs determine the parameters of the parallel channel. First let $\Delta(\mathcal{V})$ be the set of all distributions of V and $\Delta(\mathcal{V} \mid \mathcal{W})$ the set of all conditional distributions of V given W. Now the mapping

$$F : \Delta(\mathcal{X}_1 \times \mathcal{X}_2) \to \Delta(\mathcal{Z} \mid \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}) \tag{1.30}$$

gives us for each (y^{n-1}, z^{n-1}) the transition probabilities

$$\begin{aligned} P^+_{z_n \mid x_{1n} x_{2n} y_n}(\cdot \mid \cdot, \cdot, \cdot, (\; & y^{n-1}, z^{n-1})) \\ & = F\left(P_{x_{1n}x_{2n}}(\cdot, \cdot \mid (y^{n-1}, z^{n-1}))\right). \end{aligned} \tag{1.31}$$

Noting that $P_{x_{1n}x_{2n}}(\cdot, \cdot \mid (y^{n-1}, z^{n-1}))$ is completely determined by (y^{n-1}, z^{n-1}), we conclude that the transition probabilities of the parallel channel are functions of (y^{n-1}, z^{n-1}).

We now return to the proof of Theorem 1. Noting that here (29) (instead of $(W_1, W_2, (Y^{n-1}, Z^{n-1})) - (X_{1n}, X_{2n}) - (Y_n, Z_n)$) expresses the Markov relation and proceeding along the lines of (13) and (14), we find that (19), (20), and (21) hold for adaptive parallel channels as well, where now

$$\begin{aligned} P((\ & y^{n-1}, z^{n-1}), x_{1n}, x_{2n}, y_n, z_n) \\ & = P(y^{n-1}, z^{n-1})\, P(x_{1n}, x_{2n} \mid (y^{n-1}, z^{n-1})) \\ & \cdot P^*(y_n \mid x_{1n}, x_{2n}) P^+(z_n \mid x_{1n}, x_{2n}, y_n, (y^{n-1}, z^{n-1}))\,. \end{aligned} \tag{1.32}$$

Define

$$\begin{aligned} T \triangleq (S, Y^{S-1}, \ & Z^{S-1}) \qquad X_1 \triangleq X_{1S} \qquad X_2 \triangleq X_{2S} \\ & Y \triangleq Y_S \qquad Z \triangleq Z_S \end{aligned} \tag{1.33}$$

where S is a random variable uniformly distributed over $\{1, 2, \cdots, N\}$ and independent of $(W_1, W_2, X_1^N, X_2^N, Y^N, Z^N)$. We can now deduce the following.

Theorem 3: For the single-output TWC $(\mathcal{X}_1 \times \mathcal{X}_2, P^*(y \mid x_1, x_2), \mathcal{Y})$ and any mapping $F : \Delta(\mathcal{X}_1 \times \mathcal{X}_2) \to \Delta(\mathcal{Z} \mid \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y})$, we have $\mathcal{C}_{TWC} \subset \mathcal{B}_{III}$ where

$$\begin{aligned} \mathcal{B}_{III} \triangleq \ & \mathrm{cl}\,(\{(R_1, R_2) : 0 \le R_1 \le I(X_1; Y, Z \mid X_2, T), \\ & 0 \le R_2 \le I(X_2; Y, Z \mid X_1, T), \\ & 0 \le R_1 \le I(X_1; Y \mid X_2), \\ & 0 \le R_2 \le I(X_2; Y \mid X_1), \\ & \text{for some} \quad P(t, x_1, x_2, y, x) \\ & = P(t, x_1, x_2) P^*(y \mid x_1, x_2) P^+(z \mid x_1, x_2, y, t) \\ & \text{such that for all } t \quad P^+_{z|x_1 x 2y}(. \mid \cdot, \cdot, \cdot, t) \\ & (= F(P_{x_1 x_2}(\cdot, \cdot \mid t)) \\ & \text{and such that} \quad I(X_1; X_2 \mid T) \le I(X_1; X_2 \mid Y, Z, T) \\ & \text{with} \quad |\mathcal{T}| \le |\mathcal{X}_1| \cdot |\mathcal{X}_2| + 3\})\,. \end{aligned} \tag{1.34}$$

Here cl denotes closure. This operation is necessary to guarantee that $\mathcal{B}_{III}$ is closed for *noncontinuous* mappings F as well. These noncontinuous mappings also require the cardinality constraint to be one more than in the fixed- parallel-channel case because Caratheodory's theorem (instead of the Fenchel-Eggleston theorem) must be applied to get

a support lemma (see Eggleston [11]). In the case where the mapping F is noncontinuous but the mutual information $I(X_1; Y, Z \mid X_2, T = t)$, $I(X_2; Y, Z \mid X_1, T = t)$, and $I(X_1; X_2 \mid Y, Z, T = t)$ still are continuous functions of $P(x_1, x_2 \mid T = t)$, and also in the case where the mapping F is continuous, the constraint again becomes $|\mathcal{T}| \leq |\mathcal{X}_1| \cdot |\mathcal{X}_2| + 2$ and the closing operation is superfluous.

Applications:

e) A corollary: We next consider an extension to Corollary 2. We say a channel is in *class* $\mathcal{D}_2$ if it is a single-output TWC for which there exist a finite alphabet $\mathcal{U}$ and two mappings $f_1 : \mathcal{Y} \times \mathcal{X}_2 \to \mathcal{U}$ and $f_2 : \mathcal{Y} \times \mathcal{X}_1 \to \mathcal{U}$ such that $P^*(y \mid x_1, x_2) = 0$ if $f_1(y, x_1) \neq f_2(y, x_2)$, and for which $I(X_1; Y_2) = 0$ implies that $I(X_1; X_2 \mid Y, U) = 0$ where U is the random variable which is equal to $f_1(Y, X_1) = f_2(Y, X_2)$ with probability one. From this definition we can see that channels in class $\mathcal{D}_2$ have an *implicit extra output* U and that this extra output together with Y cannot generate any dependence from independent input assignments. We shall use Theorem 3 to show that for these channels Shannon's inner bound region is the capacity region. An alternative proof of this result can be found in [12].

For our proof we need a wringing lemma. The lemma is stated below, and its proof can be found in Appendix 1.

Wringing Lemma: Let $\epsilon \geq 0$. If A and B are two discrete random variables with $I(A; B) \leq \epsilon$, then a third discrete random variable C exists such that $I(A; B \mid C) = 0$ and $H(C) \leq \Theta(\epsilon)$. The function $\Theta(\cdot)$ depends on $|\mathcal{A}|$ and $|\mathcal{B}|$ only, is concave, and $\lim_{\epsilon \downarrow 0} \Theta(\epsilon) = 0$.

We can now specify the mapping F by defining the random variable Z as a function of $P(x_1, x_2 \mid t)$:

$$Z \triangleq (U, C_t) \tag{1.35}$$

where C_t is a random variable for which $I(X_1; X_2 \mid Y, U, C_1, t) = 0$ with entropy $H(C_t \mid Y, U, t) \leq \Theta(\epsilon)$ when $I(X_1; X_2 \mid Y, U, t) \leq \epsilon$. In addition, $\lim_{\downarrow 0} \Theta(\epsilon) = 0$. The existence of this random variable is guaranteed by a conditional version of the wringing lemma, which holds because of the concavity of $\Theta(.)$. Because of (35) we conclude that $I(X_1; X_2 \mid Y, Z, T) = I(X_1; X_2 \mid Y, U, C_T, T) = 0$. It now follows from

the dependence balance that we may restrict ourselves to probability distributions for which $P(x_1, x_2 \mid t) = P(x_1 \mid t)P(x_2 \mid t)$. Since in this case $H(C_T \mid Y, U, T) = 0$, we find that

$$\begin{aligned} I(X_1; Y, Z \mid X_2, T) &= I(X_1; Y, U, C_T \mid X_2, T) \\ &\leq I(X_1; Y, U \mid X_2, T) + H(C_T \mid Y, U, T) \\ &= I(X_1; Y, U \mid X_2, T) \\ &= I(X_1; Y \mid X_2, T) + I(X_1; U \mid Y, X_2, T) \\ &= I(X_1; Y \mid X_2, T) \end{aligned} \tag{1.36a}$$

$$\begin{aligned} I(X_1; Y \mid X_2, T) &= H(Y \mid X_2, T) - H(Y \mid X_1, X_2, T) \\ &\leq H(Y \mid X_2) - H(Y \mid X_1, X_2) \\ &= I(X_1; Y \mid X_2), \end{aligned} \tag{1.36b}$$

and in a similar way,

$$I(X_2; Y, Z \mid X_1, T) = I(X_2; Y \mid X_1, T) \leq I(X_2; Y \mid X_1). \tag{1.36c}$$

Thus we obtain the following corollary.

Corollary 3: For a single-output TWC $(\mathcal{X}_1 \times \mathcal{X}_2, P^*(y \mid x_1, x_2), \mathcal{Y})$ in class $\mathcal{D}_2$ the capacity region $\mathcal{C}_{TWC} = \mathcal{B}_{si}$.

f) An example: Consider the deterministic channel with $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y} = \{0, 1, 2\}$. The relation between the input and the output of the channel is given in Table 2.

Table 1.2: The output of the channel as a function of its inputs

X_1	0	1	2	X_2
0	0	1	2	
1	1	0	2	
2	1	2	0	Y

Defining the mappings f_1 and f_2 as in Table 3, we obtain the configuration in Table 4 for the channel and parallel channel. This channel is equivalent to the channel in Table 1. Just like the channel in

Table 1, it cannot create dependence between the inputs, and thus $I(X_1; X_2 \mid Y, U) = 0$ always. Therefore, the channel in Table 2 belongs to class $\mathcal{D}_2$ and, because of corollary 3, its capacity region is Shannon's inner bound region. In Appendix II it is shown that the Zhang *et al.* bound (and therefore Shannon's outer bound as well) yields rate points outside Shannon's inner bound for the channel of Table 2. From this and the fact that the Zhang *et al.* bound follows from Theorem 3, we may conclude that for single-output TWC's our Theorem 3 is in general stronger than the Zhang *et al.* bound and Shannon's outer bound.

Table 1.3: The mappings f_1 and f_2

			X_1					X_2	
	0	1	2	Y	0	1	2	Y
0	0	2	–		0	1	–	
1	1	1	2		1	2	1	
2	2	1	1	$f_1(Y, X_1)$	2	2	2	$f_2(Y, X_2)$

Table 1.4: The outputs of both channels as a function of their inputs

X_1	0	1	2	X_2
0	0,0	1,2	1,2	
1	1,1	0,1	2,2	
2	1,1	2,1	0,2	Y, U

g) Another example: The preceding example demonstrated the use of an implicit output of the single-output TWC to obtain a better upper bound for its capacity region. With the channel in Table 5 we give an example of a channel for which $I(X_1; X_2) = 0$ implies that $I(X_1; Y_2 \mid Y) = 0$. Note that now the channel has no implicit output. From the table we see that different values of y divide up the $\mathcal{X}_1 \times \mathcal{X}_2$ space in *independent structures* (rectangles). Hence for a product input distribution, for each output y, the output distribution is again

a product distribution. Therefore, this channel is in class $\mathcal{D}_2$ and its capacity region is Shannon's inner bound region.

Table 1.5: A channel for which $I(X_1; X_2) = 0$ implies that $I(X_1; X_2 \mid Y) = 0$

X_1	0	1	2	3	X_2
0	0	0	3	3	
1	1	2	3	3	
2	4	4	4	5	
3	4	4	4	5	Y

h) The BMC: Here we show that 0.64628 bit/transmission is an upper bound for $R_1 = R_2$ of the maximal equal rate point of the BMC. With this result we improve upon the Zhang *et al.* bound [7] (0.64891). While in the Zhang *et al.* bound Z was chosen equal to X_1 in Corollary 1 (and equal to X_2 in its symmetric counterpart), we choose a Z which is less "informative" than X_1 or X_2. However, we still require that

$$X_1 - (Y, Z, T) - X_2. \tag{1.37}$$

The results in the Z-channel are given in Table 6 ($0 \leq p \leq 1$).

Table 1.6: A Z-channel for the BMC

(X_1, X_2, Y)	0	1	2	Z
(0,0,0)	0	p	$1-p$	
(0,1,0)	0	1	0	
(1,0,0)	0	0	1	
(1,1,1)	1	0	0	$P^*(z \mid x_1, x_2, y)$

We can easily verify that $I(X_1; X_2 \mid Y, Z, T) = 0$ since $Z = 0$ implies that $X_1 = X_2 = 1$, $Z = 1$ implies that $X_1 = 0$ and $Z = 2$ implies that $X_2 = 0$. Thus (37) holds, and we may restrict ourselves to input distributions of the product type.

If $p = 0$, then Z completely determines X_2 (and nothing more than that). On the other hand, if $p = 1$ then Z completely determines X_1 (and nothing more). Hence choosing $p \equiv 0$ or $p \equiv 1$ gives us the Zhang *et al.* bounds. The question is whether there is a better choice for p. To answer this question, note that

$$A \triangleq I(X_1; Z \mid Y, X_2, T = t) + I(X_2; Z \mid Y, X_1, T = t) \tag{1.38}$$

is the sum of the *information leak quantities* for $T = t$. We can therefore choose p_{opt} so that it minimizes A. Expressing A as a function of p we obtain

$$A = \Phi_2 h(\Phi_1 p) + \Phi_1 h(\Phi_2(1-p)) - 2\Phi_1\Phi_2 h(p) \tag{1.39}$$

where $\Phi_1 \triangleq P(X_1 = 0 \mid T = t)$ and $\Phi_2 \triangleq P(X_2 = 0 \mid T = t)$. Since (38) (and therefore also (39)) is convex-$\cup$ in p, we deduce via differentiation of A that

$$\begin{aligned} P_{opt} &= \frac{\Phi_1(1-\Phi_2)}{\Phi_1(1-\Phi_2)+(1-\Phi_1)\Phi_2} \\ &= \frac{P(x_1=0|t)P(x_2=1|t)}{P(x_1=0|t)P(x_2=1|t)+P(x_1=1|t)P(x_2=0|t)} \end{aligned} \tag{1.40}$$

minimizes A if the denominator of (40) is positive. It turns out that if the denominator of (40) equals $0, A = 0$ and p_{opt} may be chosen arbitrarily.

The mapping F is now defined by (40) and Table 5. Note that the mapping is not continuous in $P_{x_1,x_2|t}$. However, $I(X_1; Y, Z \mid X_2, T)$ and $I(X_2; Y, Z \mid X_1, T)$ are. Therefore, it suffices to take $|\mathcal{T}| \leq 5$. Computer evaluation of the resulting bound shows that *no more than 0.64628 bit/transmission are achievable for the BMC.*

1.7 Results for the multiple access channel with feedback

The close relationship between the single-output TWC and the MAC with feedback is obvious. Therefore, it is not surprising that Theorems 1-3 hold for the MAC with feedback if we add the constraint

$$R_1 + R_2 \leq I(X_1; X_2; Y) \tag{1.41}$$

and adjust the cardinality bounds for T.

The MAC variant of Corollary 2 tells us that the feedback capacity region for MAC's in class $\mathcal{D}_1$ equals the achievable region found by Cover and Leung [13], a result previously obtained by Willems [14] via a different approach. The MAC variant of Corollary 3 can be thought of as an extension of Willems' result. It states that for MAC's in class $\mathcal{D}_2$ the Cover-Leung region is again the feedback capacity region. Note that class $\mathcal{D}_1$ is strictly inside class $\mathcal{D}_2$.

1.8 Conclusion

In this paper we derived upper bounds for the capacity region of the general single-output TWC. Our basic idea was that each code must produce the dependence (measured in terms of mutual information) it consumes. We introduced a parallel channel to decrease the dependence production without introducing excessively large information leaks. Finally, we assumed that the transition probabilities of the parallel channel are adaptable to the actual input distribution. These methods yielded a number of results. First, we were able to show that, for channels in a certain class, the capacity region is Shannon's inner bound region. Roughly speaking, this class contains channels that cannot produce "real" dependence when the inputs are *a priori* independent. Secondly, our upper bound improves upon the Zhang *et al.* upper bound. We show that, for the BMC, rate pairs with $R_1 = R_2 > 0.64628$ bit/transmission cannot be achieved. This is still far from Schalkwijk's best achievable rate point for which $R_1 = R_2 = 0.63056$ bit/transmission.

For the MAC with feedback we can derive analogous bounds. For MAC's in class $\mathcal{D}_2$ we determined the feedback capacity region. This generalizes Willems [14] result. We finally note that more detailed derivations of the results in this paper can be found in Hekstra's master thesis [15].

1.9 Acknowledgement

We thank both reviewers for their valuable comments.

Appendix A

The proof of the wringing lemma

Without loss of generality, we assume that $P(a) \neq 0$ and $P(b) \neq 0$. We use the notation $P(a,b) := Pr\{A = a, B = b\}, P(a) := Pr\{A = a\}$ and $P(b) := Pr\{B = b\}$. Let $h(.)$ denote the binary entropy function and assume that all logarithms in this Appendix are natural.

In our proof we construct a random variable C that will satisfy the conditions in the lemma. First we define

$$\lambda := \min_{(a,b)} \frac{P(a,b)}{P(a)P(b)}. \tag{A.1}$$

Let the pair (a', b') achieve this minimum. Note that $\lambda \leq 1$ or, equivalently, $P(a', b') \leq P(a')P(b')$. Assume that $P(a', b') < P(a')P(b')$. Since for all (a, b) the inequality $P(a, b) \geq \lambda P(a)P(b)$ holds, we obtain that

$$\frac{Pr\{A \neq a', B \neq b'\}}{Pr\{A \neq a'\}Pr\{B \neq b'\}} \geq \lambda = \frac{P(a', b')}{P(a')P(b')}. \tag{A.2}$$

The fact that $P(a', b') < P(a')P(b')$, $Pr\{A \neq a', B \neq b'\} = 1 - P(a') - P(b') + P(a', b')$ together with (A2) now yields that

$$P(a') + P(b') \leq 1 \tag{A.3a}$$

and consequently

$$P(a')P(b') \leq 1/4 \tag{A.3b}$$

If $\lambda = 1$ then A and B are independent. Without loss of generality we may assume that $P(a') \leq P(b')$.

The log-sum inequality (see Csiszár and Körnmer [7, p. 481]) along with a lower bound for the divergence in terms of variational distance [7, p. 58] yield

$$\begin{aligned} \epsilon \geq & \quad I(A;B) \\ = & \quad P(a',b') \cdot \log\left(\tfrac{P(a',b')}{P(a')P(b')}\right) \\ + & \quad \textstyle\sum_{(a,b)\neq(a',b')} P(a,b) \cdot \log\left(\tfrac{P(a,b)}{P(a)P(b)}\right) \\ \geq & \quad P(a',b') \cdot \log\left(\tfrac{P(a',b')}{P(a')P(b')}\right) \\ & \quad +(1-P(a',b')) \cdot \log\left(\tfrac{1-P(a',b')}{1-P(a')P(b')}\right) \\ \geq & \quad 2(P(a')P(b') - P(a',b'))^2 \\ \geq & \quad P(a')^2 P(b')^2 (1-\lambda)^2. \end{aligned} \tag{A.4}$$

We can now distinguish between two cases.

a) Assume first that $P(a')P(b') \geq (1-\lambda)$. Then we obtain from (A3b) that $1-\lambda \leq 1/4$. Furthermore, (A4) implies that $\epsilon \geq (1-\lambda)^4$. We therefore conclude that $1-\lambda \leq \min\{1/4, \epsilon^{1/4}\}$. Let $P(a,b) = \lambda P(a)P(b) + (1-\lambda)Q(a,b)$ where $Q(a,b)$ is some probability distribution. We introduce the random variable D which takes values in the set $\{*\} \subset \mathcal{A} \times \mathcal{B}$. We define for each (a,b)

$$Pr\{A=a, B=b, D=d\} := \begin{cases} \lambda P(a)P(b), & \text{if } d = * \\ (1-\lambda)Q(a,b), & \text{if } d = (a,b) \\ 0, & \text{otherwise} \end{cases} \tag{A.5}$$

Observe that $I(A;B \mid D) = 0$ and that

$$H(D) \leq h(\min\{1/4, \epsilon^{1/4}\}) + \min\{1/4, \epsilon^{1/4}\} \cdot \log(|\mathcal{A}| \cdot |\mathcal{B}|). \tag{A.6}$$

b) Now assume that $(1-\lambda) > P(a')P(b')$. In this case (A4) implies that $\epsilon > P(a')^4 P(b')^4$. From $P(a') \leq P(b')$ it follows that $P(a') < \epsilon^{1/8}$, and from (A3a) we find that $P(a') \leq 1/2$. We conclude that $P(a') \leq \min\{1/2, \epsilon^{1/8}\}$. Next we introduce the random variable $D \in \{*, a'\}$is equal to * if $A \neq a'$ and equal to a' if $A = a'$. Then

$$H(D) \leq h(\min\{1/2, \epsilon^{1/8}\}), \tag{A.7}$$

and furthermore,

$$\begin{aligned} \epsilon &\geq I(A;B) = I(A,D;B) = I(D;B) + I(A;B \mid D) \\ &\geq I(A;B \mid D) = (1 - P(a')) \cdot I(A;B \mid D = *) \\ &\geq (1/2).I(A;B \mid D = *). \end{aligned} \tag{A.8}$$

Hence

$$I(A;B \mid D = *) \leq 2\epsilon, \tag{A.9}$$

and it appears that we have transformed the original problem into an equivalent one. Now $I(A;B \mid D = *) \leq 2\epsilon$, and we have to construct a random variable C with $H(C \mid D = *) \leq \theta(\epsilon) - H(D)$ such that $I(A;B \mid C, D = *) = 0$, etc.

Alternatives a) and b) suggest an iterated construction of C. Starting with the original distribution we have to perform step b) as long as

$$\begin{aligned} Pr\{ & A = a', B = b' \mid D_1 = *, \cdots, D_m = *\} \\ < & Pr\{A = a' \mid D_1 = *, \cdots, D_m = *\} \\ & \cdot Pr\{B = b' \mid D_1 =, \cdots, D_m = *\} \\ & \cdot (1 - Pr\{A = a' \mid D_1 = *, \cdots, D_m = *\} \\ & \cdot Pr\{B = b' \mid D_1 = *, \cdots, D_m = *\}). \end{aligned} \tag{A.10}$$

Here D_m is the random variable that is constructed during iteration m. Note, furthermore, that $\epsilon_m = 2^{m-1} \cdot \epsilon$. It is impossible to satisfy (A10) for $m \geq |\mathcal{A}| + |\mathcal{B}| - 3$. In this case either $Pr\{A = a', B = b' \mid D_1 = *, \cdots, D_m = *\} = Pr\{A = a' \mid D_1 = *, \cdots, D_m = *\}$ or $Pr\{A = a', B = b' \mid D_1 = *, \cdots, D_m = *\} = Pr\{B = b' \mid D_1 = *, \cdots, D_m = *\}$. After having performed step b) $M \leq |\mathcal{A}| + |\mathcal{B}| - 3$ times, we perform step a) once. Now $\epsilon_{M+1} = 2^M \cdot \epsilon$, and the random variable D_{M+1} is constructed. We can finally compose the (discrete) random variable C as follows:

$$C := (D_1, \cdots, D_M, D_{M+1}). \tag{A.11}$$

Note that

$$\begin{aligned} I(A;B \mid C) &= I(A;B \mid D_1, \cdots, D_M, D_{M+1}) \\ &= Pr\{D_1 = *, \cdots, D_M = *, D_{M+1} = *\} \\ &\quad \cdot I(A;B \mid D_1 = *, \cdots, D_M = *, D_{M+1} = *) \\ &= 0 \end{aligned} \tag{A.12}$$

and that

$$\begin{aligned} H(C) &\leq \textstyle\sum_{m=1,M} H(D_m) + H(D_{M+1}) \\ &\leq \textstyle\sum_{m=1,M} h(\min\{1/2, \epsilon_m^{1/8}\}) + h(\min\{1/4, \epsilon_{M+1}^{1/4}\}) \\ &\quad + \min\{1/4, \epsilon_{M+1}^{1/8}\} \cdot \log(|\mathcal{A}| \cdot |\mathcal{B}|) \\ &\leq \textstyle\sum_{m=1,M} h(\min\{1/2, (2^{M-1}.\epsilon)^{1/8}\}) \\ &\quad + h(\min\{1/4, (2^M.\epsilon)^{1/4}\}) \\ &\quad + \min\{1/4, (2^M.\epsilon)^{1/4}\} \cdot \log(|\mathcal{A}| \cdot |\mathcal{B}|) \\ &\quad \leq (|\mathcal{A}| + |\mathcal{B}| - 2) \cdot h(\delta) + \delta \cdot \log(|\mathcal{A}| \cdot |\mathcal{B}|) := \Theta(\epsilon) \end{aligned} \tag{A.13}$$

where $\delta := \min\{1/2, (2^{|\mathcal{A}|+|\mathcal{B}|-3} \cdot \epsilon)^{1/8}\}$. We conclude our proof with the observation that $\Theta(.)$ is concave and right-continuous at zero, $\Theta(0) = 0$. Together with (A13) this proves our lemma.

We remark that the wringing lemma stated here is not the first of its kind in information theory. Wringing techniques were introduced a few years ago by Dueck [16] and Ahlswede [17], [18] for weakly dependent sequences.

Appendix B

The Zhang *et al.* bound is not tight

Consider the channel in Table 2. By Corollary 3 the capacity region of this channel is Shannon's inner bound region. We here derive an upper bound for $R_1 + R_2$ of the inner bound rate pairs. Without loss of generality, let

$$\begin{array}{ll} P(x_1 = 0) \triangleq 1 - \alpha_1 & P(x_1 = 1) \triangleq \alpha_1(1 - \beta_1) \\ P(x_1 = 2) \triangleq \alpha_1\beta_1 & P(x_2 = 0) \triangleq 1 - \alpha_2 \\ P(x_2 = 1) \triangleq \alpha_2(1 - \beta_2) & P(x_2 = 2) \triangleq \alpha_2\beta_2 \end{array} \tag{B.1}$$

for some $0 \leq \alpha_1, \alpha_2, \beta_1, \beta_2 \leq 1$. In terms of α's and β's we can express the sum of the rate constraints C as

$$\begin{aligned} C &= I(X_1; Y \mid X_2) + I(X_2; Y \mid X_1) \\ &= h(\alpha_1) + \alpha_1\alpha_2 h(\beta_1) + h(\alpha_2) + \alpha_1\alpha_2 h(\beta_2) \\ &\leq h(\alpha_1) + h(\alpha_2) + 2\alpha_1\alpha_2 \end{aligned} \tag{B.2}$$

where the last inequality follows from the fact that $h(x) \leq 1$ bit for $0 \leq \times \leq 1$. Note that in this Appendix the base of the logarithms is two. Defining

$$\alpha \triangleq \frac{\alpha_1 + \alpha_2}{2}, \tag{B.3}$$

then $0 \leq \alpha \leq 1$ and

$$C \leq 2h(\alpha) + 2\alpha^2. \tag{B.4}$$

Numerical evaluation shows that $C \leq 2.74885$ bit/transmission. Now consider the Zhang /it et al. bound for this channel. The assignment

$$\begin{array}{ll}
P(t=0) = 0.86505 & P(t=1) = 0.13495 \\
P(x_1 = 0 \mid t=0) = 0.17364 & P(x_1 = 1 \mid t=0) = P(x_1 = 2 \mid t=0) \\
 & = 0.41318 \\
P(x_2 = 0 \mid t=0) = 0.25168 & P(x_2 = 1 \mid t=0) = P(x_2 = 2 \mid t=0) \\
 & = 0.37416 \\
P(x_1 = 0 \mid t=1) = 0.87568 & P(x_1 = 1 \mid t=1) = P(x_1 = 2 \mid t=1) \\
 & = 0.06216 \\
P(x_2 = 0 \mid t=1) = 0.41200 & P(x_2 = 1 \mid t=1) = P(x_2 = 2 \mid t=1) \\
 & = 0.29400
\end{array} \tag{B.5}$$

yields

$$R_1 + R_2 = 2.76160 \quad \text{bit/transmission} \tag{B.6}$$

From the above we conclude that the Zhang *et al.* bound is not tight for the channel in Table 2. Since the Zhang *et al.* bound improves upon Shannon's outer bound, the latter bound cannot be tight either. It is Corollary 3 that gives us the capacity region for this channel.

Bibliography

[1] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley, Symp. Math. Statist. and Prob.*, 1961, pp. 611-644 (reprinted in *Key Papers in the Development of Information Theory*), D. Slepian, Ed. New York: IEEE Press, (1974, pp. 339-372).

[2] G. Dueck, "The capacity region of the two-way channel can exceed the inner bound," *Inform. Contr.*, vol. 40, pp. 258-266, Mar. 1979.

[3] J.P.M. Schalkwijk, "The binary -multiplying channel - A coding scheme that operates beyond Shannon's inner bound," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 107-110, Jan. 1982.

[4] —, "On an extension of an achievable rate region for the binary multiplying channel," *IEEE Trans. Inform. Theoty*, vol. IT-29, pp. 445-448, May 1983.

[5] T.S. Han, "A general coding scheme for the two-way channel," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 35-44, Jan. 1984.

[6] Z. Zhang, T. Berger, and J. P. M. Schalkwijk, "New outer bounds to capacity regions of two-way channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 383-386, May 1986.

[7] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Budapest, Hungary: Akademiai Kiado, 1981.

[8] W. J. McCill, "Multivariate information transmission," *Psichometrika*, vol. 19, pp. 97-116, 1954.

[9] R. M. Fano, *The Transmission of Information: A Statistical Theory of Communication.* Cambridge, MA: MIT Press, 1961.

[10] T.S. Han, "Multiple mutual informations and multiple interactions in frequency data," *Inform. Contr.*, vol. 46, no. 1, pp. 26-45, 1980.

[11] H.G. Eggleston, *Convexity.* New York: Cambridge Univ. Press, 1963.

[12] A. P. Hekstra and F.M.J. Willems, "Capacity regions for multiple access channels with feedback and two-way channels," in *Verhandelingen 5de Symp. Inform. Theorie Benelux*, Aalten, May 24-25, 1984, pp. 73-79.

[13] T. M. Cover and C. S. K. Leung, "An achievable rate region for the multiple-access channel with feedback," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 292-298, May 1981

[14] F. M. J. Willems, "The feedback capacity region of a class of discrete memoryless multiple access channels." *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 93-95, Jan. 1982.

[15] A. P. Hekstra, "Dependence balance outer bounds for the equal output two-way channel and the multiple access channel with feedback," graduate paper, Elec. Eng. Dept., Eindhoven Univ. of Technology, Eindhoven, The Netherlands, May 1985.

[16] G. Dueck, "A strong converse to the coding theorem for the multiple access channel," *J. Comb., Inf., Syst. Sci.*, vol. 6, pp. 187-196, 1981.

[17] R. Ahlswede, "An elementary proof of the strong converse theorem for the multiple-access channel." *J.Comb., Inf., Syst. Sci.*, vol. 7, pp. 216-230, 1982.

[18] R. Ahlswede, "Multiple descriptions without excess rate," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 721-726, Nov. 1985.

Chapter 2

Asymptotic component densities in programmable gate arrays realizing all circuits of a given size

Abstract

A Programmable Gate Array (PGA) is modeled as a square grid. Some grid nodes are processing nodes containing electrical elements. The rest are switching nodes capable of connecting wires incident on them. Two possible types of switching nodes are considered. In vertex connectivity each switching node can connect only one pair of wires. In edge connectivity each switching node can simultaneously connect two pairs of wires. The PGA must be capable of implementing any graph of size at most k and degree at most four. We prove tight bounds on the highest achievable density of processing nodes.

In edge connectivity the highest achievable density is $\Theta(1/k)$. In vertex connectivity the highest achievable density is $\Theta(1/k^2)$. If the grid is augmented by the diagonal edges then the highest achievable

[0]Co-authored with T. Berger from Cornell University, Ithaca, USA, and A. Orlitsky from AT&T Bell Labs., Murray Hill, USA. Published in Algorithmica, vol. 9, nr. 2, pp. 101-127, Feb. 1993.

density is $\Theta(1/k)$ even with vertex connectivity. These extend known results for embedding graphs in grids.

Small graphs of degree one are further examined. For $k = 2$ and $k = 3$ the highest density of processing nodes equals the highest density of parked cars in a square parking lot where each car can exit. Both densities are $2/3$. For $k = 4$ the highest density is $1/2$.

2.1 Introduction

A *Programmable Gate Array (PGA)* consists of many components placed on a wafer and of wires capable of connecting them. By selecting a subset of the components and connecting the appropriate wires, various circuits can be implemented. A PGA thereby combines the versatility of a printed circuit board with the speed of a single-wafer VLSI chip.

Ideally, a PGA would have a wire between any two components. Then every circuit could be implemented by connecting the appropriate wire ends to the components. However, the number of wires in such a PGA would grow as the square of the number of components.

Therefore, a compromise usually is adopted whereby the PGA resembles a graph we call the *PGA graph.* The wires are the edges of the PGA graph; the points where they can be connected are the nodes. Some of the nodes contain the electrical elements; these are the *processing nodes.* Each processing node can connect the electrical element it contains to any of the wires incident upon it, but cannot directly connect two wires[1]. The other nodes are *switching nodes.* They can connect pairs of wires incident upon them. We distinguish between two assumptions concerning the connection capabilities of the switching nodes:

Vertex Connectivity — Each switching node can connect only one pair of wires.

Edge Connectivity — Each switching node acts as a crossbar and can simultaneously connect several pairs of wires.

[1]Another model where processing nodes can connect wires is discussed in Section 2.7.

These names were chosen in part because in vertex connectivity the switching nodes can be regarded as connecting the two wires to the node (vertex), while in edge connectivity the switching nodes can be regarded as connecting pairs of wires (edges) directly; however, the main reason for adopting these names will become clear in the next section. In circuits implemented by PGA's with vertex connectivity, the wires correspond to vertex-disjoint paths in the PGA graph whereas in circuits implemented by PGA's with edge connectivity, the wires correspond to edge-disjoint paths. Edge connectivity clearly requires a more advanced technology. However, one of our main conclusions is that edge connectivity allows for a significant increase in the number of components the PGA can contain.

Remark: Previous works on embedding graphs in grids considered one-layer graph embeddings and two-layer graph embeddings. One-layer graph embedding is equivalent to vertex connectivity. Two-layer graph embedding is similar to edge connectivity except that wires can run in two layers, hence the two are equivalent if we ignore constant factors pf the number of wires. In current PGA's, cf. [7] and [5], the only freedom afforded the user is to determine switching-node connections; hence we use vertex and edge connectivities.

Circuits, too, can be modeled as graphs, called *circuit graphs*. The vertices correspond to the electrical elements of the circuit and the edges represent the connections. For a PGA to *implement* a circuit, the processing and switching nodes are reconfigured by the user. Two processing nodes are connected by a chain of PGA wires if and only if the corresponding electrical elements are connected in the circuit. We assume that each electrical element of the circuit corresponds to a unique component of the PGA. This is the case, for example, if each type of electrical element appears in the PGA exactly once[2].

Although any graph can serve as a PGA graph, we shall consider only the standard *two-dimensional square grid*. The grid is popular, among other reasons, for its regularity, its high connectivity, and its simple layout. Figure 2.1 illustrates a PGA. The processing nodes are numbered and the switching nodes are drawn empty.

[2]This assumption constitutes a major departure from most real PGA's. For further discussion, see Section 2.7.

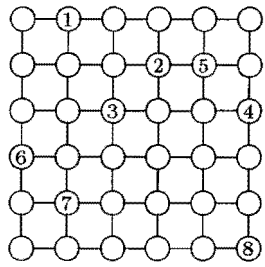


Figure 2.1: A 6 by 6 grid PGA with 8 processing nodes.

Example 1 *Assume that the PGA graph depicted in Figure 2.1 has to embed the circuit graph* $1-2$, $3-4$, $5-6$, *where a dash represents a required connection. With vertex connectivity we can connect the wires as shown by the thick lines in Figure 2.2(a). With edge connectivity we can either use the same paths or the simpler paths shown in Figure 2.2(b). Note that the path form 3 to 4 crosses the path from 5 to 6 at a switching node. This is permissible in edge-connectivity PGA's. If the circuit contained the additional wire* $7-8$, *it could not have been implemented with vertex-connectivity but could have been implemented with edge connectivity.* □

It is desirable to have many processing nodes in the PGA. This, however, limits the circuits that can be implemented, both because there are fewer switching nodes left for connecting the wires and because, if the processing nodes are too close together, not all circuits using these nodes can be implemented[3]. It is this tradeoff between density and connectivity that we shall investigate. For a given k, we want to determine the largest proportion of grid nodes that can be used as processing nodes subject to the requirement that every k-element circuit using these nodes can be implemented.

[3]We shall see later that this is the more dominant of the two reasons. Hence, for large PGA's there is a diminishing difference between PGA's with processing nodes that can connect wires and PGA's where the processing nodes cannot connect wires. See Section 2.7.

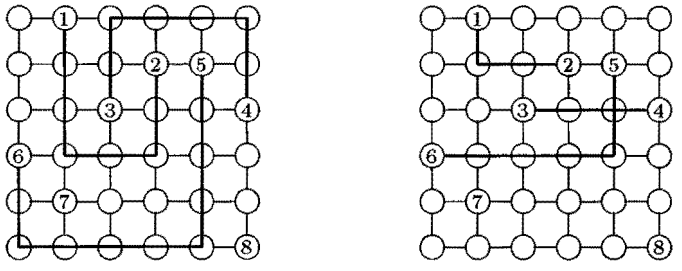


Figure 2.2: Embedding a circuit graph in a PGA graph.

We state the problem formally in the next section. We define $\nu_{d,k}$ to be the largest proportion of grid nodes that can serve as processing nodes in a vertex-connectivity PGA if every circuit of degree d and size k whose elements are among the processing nodes can be implemented by the PGA. We define $\epsilon_{d,k}$ similarly for edge-connectivity PGA's.

We assume that each edge of the PGA graph represents one actual wire. Since every grid node has degree at most four, the PGA cannot implement circuits of degree larger than four[4]. Therefore $\epsilon_{d,k} = 0$ for $d \geq 5$.

Recall that when a PGA implements a circuit, each circuit wire is mapped into a sequence of wires, or path, in the PGA. In a vertex-connectivity PGA these paths cannot intersect at any node, so the crossing number of the circuit graph cannot exceed the crossing number of the PGA graph. Grids are planar graphs; hence, vertex-connectivity grid PGA's can implement only planar circuits. Since there are nonplanar graphs of degree three, $\nu_{d,k} = 0$ for $d \geq 3$. Define $\nu_{d,k}^{planar}$ as $\nu_{d,k}$ except that only planar graphs need be embedded. Clearly, $\nu_{1,k}^{planar} = \nu_{1,k}$ and $\nu_{2,k}^{planar} = \nu_{2,k}$.

The simplest type of circuit that involves k elements consists of pairs of connected elements. Such a circuit has degree 1. First, we consider

[4]In real PGA's, each grid edge represents several wires. Circuits of higher degree can therefore be implemented. This is discussed in Section 2.7.

embedding degree-1 graphs of small size. Degree-1 graphs with 2 or 3 vertices have only one edge, hence the same PGA's can implement both. Also, for one embedded edge, there is no advantage to edge connectivity. Therefore $\nu_{1,2}$, $\nu_{1,3}$, $\epsilon_{1,2}$, and $\epsilon_{1,3}$ coincide. In Section 2.3 we draw on an analogy to parking lots to prove that

$$\nu_{1,2} = \nu_{1,3} = \epsilon_{1,2} = \epsilon_{1,3} = \frac{2}{3} .$$

We argue that the largest density of parked cars in a square lot is precisely $\nu_{1,2}$. The standard parking lot arrangement consists of two parked rows, then one empty row, followed by two more parked rows, and so on. We show that this arrangement, which clearly achieves a $\frac{2}{3}$ density of parked cars, is asymptotically optimal.

In Section 2.4 we consider embedding degree-1 graphs of size 4 in vertex-connectivity PGA's. We show that

$$\nu_{1,4} = \frac{1}{2} .$$

We then turn to embedding graphs of general fixed size. In Section 2.5 we consider embedding size-k circuit graphs in edge-connectivity PGA's. Clearly $\epsilon_{0,k} = 1$, and we remarked earlier that $\epsilon_{d,k} = 0$ for $d \geq 5$. We show that for all k

$$\frac{1}{8(k+1)} \leq \epsilon_{4,k} \leq \epsilon_{3,k} \leq \epsilon_{2,k} \leq \epsilon_{1,k} \leq \frac{32}{k} .$$

That is, for $1 \leq d \leq 4$

$$\epsilon_{d,k} \in \Theta\left(\frac{1}{k}\right) .$$

Hence, regardless of whether the PGA must implement just the simplest kind of circuits (pairs of processors) or the most complicated ones (arbitrary degree-4 circuits), the highest possible density of processing nodes is asymptotically of the same order.

In grid graphs each vertex has degree ≤ 4. Hence with edge connectivity, any switching node can simultaneously connect at most two pairs of wires. With vertex connectivity each switching node can connect one pair. Although this difference might appear inconsequential, we show in

Section 2.6 that it accounts for a large difference in achievable density. Again, $\nu_{0,k} = 1$ and $\nu_{d,k} = 0$ for $d \geq 5$, but for all k,

$$\frac{c_1}{k^2} \leq \nu_{4,k}^{planar} \leq \nu_{3,k}^{planar} \leq \nu_{2,k} \leq \nu_{1,k} \leq \frac{c_2}{k^2} .$$

where c_1 and c_2 are constants. That is, for $1 \leq d \leq 4$,

$$\nu_{d,k}^{planar} \in \Theta\left(\frac{1}{k^2}\right) .$$

This shows that as k, the size of the circuits guaranteed implementable, increases, the component densities achievable in vertex and edge connectivity PGA's decrease at different rates. Hence, the capability to connect more than one pair of wires at a switching node allows for a significant increase in density.

The results of Sections 2.5 and 2.6 are related to those of [4], [1], and [2]. These papers consider the minimal size of a grid that can embed planar graphs of size n with prescribed vertex locations. Ignoring slight model differences, their results show that in the model equivalent to vertex connectivity the grid has to be of size $\Theta(n^3)$ while in the model equivalent to edge connectivity the required size is $\Theta(n^2)$.

These results can be derived from those described in Sections 2.5 and 2.6. Let n denote both the number of processing nodes and the size of the embedded graphs[5]. For edge connectivity, we show that $\frac{n}{grid\ size} = \Theta\left(\frac{1}{n}\right)$, therefore $grid\ size = \Theta(n^2)$. For vertex connectivity, $\frac{n}{grid\ size} = \Theta\left(\frac{1}{n^2}\right)$, therefore $grid\ size = \Theta(n^3)$. Thus, the results of Sections 2.5 and 2.6 can be viewed as generalizing known results to the embedding of graphs of size k where k is not necessarily the number of processing nodes.

Most of the paper is devoted to the model discussed in this introduction. The PGA graph is a two dimensional grid. Each edge represents one wire. The electrical elements are located in the processing nodes. A processing node can only connect the element it contains to a wire; it cannot connect any pair of wires. In Section 2.7 we discuss alternative models. We show that the asymptotic results remain valid even if the processors are placed in the squares rather than at the nodes and even

[5]The proofs have to be slightly modified to account for increasing graph size.

if each can connect to all vertices around it. They are unchanged even if the processing nodes can connect pairs of wires.

We show that adding the diagonal edges to the grid, which only doubles the number of wires (see Figure 2.18), increases the asymptotic component density in vertex-connectivity PGA's to $\Theta\left(\frac{1}{k}\right)$, that which is achievable with edge-connectivity grid PGA's (with or without the diagonal edges). As we shall see later on, this occurs partly because the resulting PGA graph is no longer planar.

2.2 Definitions

A *graph* is an ordered pair (V, E). V is a set, called the *vertex set*; its elements are *vertices* or *nodes*. E is a collection of *edges*: unordered pairs of vertices. An edge consisting of two identical vertices is a *self edge*, and an edge appearing more than once in E is a *multiple edge*. The following standard definitions assume that $G = (V, E)$ is a graph. Strictly speaking, all definitions should be qualified by the underlying graph (e.g., *vertices of* G) but, for brevity, G is omitted throughout.

If u and v are vertices and $\{u, v\}$ is an edge, u and v are *adjacent*, *neighbors*, or *connected by* $\{u, v\}$, and the edge $\{u, v\}$ *connects* u and v. A *path* is a sequence $v_0, \ldots, v_l$ of vertices such that v_i is adjacent to v_{i+1} for $i = 0, \ldots, l-1$. The vertices v_0 and v_l are the *end vertices* of the path and the vertices $v_1, \ldots, v_{l-1}$ are the *interior vertices*. The edges $\{v_i, v_{i+1}\}$ for $i = 0, \ldots, l-1$ are the *edges of the path*. Let u and v be vertices of G. A *path between* u *and* v, or a *path connecting* u *and* v, is a path whose end vertices are u and v. u and v are *connected* if there is a path between them. Note that a single vertex is a path; hence a vertex is always connected to itself even if it is not contained in a self loop. Let $S \subseteq V$ be a set of vertices. A *path in* S is a path whose interior vertices are in S. The end vertices may or may not be in S.

Two paths $u_0, \ldots, u_l$ and $v_0, \ldots, v_m$ are *vertex disjoint* if they do not share any vertices: $u_i \neq v_j$ for all $0 \leq i \leq l$ and all $0 \leq j \leq m$. The paths are *edge disjoint* if they do not share any edge. A set of paths is *vertex disjoint (edge disjoint)* if every pair of two paths in the set is vertex disjoint (edge disjoint).

Let $G' = (V', E')$ be a graph with $V' \subseteq V$ and let $S \subseteq V$. A

vertex-disjoint embedding of G' *in* S is a mapping that, with each edge $\{v_1, v_2\} \in E'$, associates a path in S connecting v_1 and v_2 such that the resulting set of paths is vertex disjoint. If such an embedding exists, G' is *vertex-disjoint embeddable in* S. An *edge-disjoint embedding of* G' in S, and being *edge-disjoint embeddable in* S, are defined similarly except that the paths have to be edge disjoint.

Recalling the Programmable Gate Array model described in the introduction, think of G as the PGA graph with E as the set of wires and V as the set of nodes, and think of S as the set of switching nodes. The graph G' represents a circuit we want the PGA to implement. Each vertex of G' is an electrical element corresponding to a unique component, or vertex, of V. Note that the vertices of G' are fixed. They correspond to specific components in the PGA. The only freedom in the embedding is in the choice of the interior vertices of the paths. In vertex-connectivity PGA's, the paths cannot cross at a vertex. Hence the circuit G' can be implemented by the PGA if and only if G' is vertex-disjoint embeddable in S. In the edge-connectivity model, G' can be implemented by the PGA if and only if G' is edge-disjoint embeddable in S.

The *size* $|S|$ of a set S is the number of its elements. The *size* of a graph is the size of its vertex set. A graph is *size-k* if its size is at most k. The *degree* of a vertex is the number of edges containing it. A self edge counts twice and multiple-edge counts add. The *degree* of a graph is the largest degree of its vertices. A graph is *degree-d* if its degree is at most d. The complement of a subset S of vertices is $V - S \stackrel{def}{=} \{v : v \in V \text{ and } v \notin S\}$.

Let d and k be positive integers. A subset P of V is (d,k)-*vertex-disjoint embedding* if every degree-d, size-k graph with vertices in P is vertex-disjoint embeddable in $V - P$. (Note that it is P that we call(d,k)-vertex-disjoint embedding although the graphs are embeddable in $V - P$.) The (d,k)-*vertex-disjoint density* of a graph G is the size of the largest (d,k)-vertex-disjoint embedding set in G, normalized by the size of G:

$$\nu_{d,k}(G) \stackrel{def}{=} \frac{\max\{|P| : P \text{ is } (d,k)\text{-vertex-disjoint embedding }\}}{|V|}.$$

In the PGA model, G is the PGA graph, P represents the process-

ing nodes, and $V - P$ is the set of switching nodes, assumed vertex connecting. P is (d,k)-vertex-disjoint embedding if and only if every degree-d, size-k circuit whose electrical elements correspond to components of P can be implemented by the PGA. The (d,k)-vertex-disjoint density of G is the largest possible proportion of processing nodes in a vertex-connecting PGA based on G if this PGA can implement every degree-d, size-k circuit.

We define a *(d,k)-edge-disjoint embedding* set similarly, and let the *(d,k)-edge-disjoint density* of a graph be:

$$\epsilon_{d,k}(G) \stackrel{def}{=} \frac{\max\{|P| : P \text{ is } (d,k)\text{-edge-disjoint embedding }\}}{|V|} .$$

For all d and k and all graphs G,

$$\nu_{d,k}(G) \leq \epsilon_{d,k}(G) ,$$

and if $d' \geq d''$ and $k' \geq k''$ then

$$\nu_{d',k'}(G) \leq \nu_{d'',k''}(G) .$$

Because of the application motivating this paper, we are not interested in the full generality of (d,k)-vertex-disjoint or (d,k)-edge-disjoint embeddings. Rather, we restrict our attention to the *2-dimensional square grid graphs*, which model most existing PGA's[6]. For every $m,n \geq 1$, the *2-dimensional m by n grid* denoted by $G_{m,n} \stackrel{def}{=} (V_{m,n}, E_{m,n})$ is the graph whose vertex set is $V_{m,n} \stackrel{def}{=} \{0,\ldots,m-1\} \times \{0,\ldots,n-1\}$ and whose edges are the *horizontal* edges: $\{\{(i,j-1),(i,j)\} : i = 0,\ldots,n-1,\ j = 1,\ldots,n-1\}$ and the *vertical* edges: $\{\{(i-1,j),(i,j)\} : i = 1,\ldots,n-1,\ j = 0,\ldots,n-1\}$; cf. Figure 2.3. Note that the first coordinate corresponds to the *vertical* axis and that coordinate values increase from top to bottom and from left to right. These conventions are adopted throughout the paper. We call $G_{n,n}$ a *square grid* and denote $G_{n,n}$, $V_{n,n}$, and $E_{n,n}$ by G_n, V_n, and E_n respectively.

A *planar representation* of a graph G is a drawing in the plane consisting of a dot corresponding to each vertex of G and a simple curve

[6]For several variations, see Section 2.7.

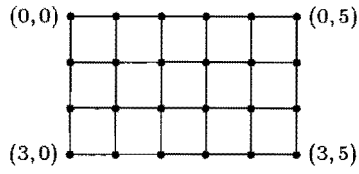


Figure 2.3: $G_{4,6}$.

in the plane connecting two dots if and only if the corresponding vertices are connected by an edge in G. At most two lines are allowed to intersect at any point in the plane. Every graph has a planar representation. The *crossing number* of a graph G is the minimal number of line crossings in a planar representation of G. A graph with crossing number 0 is *planar*.

If G' is vertex-disjoint or edge-disjoint embeddable in a subset of V, then the degree of every vertex $v \in V'$ in G' is not larger than its degree in G. Therefore in grid graphs $\epsilon_{d,k}(G_n)$ is nonzero only for $d \leq 4$. In practice, each edge of G_n often represents several actual wires, in which case circuits of degree larger than four can be implemented[7].

Furthermore, with vertex connectivity when a graph G' is embedded in another graph G, each edge of G' is mapped into a path in G and distinct paths do not intersect at any node. A planar representation of G can therefore be used to derive a planar representation of G' with at most as many line crossings. Hence, in vertex connectivity, a graph can be embedded only in a graph with at least as large a crossing number. Grid graphs are planar and therefore can only embed planar graphs. Since there is a non-planar graph of six vertices and degree three, $\nu_{d,k}(G_n)$ is zero for $d \geq 3$ and $k \geq 6$.

We are interested in $\nu_{d,k}(G_n)$ for large grids. Therefore we define the (d,k)-*vertex-disjoint density*:

$$\nu_{d,k} \stackrel{def}{=} \limsup_{n \to \infty} \nu_{d,k}(G_n) \ .$$

[7]See Section 2.7.

Analogously, we define the (d,k)-*edge-disjoint density*:

$$\epsilon_{d,k} \stackrel{def}{=} \limsup_{n\to\infty} \epsilon_{d,k}(G_n) \ .$$

$\nu_{d,k}$ has the interpretation of the largest proportion of grid vertices that can serve as processing nodes in a vertex-connectivity PGA if every degree-d, size-k circuit can be implemented by the PGA. $\epsilon_{d,k}$ has the same interpretation for edge-connectivity PGA's.

A description and a brief interpretation of the results obtained in the paper were given in the last part of the introduction.

2.3 Embedding Degree-1 Graphs of Size 2 or 3

Consider embedding degree-1 graphs with 2 or 3 vertices. Such graphs have at most one edge. Therefore, a set is $(1,2)$-vertex-disjoint embedding if and only if it is $(1,3)$-vertex-disjoint embedding (similarly for edge-disjoint embeddings). Furthermore, there is no advantage to a path that intersects itself; hence, vertex-disjoint and edge-disjoint connectivity coincide and

$$\nu_{1,2} = \nu_{1,3} = \epsilon_{1,2} = \epsilon_{1,3} \ .$$

We determine $\nu_{1,2}$. To obtain a lower bound, we draw on an analogy. Consider a parking lot consisting of an n by n array of square spaces. One of the spaces on the perimeter is marked EXIT. Each space either contains a parked car or is vacant. Cars can move from a space to any of the four adjacent spaces provided it is vacant. The parking lot can accommodate n^2 cars but then only one car can move to the EXIT space: the car that is there already. It is desirable to allow each car to "EXIT" without moving any other car. How many cars can be parked?

The parking lot problem is closely related to $(1,2)$-vertex-disjoint embeddability. An n by n parking lot with p parked cars can be used to derive a $(1,2)$-vertex-disjoint embedding set P of size p in G_n. Consider the natural correspondence between G_n and an n by n parking lot in which each node of G_n represents the corresponding parking space. Let

P consist of the nodes corresponding to spaces containing parked cars, as illustrated in Figure 2.4. (Note that the arrangement shown is not valid because the car marked "1" cannot EXIT.) Since every car can EXIT, there is a path in $V_n - P$ from every node in P to the EXIT node (the node corresponding to the EXIT space). Hence, there is a path in $V_n - P$ between any two nodes in P. The set P of processing nodes is therefore $(1,2)$-vertex-disjoint embedding and of size p. The converse, though not used (we prove the upper bound directly), is also correct. If a set $P \subseteq V_n$ is $(1,2)$-vertex-disjoint embedding, eliminate from P a node closest to the EXIT node; the parking spaces corresponding to the remaining nodes can be used to park cars. Therefore, as n increases, the proportion of cars that can be parked in an n by n parking lot approaches $\nu_{1,2}$.

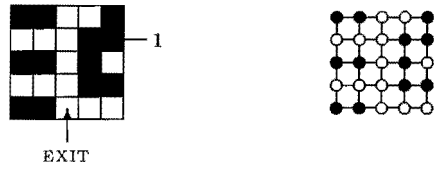


Figure 2.4: An arrangement of parked cars and the corresponding PGA.

The standard parking-lot arrangement is depicted in Figure 2.5. Two pairs of parked columns are separated by one column of vacant spaces. For n divisible by 3, the total number of parked cars is $\frac{2}{3}n(n-1)+2$. Hence,

$$\nu_{1,2}(G_n) \geq \frac{2}{3} - \frac{2}{3n} + \frac{2}{n^2} ,$$

implying

$$\nu_{1,2} \geq \frac{2}{3} .$$

In Theorem 1 we shall show that

$$\nu_{1,2} \leq \frac{2}{3} ,$$

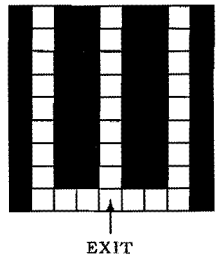


Figure 2.5: Standard arrangement of a square parking lot.

thereby proving that the standard arrangement asymptotically achieves optimal density. First, we need some preliminaries.

Let $S \subseteq V$ be a set of vertices. Two vertices (not necessarily in S) are *connected in* S if there is a path in S that connects them. As a property, being connected in S is reflexive and symmetric but, in general, not transitive. For example, in the graph $u-v-w$, where edges are denoted by $-$, the vertices u and v are connected in $S \stackrel{def}{=} \{u, w\}$, as are v and w. However, the only path connecting u and w contains the interior vertex v, so u is not connected to w in S. However, when restricted to elements of S, the property of being connected in S is also transitive, hence an equivalence relation. Therefore, a set S is partitioned into equivalence classes of *S-connected components*. A set S of vertices is *self connected* if every two of its elements are connected in S. The following properties hold for any set S:

P1. If two vertices are connected in S, they are connected in one of the S-connected components.

P2. Each of the S-connected components is self connected.

P3. S is self connected if and only if it has only one S-connected component.

P4. S is self connected if and only if $G|_S$, the *restriction* of G to S, is connected. ($G|_S$ is the graph whose vertex set is S and in which $u, v \in S$ are connected by an edge if and only if $\{u, v\}$ is an edge in E.)

Let $S \subseteq V$. The *vicinity of* S is the set of vertices outside S that are adjacent to vertices of S:

$$\mathcal{V}(S) \stackrel{def}{=} \{v \in V - S : v \text{ is adjacent to some } s \text{ in } S\} .$$

Lemma 1 *If S is self connected in a two-dimensional grid, then*

$$|\mathcal{V}(S)| \leq 2|S| + 2 .$$

Proof

By induction on $|S|$. If $|S| = 1$ then S has at most 4 neighbors. If S is self connected and has more than one element, there is an element $s \in S$ such that $S - \{s\}$ is self connected (S can be chosen as a leaf of a spanning tree for $G|_S$). By the induction hypothesis, $|\mathcal{V}(S - \{s\})| \leq 2(|S| - 1) + 2 = 2|S|$.

Let $T \stackrel{def}{=} (S - \{s\}) \bigcap \mathcal{V}(\{s\})$ be the set of vertices in $S - \{s\}$ that are neighbors of s. T is not empty because S is self connected and has at least 2 elements. Clearly,

$$\mathcal{V}(S) = \Big(\mathcal{V}(S - \{s\}) \bigcup \mathcal{V}(\{s\})\Big) - \Big(T \bigcup \{s\}\Big) .$$

Therefore,

$$\begin{aligned} |\mathcal{V}(S)| &\leq |\mathcal{V}(S - \{s\})| + |\mathcal{V}(\{s\})| - |\mathcal{V}(S - \{s\}) \bigcap \mathcal{V}(\{s\})| - |T| - 1 \\ &\leq 2|S| + 4 - 0 - 1 - 1 \\ &= 2|S| + 2. \quad \square \end{aligned}$$

Theorem 1

$$\nu_{1,2} = \frac{2}{3} .$$

Proof

We argued earlier that $\nu_{1,2} \geq \frac{2}{3}$. Now we show that $\nu_{1,2} \leq \frac{2}{3}$. Let S and P be disjoint sets of vertices in G_n such that P is connected in S. Pick a vertex $p \in P$ and let $S_1, S_2, \ldots, S_I$ be the S-connected components with elements adjacent to p (that is, $p \in \mathcal{V}(S_i)$ for $1 \leq i \leq I$). These

components are disjoint and p has at most 4 neighbors; hence $0 \leq I \leq 4$. *Since P is connected in S, any element of P is S-connected to p. From property 2.3 above, any element of P is* S_i*-connected to p for some* $1 \leq i \leq I$. *But a vertex v is* S_i *connected to p if and only if* $v \in \mathcal{V}(S_i)$ *or* $v \in \mathcal{V}(p)$. *Therefore,* $P \subseteq \mathcal{V}(p) \cup \left(\cup_{i=1}^{I} \mathcal{V}(S_i)\right)$. *By the previous lemma,*

$$\begin{aligned} |P| &\leq (4-I) + \sum_{i=1}^{I} |\mathcal{V}(S_i)| - (I-1) \\ &\leq (4-I) + \sum_{i=1}^{I} 2|S_i| + 2I - (I-1) \\ &\leq 2|S| + 5 \ . \end{aligned}$$

But $|P| + |S| \leq n^2$. *Combined, the last two inequalities yield:* $|P| \leq 2(n^2 - |P|) + 5$, *or*

$$|P| \leq \frac{2}{3}n^2 + \frac{5}{3} \ .$$

Therefore, $\nu_{1,2} = \lim_{n\to\infty} \nu_{1,2}(G_n) \leq \lim_{n\to\infty} \frac{2}{3} + \frac{5}{3n^2} = \frac{2}{3}$. □

2.4 Embedding Degree-1 Graphs of Size 4

Unlike graphs of size two or three, embedding graphs of size 4 may be different with vertex and edge connectivity. We consider only vertex connectivity.

The subset of V_n consisting of the center $n-2$ elements in every other column (see Figure 2.6) is easily seen to be $(1,4)$-vertex-disjoint embedding, so $\nu_{1,4} \geq \frac{1}{2}$. We prove a matching lower bound thereby showing that

$$\nu_{1,4} = \frac{1}{2} \ .$$

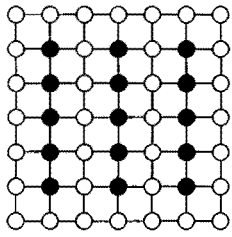


Figure 2.6: A $(1,4)$-vertex-disjoint embedding set of size 15 in G_7.

Let $\mathcal{Z}$ denote the set of integers. Define the *south-eastern neighborhood of a point* $s = (i,j) \in \mathcal{Z}^2$ to be

$$SE(s) \stackrel{def}{=} \{(i+1,j),(i,j+1),(i+1,j+1)\} ,$$

the set of grid points, 'south,' 'east,' and 'south east' of s. Define the *south-eastern neighborhood of a set* $S \subseteq \mathcal{Z}^2$ to be

$$SE(S) \stackrel{def}{=} \bigcup_{s \in S} SE(s) - S ,$$

the set of points in $\mathcal{Z}^2 - S$ that are 'south,' 'east,' or 'south east' of some point in S. Figure 2.7 illustrates a set and its south-eastern vicinity. The elements of the set are denoted by dots and those of the south-eastern neighborhood by ×'s. Lemma 5, proved below, shows that if P is $(1,4)$-vertex-disjoint embedding, then $|SE(P)| \geq |P|$. This enables us to prove

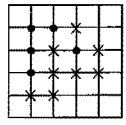


Figure 2.7: A set and its south-eastern neighborhood

Theorem 2

$$\nu_{1,4} \leq \frac{1}{2} .$$

Proof

Let P be $(1,4)$-vertex-disjoint embedding in G_n. By definition, P and $SE(P)$ are disjoint and their union is contained in $\{0,\ldots,n\}\times\{0,\ldots,n\}$. Therefore,

$$|P| + |SE(P)| \leq (n+1)^2$$

From Lemma 5, $|P| \leq |SE(P)|$, so

$$\frac{|P|}{n^2} \leq \frac{1}{2} + \frac{1}{n} + \frac{1}{2n^2} .$$

This holds for all n and all $(1,4)$-vertex-disjoint embedding sets in G_n. Letting n tend to infinity, we obtain the theorem. □

To prove that $|SE(P)| \geq |P|$, we first show that $(1,4)$-vertex-disjoint embedding sets are comprised of simple subsets. Let $d(s_1,s_2) \stackrel{def}{=} ((i_1-i_2)^2 + (j_1-j_2)^2)^{1/2}$ denote the *Euclidean distance* between two points $s_1 = (i_1,j_1)$ and $s_2 = (i_2,j_2)$ in $\mathcal{Z}^2$. A mapping $\phi : \mathcal{Z}^2 \rightarrow \mathcal{Z}^2$ is *distance-preserving* or an *isometry* if $d(s_1,s_2) = d(\phi(s_1),\phi(s_2))$ for all $s_1,s_2 \in \mathcal{Z}^2$. Two subsets S and T of $\mathcal{Z}^2$ are *isometric* if there is an isometry from S onto T. Since any isometry is $1-1$, isometric sets have the same cardinality. (Also, two isometric sets are related via a sequence of rotations, translations, and reflections.)

The restriction $G_n|_P$ of the square grid G_n to a set $P \subseteq V_n$ was defined in Section 2.3. Let $G_1(P),\ldots,G_k(P)$ be the connected components of $G_n|P$ (see Figure 2.8), and for $i = 1,\ldots,k$ let P_i be the vertices of $G_i(P)$.

Define $\mathcal{Z}^+ \stackrel{def}{=} \{0,1,2,\ldots\}$ and $\mathcal{Z}^- \stackrel{def}{=} \{0,-1,-2,\ldots\}$. The following lemma can be proved by inspection.

Lemma 2 *If P is $(1,4)$-vertex-disjoint embedding in G_n then each P_i is isometric to a connected subset of one of the following sets:*

1. $S_L \stackrel{def}{=} \mathcal{Z}^- \times \{0\} \cup \{0\} \times \mathcal{Z}^+$

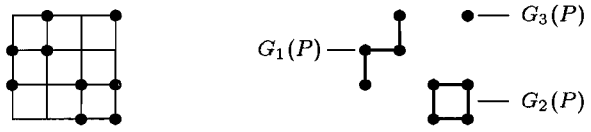


Figure 2.8: A set P in G_4 and its connected-components.

2. $S_T \stackrel{def}{=} \{-1\} \times \mathcal{Z} \cup \{(0,0)\}$

3. $S_Z \stackrel{def}{=} \{0\} \times \mathcal{Z}^+ \cup \{-1\} \times \mathcal{Z}^-$.

Subsets of S_L, S_T, and S_Z are illustrated in Figure 2.9 which also suggests the reason why these three subscripts where selected. □

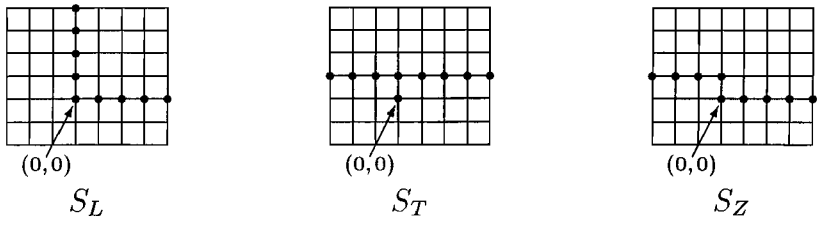


Figure 2.9: The different types of $(1,4)$-vertex-disjoint embedding connected components.

Lemma 3 *If C is a $(1,4)$-vertex-disjoint embedding connected component, then*

$$|SE(C)| = |C| + 2 \ .$$

Proof

By induction on the size of C. There are fourteen types of connected components: the four rotations of each of S_L, S_T, and S_Z, the 'horizontal' connected component, and the 'vertical' one. Induction is carried

out separately for each. Figure 2.10 shows four of the inductions bases. □

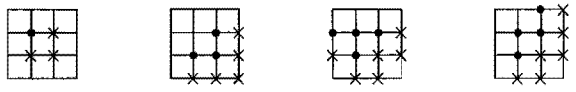


Figure 2.10: A few induction bases showing that $|SE(C)| = |C| + 2$.

Let $P \subseteq V_n$ be $(1,4)$-vertex-disjoint embedding and let $P_1, \ldots, P_k$ be the vertex sets of P's connected components as defined earlier. For each $i \in \{1, \ldots, k\}$ define S_i, T_i, and U_i as follows:

$$S_i \stackrel{def}{=} \{x \in SE(P_i) : x \notin SE(P_j) \text{ and } x \notin P_j \text{ for all } j \neq i\},$$
$$T_i \stackrel{def}{=} \{x \in SE(P_i) : x \in SE(P_j) \text{ for some } j \neq i\},$$
$$U_i \stackrel{def}{=} \{x \in SE(P_i) : x \in P_j \text{ for some } j \neq i\}.$$

For every $i \in \{1, \ldots, k\}$, the sets S_i, T_i, and U_i are disjoint and their union is $SE(P_i)$. First, we bound the number of points in T_i and U_i.

Lemma 4 *Let $P \subseteq V_n$ be $(1,4)$-vertex-disjoint embedding. For $i \in \{1, \ldots, k\}$ define P_i, S_i, T_i, and U_i as above. Then, for all $i \in \{1, \ldots, k\}$,*

$$\frac{|T_i|}{2} + |U_i| \leq 2 \ .$$

Proof
All fourteen cases mentioned in the proof of Lemma 3 need to be examined. We illustrate four of them here.

Let P_i be the connected component depicted in Figure 2.11(a). Only the point u_1 may belong to U_i and only t_1 and t_2 may belong to T_i. Hence $|T_i| \leq 2$ and $|U_i| \leq 1$, so $\frac{|T_i|}{2} + |U_i| \leq 2$. Note that if $t_3 \in T_i$, then $v \in P$, and the connections $v-2$, $1-3$ cannot be made simultaneously; similarly, $t_4 \notin T_i$.

Substituting u_2 for t_4 at the end of the last sentence, the argument above also shows that $\frac{|T_i|}{2} + |U_i| \leq 2$ for the connected component P_i depicted in Figure 2.11(b).

For the connected component P_i depicted in Figure 2.11(c), only the point u_1 may belong to U_i and only t_1, t_2, and t_3 may belong to T_i. If $\frac{|T_i|}{2} + |U_i| > 2$, then $u_1 \in U_i$ and all of t_1, t_2, t_3 are in T_i. In particular, u_1 and v are both in P. But then, regardless of the other points in P, the connections $u_1 - 1$ and $v - 2$ cannot be made simultaneously.

For the connected component P_i depicted in Figure 2.11(d), only the points u_1 and u_2 may belong to U_i, and only t_1 and t_2 may belong to T_i. If $\frac{|T_i|}{2} + |U_i| > 2$, then u_1 and u_2 are both in U_i and at least one of t_1 and t_2, say t_1, is in T_i. But then u_1, u_2, and v are in P and, regardless of the other points in P, the connections $u_1 - 1$ and $v - 2$ cannot be made simultaneously. □

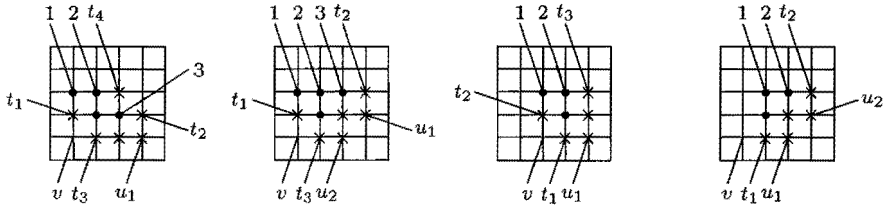


Figure 2.11: Four connected components satisfying $\frac{|T_i|}{2} + |U_i| \leq 2$.

Lemma 5 *Let P be $(1,4)$-vertex-disjoint embedding in G_n. Then*

$$|SE(P)| \geq |P| .$$

Proof

Let $P_1, \ldots, P_k$ be the vertex sets of the connected components of P and define S_i, T_i, and U_i for $i \in \{1, \ldots, k\}$ as above. An element of V_n belongs to $SE(P)$ if and only if it belongs to T_i or U_i for some $i \in \{1, \ldots, k\}$. Hence,

$$SE(P) = \bigcup_{i=1}^{k} (S_i \bigcup T_i) = (\bigcup_{i=1}^{k} S_i) \bigcup (\bigcup_{i=1}^{k} T_i) .$$

By definition, all the S_i's are disjoint and each element of $\bigcup_{i=1}^{k} T_i$ belongs to T_i for exactly two indices i. Hence, using Lemmas 3 and 4

$$\begin{aligned} |SE(P)| &= \sum_{i=1}^{k} |S_i| + \frac{\sum_{i=1}^{k} |T_i|}{2} \\ &= \sum_{i=1}^{k} (|S_i| + |T_i| + |U_i|) - (\frac{|T_i|}{2} + |U_i|) \\ &\geq \sum_{i=1}^{k} (|P_i| + 2) - 2 \\ &= |P| \ . \end{aligned}$$

□

2.5 Edge-Disjoint Embeddings of Size-k Graphs

In this section we assume that the switching nodes are edge connecting. Each can simultaneously connect two wire pairs. We show that for all $k \geq 1$,

$$\frac{1}{8(k+1)} \leq \epsilon_{4,k} \leq \epsilon_{3,k} \leq \epsilon_{2,k} \leq \epsilon_{1,k} \leq \frac{32}{k} \ .$$

That is, for all $1 \leq d \leq 4$,

$$\epsilon_{d,k} \in \Theta\left(\frac{1}{k}\right) \ .$$

As mentioned in the introduction, $\epsilon_{d,k} = 0$ for $d \geq 5$ and $\epsilon_{0,k} = 1$. We begin by proving the upper bound.

Theorem 3 *For all $k \geq 1$,*

$$\epsilon_{1,k} \leq \frac{32}{k} \ .$$

Proof

Let $n > \frac{k}{4}\max\{\frac{1}{\sqrt{2}}, \frac{\sqrt{k}}{8}\}$ *and let* $P \subset V_n$ *be* $(1,k)$*-edge-disjoint embedding in* G_n*. We assume that*

$$\frac{|P|}{n^2} > \frac{32}{k}$$

and derive a contradiction. As illustrated in Figure 2.12, partition V_n *into* $\frac{2|P|}{k}$ *subsets*[8]*, each a subgrid of dimensions at most* $n\sqrt{\frac{k}{2|P|}}$ *by* $n\sqrt{\frac{k}{2|P|}}$.

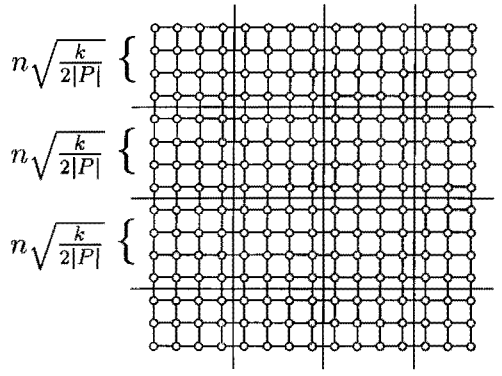


Figure 2.12: Partition of G_n *into subgrids.*

At least one of these $\frac{2|P|}{k}$ *subgrids contains* $\frac{k}{2}$ *elements of* P*. Also, by the assumptions just made,*

$$|P| > \frac{32}{k}n^2 > \frac{32}{k}\left(\frac{k}{4\sqrt{2}}\right)^2 = k$$

and

$$|P|^2 > \left(\frac{32}{k}n^2\right)^2 > n^2\left(\frac{k\sqrt{k}}{32}\right)^2\left(\frac{32}{k}\right)^2 = n^2k\ .$$

[8]For simplicity we disregard integer roundoffs (e.g. $\left\lceil\frac{2|P|}{k}\right\rceil$ subsets). Since the derived bound is asymptotic (n is arbitrarily large), it is not affected by these approximations.

Hence,

$$n^2 \frac{k}{2|P|} < \frac{|P|}{2} < |P| - \frac{k}{2} ,$$

so there must be at least $\frac{k}{2}$ *elements of* P *outside any subgrid.*

Consider a subgrid with at least $\frac{k}{2}$ *elements of* P. *Since* P *is* $(1,k)$*-edge-disjoint embedding, there must be* $\frac{k}{2}$ *edge-disjoint paths, each connecting an element of* P *in the subgrid and an element of* P *outside the subgrid. Therefore, there must be at least* $\frac{k}{2}$ *edges connecting the subgrid to the rest of the grid. But the number of such edges is bounded above by the circumference of the subgrid which is at most* $4n\sqrt{\frac{k}{2|P|}}$. *Hence,*

$$\frac{k}{2} \leq \textit{circumference} \leq 4n\sqrt{\frac{k}{2|P|}}$$

which contradicts our initial assumption. □

Next, we prove a matching lower bound by applying a combination of standard embedding and 'snaking' techniques (e.g., [4], [1]) to our problem.

Theorem 4 *For all* $k \geq 1$,

$$\epsilon_{4,k} \geq \frac{1}{8(k+1)} .$$

Proof

First we show that the set $P \stackrel{def}{=} \{(1, 4i+1) : i = 0, \ldots, n-1\}$ *is* $(4,k)$*-edge-disjoint embedding in the grid* $G_{2k+2,4n}$. *Then we show how to "map" this rectangular grid into the square grid* $G_{2(2k+2)+\lceil\sqrt{8n(k+1)}\rceil}$. *This implies*

$$\epsilon_{4,k} \geq \limsup_{n\to\infty} \frac{n}{\left(2(2k+1) + \left\lceil\sqrt{8n(k+1)}\right\rceil\right)^2} = \frac{1}{8(k+1)} .$$

Figure 2.13 shows $G_{2k+2,4n}$, *the set* P *for* $k = n = 4$, *and an embedding of a sample graph. When embedding a degree-4, size-k graph* G',

we sequentially embed each edge (in any order). Suppose that an edge connecting the i'th and the j'th nodes of G' has not yet been embedded. Since G' is of degree ≤ 4, at least one of the four lines leaving vertex $(1, 4i+1)$ and one of the four lines leaving vertex $(1, 4j+1)$ are not used. We use these lines to get from the vertices to the first row that has not been used and connect the paths in that row. The row will not be used again.

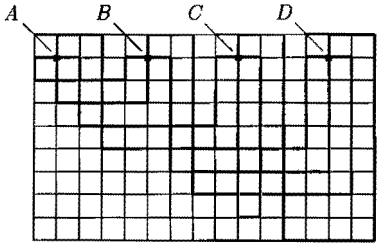


Figure 2.13: Embedding the graph A-B, A-B, A-C, A-D, B-D, B-D, C-C, C-D, in $G_{10,16}$.

The process is repeated until all edges have been embedded in $G_{2k+2,4n}$. Since G' is of size k and of degree at most 4 it has at most $2k$ edges so $2k+2$ rows suffice. Therefore, G' can be embedded in $G_{2k+2,4n}$.

To map $G_{2k+2,4n}$ into $G_{2(2k+2)+\left\lceil\sqrt{8n(k+1)}\right\rceil}$, we 'fold' it as shown in Figure 2.14 into a rectangular grid at most $\left\lceil\sqrt{8n(k+1)}\right\rceil$ by $\left\lceil\sqrt{8n(k+1)}\right\rceil$ in size. The added strips of width $2k+1$ ensure that the original paths can be rerouted in the folded version. The resulting grid has side at most $2(2k+1)+\left\lceil\sqrt{8n(k+1)}\right\rceil$. □

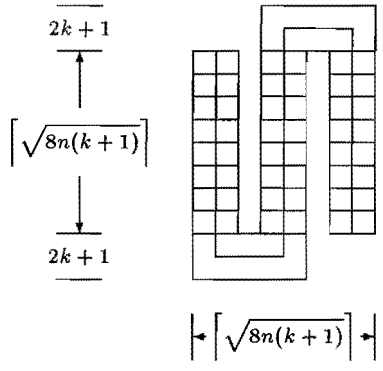


Figure 2.14: Mapping a rectangular grid into a square grid.

2.6 Vertex-Disjoint Embeddings of Size-k Graphs

In Section 2.5 we considered edge-connectivity PGA's. Each switching node could connect two wire pairs. We showed that the (d, k)-edge-disjoint density is

$$\epsilon_{d,k} \in \Theta\left(\frac{1}{k}\right) .$$

Now, we assume that the switching nodes of the PGA are vertex-connecting, that is, each can connect only one pair of wires. We show that this seemingly minor distinction accounts for a decided difference in achievable densities to the effect that for all k,

$$\frac{c_1}{k^2} \leq \nu_{4,k}^{planar} \leq \nu_{3,k}^{planar} \leq \nu_{2,k} \leq \nu_{1,k} \leq \frac{c_2}{k^2} .$$

where c_1 and c_2 are constants. That is, for $1 \leq d \leq 4$

$$\nu_{d,k}^{planar} \in \Theta\left(\frac{1}{k^2}\right) .$$

Thus as the size of implementable circuits increases, the achievable densities in vertex and edge connectivity PGA's decrease at different

rates. Switching nodes capable of simultaneously connecting two wire pairs significantly increase the achievable density.

We do not describe the lower bound here. [2] provided an algorithm for embedding a planar graph of size n in a grid of area $O(n^3)$ with prescribed vertex locations. This algorithm can be modified easily to embed any planar graph of size k in a rectangular vertex-connectivity PGA of area $O(nk^2)$ with n processing nodes. The rectangular grid can then be mapped as in the last section into a square grid of the same area. Therefore,

$$\nu_{d,k} = \Omega\left(\frac{n}{nk^2}\right) = \Omega\left(\frac{1}{k^2}\right) .$$

The matching upper bound,

$$\nu_{d,k} \leq \frac{c_2}{k^2} ,$$

uses a lemma of [10] for proving lower bounds on crossing numbers of graphs (stated here as Lemma 8) and a technique similar to one used by [1]. The proof, given in Theorem 5, is based on *chorded cycle graphs* and on two lemmas. These are described first.

For even k, a *chorded cycle of size k* (or a *chorded k-cycle*) is a graph whose vertex set is $\{1,\ldots,k\}$ and whose edges are the *cycle edges*: $\{\{k,1\},\{1,2\},\ldots,\{k-1,k\}\}$, and the *chord edges*: arbitrary non-self edges. Each vertex belongs to exactly one chord. An example of a chorded 8-cycle is shown in Figure 2.15.

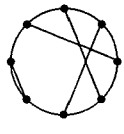


Figure 2.15: A chorded 8-cycle

The crossing number of a graph G was defined in Section 2.2 as the minimal number of line crossings in a planar representation of G. A graph with crossing number zero is planar. The graph consisting of just the cycle edges of a chorded k-cycle is planar as is the graph consisting of the chords only. The chorded k-cycle itself is not necessarily planar.

For example, the crossing number of the chorded 8-cycle in Figure 2.15 is one.

Since a chorded k-cycle has $\frac{k}{2}$ chords, its crossing number is at most $\binom{k/2}{2} \leq \frac{k^2}{8}$. The first lemma states that there is a chorded k-cycle with a crossing number asymptotically that high. Because the proof is lengthy, it is postponed until after Theorem 5.

Lemma 6 *For all even $k \geq 6$, there exists a chorded k-cycle with crossing number $\geq k^2/10^8$.*

The second lemma is the grid-graph equivalent of a well known geometrical result. If $|P|$ points are placed in the unit square, there is a path of length at most $c \cdot \sqrt{|P|}$ going through all the points. The exact constant multiplying $\sqrt{|P|}$ has been studied in several papers including [6] and [9], but is not of interest here. We prove a weak version of the lemma.

Let G be a graph. A path $v_0,\ldots,v_l$ in G is *simple* if no vertex appears twice in it: $v_i \neq v_j$ for all $0 \leq i < j \leq l$. The *length* of the path is l, the number of edges in the path.

Lemma 7 *Let $G_n = (V_n, E_n)$ be the square grid and let $P \subseteq V_n$. For every $1 \leq k \leq |P|$ there is a subset $p_0,\ldots,p_k$ of P and a simple path of length $\leq \frac{k}{|P|} n(4\sqrt{|P|}+2)$ in G_n connecting $p_0,\ldots,p_k$.*

Proof

We prove that there is a simple path of length $\leq n(2\sqrt{|P|}+1)$ connecting all points in P. The lemma then follows from averaging arguments. Partition V_n into $\sqrt{|P|}$ 'vertical stripes' $S_1,\ldots,S_{\sqrt{|P|}}$ where $S_i \stackrel{def}{=} \{0,\ldots,n-1\} \times \{(i-1)\frac{n}{\sqrt{|P|}},\ldots,i\frac{n}{\sqrt{|P|}}-1\}$. Let $a_i \stackrel{def}{=} |P \bigcap S_i|$ be the number of elements of P in the ith stripe and let $p^i_j = (x^i_j, y^i_j)$ be the jth element of $P \bigcap S_i$ where the elements are ordered by increasing x value for even i and decreasing x values for odd i (elements with the same x value are ordered by increasing y value). For even i, let $p^i_0 \stackrel{def}{=} (0,(i-1)\frac{n}{\sqrt{|P|}})$ and let $p^i_{a_i+1} \stackrel{def}{=} (n-1, i\frac{n}{\sqrt{|P|}}-1)$. For odd i reverse the definition of p^i_0 and $p^i_{a_i+1}$. For $i = 1,\ldots,\sqrt{|P|}$, the path connects p^i_j

to p^i_{j+1} for $j = 0, \ldots, a_i$ by first going 'horizontally' then 'vertically' if necessary. The path then connects $p^i_{a_i+1}$ to p_0^{i+1}. Therefore, for each i, the length of the path in stripe S_i is at most

$$\begin{aligned}\sum_{j=0}^{a_i} \left(|x^i_j - x^i_{j+1}| + |y^i_j - y^i_{j+1}|\right) &\leq \left|\sum_{j=0}^{a_i}(x^i_j - x^i_{j+1})\right| + \sum_{j=0}^{a_i} \frac{n}{\sqrt{|P|}} \\ &= |x^i_{a_i+1} - x^i_0| + (a_i+1)\frac{n}{\sqrt{|P|}} \\ &= n + (a_i+1)\frac{n}{\sqrt{|P|}} .\end{aligned}$$

The total length of the path is at most

$$\sum_{i=1}^{\sqrt{|P|}} (n + (a_i+1)\frac{n}{\sqrt{|P|}}) = n\sqrt{|P|} + |P|\frac{n}{\sqrt{|P|}} + n = n(2\sqrt{|P|}+1) . \quad \square$$

We now use Lemmas 6 and 7 to prove the upper bound.

Theorem 5 *For all $k \geq 1$,*

$$\nu_{1,k} \leq \frac{c_2}{k^2} .$$

Proof

Let $P \subseteq V_n$ be $(1,k)$-vertex-disjoint embedding and let W be a chorded k-cycle (or a chorded $(k-1)$-cycle if k is odd) with crossing number $\geq k^2/10^8$. Such a chorded cycle exists by Lemma 6. Denote the graph whose edges are the chords of W by W_c. Label the vertices of the chorded cycle $v_1, \ldots, v_k$, sequentially on the cycle. Let $p_1, \ldots, p_k$ be a sequence of elements of P such that there is a simple path of length at most $\frac{4.2kn}{\sqrt{|P|}}$ connecting $p_1, \ldots, p_k$ in that order. Lemma 7 guarantees the existence of such a sequence. Complete the path to a cycle in the plane by adding a parallel path alongside it and connecting the corresponding ends. The length of this cycle is at most $\frac{8.4kn}{\sqrt{|P|}}$. Identify vertex v_i of W with vertex p_i of V_n and find a vertex-disjoint embedding of W_c in $V_n - P$. Such an embedding exists as P is $(1,k)$-vertex-disjoint embedding and W_c is a degree-1 graph of size k.

Next, connect the simple path that goes through $p_1, \ldots, p_k$. The resulting graph is isomorphic to W and hence has a crossing number of at least $k^2/10^8$. These crossings are not cycle to cycle or chord to chord. Therefore they are all cycle edges crossing chord edges. The distance between any two chords is at least a grid unit. Therefore the total length of the path is at least $k^2/10^8$. Thus, $k^2/10^8 \leq$ length of the path $\leq \frac{8.4kn}{\sqrt{|P|}}$, or

$$\frac{|P|}{n^2} \leq \frac{8 \cdot 10^{17}}{k^2} . \quad \square$$

A graph $G = (V, E)$ is *α-expanding* for $\alpha > 0$ if every subset $S \subseteq V$ of size at most $|V|/2$ is adjacent to at least $\alpha|S|$ vertices in $V - S$. We now prove Lemma 6. The proof uses the following interesting result proved in [10]. (The constant 97 is derived using the planar-separator bound of [3].)

Lemma 8 *[10] The crossing number of a k-node, α-expanding graph is at least*
$k^2 \left[\left(\frac{\alpha}{10+\alpha} \right)^2 \frac{1}{97} - \frac{1}{k} \right]$. $\square$

We show that there exists a $\frac{1}{100}$-expanding chorded k-cycle. Let C_k be the k-vertex circle-graph with vertices: $\{1, \ldots, k\}$ and edges: $\{\{k, 1\}, \{1, 2\}, \{2, 3\}, \ldots, \{k-1, k\}\}$. A nonempty set $I \subseteq \{1, \ldots, k\}$ is an *island of C_k* if the graph $C_k|_I$ is connected. Every set $S \subseteq \{1, \ldots, k\}$ decomposes uniquely into islands of C_k. These are the connected components of $C_k|_S$. For example, the set $\{1, 3\}$ consists of 2 islands, $\{1\}$ and $\{3\}$, in C_4 and of the single island $\{1, 3\}$ in C_3.

Lemma 9 *The number of sets of size s that consist of i islands in C_k is*

$$\binom{s}{i}\binom{k-s-1}{i-1} + \binom{s-1}{i-1}\binom{k-s}{i} = \frac{k}{i}\binom{s-1}{i-1}\binom{k-s-1}{i-1} .$$

Proof
We use the following facts:

1. *The number of ways to partition the circle* $1,\ldots,l$ *into* m *islands is* $\binom{l}{m}$.

2. *The number of ways to partition the circle* $1,\ldots,l$ *into* m *islands so that 1 and* l *belong to different islands is* $\binom{l-1}{m-1}$.

Each set S *of size* s *consisting of* i *islands in* C_k *that contains the point* 1 *uniquely corresponds to a partition of the circle* $1,\ldots,s$ *into* i *islands combined with a partition of the circle* $1,\ldots,k-s$ *into* i *islands so that 1 and* $k-s$ *are in different islands. Hence there are* $\binom{s}{i}\binom{k-s-1}{i-1}$ *such sets.*

Similarly, each size s *set consisting of* i *islands in* C_k *that does not contain the point* 1, *uniquely corresponds to a partition of the circle* $1,\ldots,s$ *into* i *islands so that 1 and* s *are in different islands and a partition of the circle* $1,\ldots,k-s$ *into* i *islands. Therefore, there are* $\binom{s-1}{i-1}\binom{k-s}{i}$ *such sets.* □

Example 2 *Lemma 9 implies that there are* $\binom{5}{3}\binom{3}{2}+\binom{4}{2}\binom{4}{3}=54$ *sets of size 5 consisting of 3 islands in* C_9. *They are all 9 cyclic shifts of the sets in Figure 2.16.* □

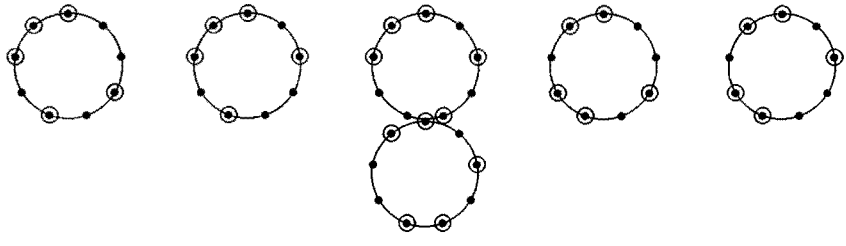


Figure 2.16: Sets of size 5 consisting 3 islands in C_9.

Let k be even, $S \subseteq \{1,\ldots,k\}$ a fixed set of size s, and $t < s$. Consider all $\prod_{i=1}^{k/2}(k+1-2i)$ different chorded k-cycles. If s and t have

different parities, no chorded k-cycle has exactly t chords connecting elements of S to elements of $\{1,\ldots,k\} - S$. If s and t have the same parity, the number of chorded k-cycles with exactly t spokes connecting S to $\{1,\ldots,k\} - S$ is

$$\binom{s}{t} \cdot \prod_{i=1}^{t}(k-s+1-i) \cdot \prod_{i=1}^{\frac{s-t}{2}}(s-t+1-2i) \cdot \prod_{i=1}^{\frac{k-s-t}{2}}(k-s-t+1-2i) .$$

Therefore, if a chorded k-cycle is picked uniformly at random, the probability that t spokes connect S to $V - S$ is:

$$\binom{s}{t} \cdot \prod_{i=1}^{\lfloor t/2 \rfloor} \frac{k-s+2-(t)_2-2i}{k+1-2i} \cdot \prod_{i=1}^{\frac{s-t}{2}} \frac{s-t+1-2i}{k-t+1+(t)_2-2i}$$

where $(t)_2$ denotes the remainder of t when divided by 2. Thus the probability that S is connected to $\{1,\ldots,k\} - S$ by fewer than αs spokes, $\alpha < \frac{1}{2}$, is bounded above by

$$\sum_{t=0}^{\lceil \alpha s \rceil - 1} \binom{s}{t} \prod_{i=1}^{\frac{s-t}{2}} \frac{s-t+1-2i}{k-t+1-2i} \leq$$

$$\lceil \alpha s \rceil \binom{s}{\lceil \alpha s \rceil - 1} \prod_{i=1}^{\frac{s-\lceil \alpha s \rceil + 1}{2}} \frac{s-\lceil \alpha s \rceil + 2 - 2i}{k-\lceil \alpha s \rceil + 2 - 2i} .$$

If $0 < s \leq \frac{k}{2}$ and S consists of $i < s$ islands, then at least $i+1$ elements of $\{1,\ldots,k\} - S$ are connected to elements of S via cycle edges. Therefore, any set of size $0 < s \leq \frac{k}{2}$ connected to fewer than αs elements of $\{1,\ldots,k\} - S$ consists of at most $\lceil \alpha s \rceil - 2$ islands. From the last lemma, there are at most $\sum_{i=1}^{\lceil \alpha s \rceil - 2} \left(\binom{s}{i}\binom{k-s-1}{i-1} + \binom{s-1}{i-1}\binom{k-s}{i} \right)$ such sets. For $\alpha \leq \frac{1}{2}$, this is at most $(\lceil \alpha s \rceil - 2)\binom{s}{\lceil \alpha s \rceil - 1}\binom{k-s}{\lceil \alpha s \rceil - 1}$.

Hence, for $\alpha < \frac{1}{3}$, the probability that a randomly chosen set of size $s \leq \frac{k}{2}$ will be connected to its complement by fewer than αs edges is at most:

$$\lceil \alpha s \rceil (\lceil \alpha s \rceil - 2)\binom{s}{\lceil \alpha s \rceil - 1}^2 \binom{k-s}{\lceil \alpha s \rceil - 1} \prod_{i=1}^{\frac{s-\lceil \alpha s \rceil + 1}{2}} \frac{s-\lceil \alpha s \rceil + 2 - 2i}{k-\lceil \alpha s \rceil + 2 - 2i}$$

$$\leq \alpha^2 s^2 \binom{s}{\lceil \alpha s \rceil - 1}^2 2^{\lceil \alpha s \rceil - 1} \prod_{i=1}^{\frac{s-3\lceil \alpha s \rceil + 3}{2}} \frac{s - \lceil \alpha s \rceil + 2 - 2i}{k - \lceil \alpha s \rceil + 2 - 2i}$$

$$\leq \alpha^2 s^2 \binom{s}{\lceil \alpha s \rceil - 1}^2 2^{\alpha s} \left(\frac{s - \lceil \alpha s \rceil}{k - \lceil \alpha s \rceil} \right)^{\frac{s-3\lceil \alpha s \rceil + 3}{2}}$$

$$\leq \frac{\alpha s}{2\pi(1-\alpha)} 2^{s\left(2h(\alpha) + \frac{5}{2}\alpha - \frac{1}{2}\right)} ,$$

where we used the inequalities

$$\binom{k-s}{\lceil \alpha s \rceil - 1} \prod_{i=\frac{s-3\lceil \alpha s \rceil + 5}{2}}^{\frac{s-\lceil \alpha s \rceil + 1}{2}} \frac{s - \lceil \alpha s \rceil + 2 - 2i}{k - \lceil \alpha s \rceil + 2 - 2i} \leq 2^{\lceil \alpha s \rceil - 1} ,$$

and (cf.[8])

$$\binom{s}{t} < \sqrt{\frac{s}{2\pi t(s-t)}} \cdot 2^{s \cdot h\left(\frac{t}{s}\right)} .$$

For $\alpha = .01$ we obtain that the probability there is a set of size s that is connected to fewer than $\frac{s}{100}$ elements in its complement is at most $\frac{s}{100} 2^{-s/3}$. Therefore, the probability that a chorded-cycle is not an expander is bounded above by $\frac{1}{100} \sum_{s=1}^{k/2} s 2^{-s/3} < 1$. We have proved:

Lemma 10 *For all even k, there is a $\frac{1}{100}$-expander chorded k-cycle.* □

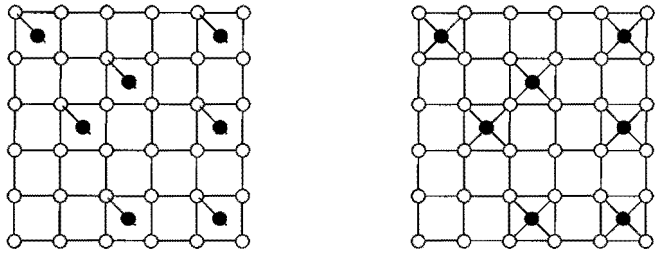
Lemmas 8 and 10 prove Lemma 6 and hence Theorem 5.

2.7 Alternative Models

We now discuss some variations of the PGA model and their effect on the achievable density. We heretofore assumed that the electrical elements are placed in the processing nodes. In most real PGA's they are placed in the 'squares' between the wires. This is modeled by the modified PGA graph shown in Figure 2.17(a).

Arguments similar to those used in Sections 2.5 and 2.6 can show that the same asymptotic results hold for such a PGA, too. Even if

each processing element were connected to all of its four neighboring nodes, as shown in Figure 2.17(b), the asymptotic densities would not change.



(a) (b)

Figure 2.17: PGA's with processors in the squares.

We assumed that a processing node cannot connect two wires even if the node is not used in the specific implementation. This is the most restrictive of several possible assumptions. The least stringent assumption allows a processing node to connect two wires even when the element it contains is used.

Relaxing this assumption increases the highest achievable density in both vertex and edge connectivity. For example, $\nu_{1,2}$ increases from 2/3 to 1. Yet the asymptotic results remain valid. This is clear in edge connectivity. To see that it holds in vertex connectivity, consider the chorded k-cycle with crossing number $\Omega(k^2)$ in the proof of Theorem 5. At most k of the crossings can correspond to edges passing through vertices of the chorded k-cycle. Still, there must be $\Omega(k^2)$ crossings between chords and rim edges. Hence the proof can proceed as before.

Allowing processing nodes to connect wires will not make a difference in the graph of Figure 2.17(a) since each processing node is connected to just one edge. Allowing the processing nodes of Figure 2.17(b) to connect wires will result in a PGA graph similar to that of Figure 2.18. This graph is discussed next.

Consider the graph of Figure 2.18. We show that any grid PGA with edge-connecting nodes can be simulated by a PGA based on the graph

of Figure 2.18 with vertex-connecting nodes with only a factor of four loss in density. Associate node (i, j) of the grid PGA with node $(2i, 2j)$ of the new PGA and leave all other nodes as switching nodes. It is not hard to see that each switching node of the original PGA together with three of its new neighbors can simulate an edge-connecting switching node. This is interesting because the PGA of Figure 2.18 uses only twice the number of wires used in a standard grid PGA. Yet even with *vertex* connectivity, it allows for a $\Theta\left(\frac{1}{k}\right)$ component density; viz., the same density which is achievable only with *edge* connectivity in standard grid PGAs. This occurs partly because the current PGA graph is not planar.

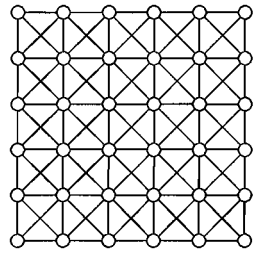


Figure 2.18: PGA consisting of the grid and the diagonal edges.

We assumed that each edge of the grid represents one wire. In practice, each edge corresponds to a *channel* consisting of several wires. This enables implementation of circuits with degree larger than four. Allowing a fixed number of wires in every channel increases the achievable density by a constant factor but the asymptotic results remain valid.

Perhaps the most critical departure of our model from real PGA's is that in practice each component represents a type of electrical element such as a gate or a memory cell. A real PGA contains many components of each type. A circuit design specifies the type of component that must be used and every component of that type will do.

One extreme version of the problem assumes that all components are of the same type. This is the traditional graph embedding problem discussed in paragraph 3 of [11] and the references therein. The user has

the freedom to choose the mapping both of the vertices and of the edges. We assumed the other extreme, namely that every component is of a different type. Hence the vertex mapping is given. The intermediate problem where there is a certain number of components of each type is interesting but was not dealt with here. So is the problem of realizing all circuits of a given size with a large PGA having only one type of component.

Acknowledgements

We are indebted to the referee who improved our upper bound on $\nu_{1,4}$ from 6/11 to 1/2, thereby establishing its exact value. We also thank Peter Shor for acquainting us with [1] and [9].

Bibliography

[1] A. Aggarwal, M. Klawe, D. Lichtenstein, N. Linial, and A. Wigderson, 'Multi-Layer Grid Embeddings', Proceedings of the 25th Annual Symposium on Foundations of Computer Science, 1985, pp. 186-195.

[2] A. Aggarwal, M. Klawe, P. Shor, 'Multi-Layer Grid Embeddings for VLSI', to appear, Algorithmica.

[3] F.R.K. Chung, 'Improved Separators from Planar Graphs', preprint.

[4] M. Cutler and Y. Shiloach, 'Permutation Layout', Networks, Vol. 8, 1978, pp. 253-278.

[5] A. El Gamal, J. Greene, J. Renyeri, E. Rogoyski, K. El-Ayat, A. Moshen, 'An Architecture for Electrically Configurable Gate Arrays', IEEE 1988 Custom Integrated Circuits Conference, pp. 15.4.1-15.4.5.

[6] L. Few, 'The Shortest Path and Shortest Road Through n Points', Mathematika 2, 1955, pp. 141-144.

[7] R. Freeman, 'User-Programmable Gate Arrays', IEEE Spectrum, December 1988, pp. 32-35.

[8] R.G. Gallager, *Information Theory and Reliable Communication*, John Wiley and Sons, 1968.

[9] H. Karloff, 'How Long Can a Euclidean Traveling Salesman Tour Be', Siam Journal of Discrete Math., Vol. 2, No. 1, pp. 91-99.

[10] F.T. Leighton, 'New Lower Bound Techniques for VLSI', Math Systems Theory 17, 1984, pp. 47-70.

[11] J.D. Ullman, *Computational Aspects of VLSI*, Computer Science Press, 1983.

Chapter 3

On the capacity of a binary channel with timing jitter at signal transitions modelled as a random walk

Abstract

Our interest is in the communication situation in which the time disparity between the receiver's clock and the transmitter's clock is modelled as a random walk. We obtain the channel capacity as a special case of capacity per unit cost ([15]).

Keywords *Timing jitter, Channel capacity, Capacity per unit cost, memoryless, random walk, Brownian motion*

3.1 Introduction

In his classical paper [1], Shannon mathematically described one-way communication and gave a single letter characterization of the infor-

[0]Published in IEEE Trans. on Inform. Theory, vol. IT-39, nr. 3, pp. 1064-1067, May 1993.

mation capacity of the general discrete memoryless (dm) one-way communication channel (OWC). In a communication system, as defined by Shannon, each channel input gives rise to one channel output. Transmitter and receiver are aware of the sequence number of the transmission, which can be regarded as a global clock to the system. The channel capacity expresses the rate at which communication is possible given the statistical uncertainty at the receiver's side about the channel inputs of the transmitter.

In many practical communication systems exact synchronization is either impossible or impractical. By the very nature of communication situations, transmitter and receiver are either physically remote or remote in the time dimension (e.g., storage applications). Oftentimes, the receiver recovers the transmitter time from the received sequence. Then, the receiver will not only have uncertainty about the amplitudes of the channel inputs, but also about their timing. Time uncertainty can affect the rate at which communication is possible.

Asynchronicity has been studied in the context of multi-user information theory. Willems [3] has solved the asynchronous Slepian-Wolf situation, and Verdu [4, 5] has studied asynchronicity for the multiple access channel. However, asynchronicity is commonly understood as the existence of a time-offset between different users that has a random value but is constant in time. This contrasts with timing jitter, for which time disparities fluctuate. The influence of timing jitter on pulse amplitude modulated systems and sampling has been dealt with in the literature [6, 7, 8]. However, the first attempt to quantify the effect of timing jitter in terms of information theoretical capacity functions was by Baggen and Wolf [2].

Consider a timing jitter channel with {0,1} binary input and output (Baggen and Wolf [2]). The bi-valuedness can be due to the transmission medium (e.g., compact disc: "pit"/"no pit") or can be due to a restriction to binary modulation and detection methods. Let $i \in \mathcal{N} = \{0, 1, 2, \ldots\}$ denote time according to the transmitter's clock and let $\overline{X} = (X_i | i \in \mathcal{N})$ represent the random input sequence. Similarly, let $j \in \mathcal{N}$ denote time according to the receiver's clock. The receiver samples the output of the channel at the ticks $j \in \mathcal{N}$ of its clock and thus obtains the output sequence $\overline{Y} = (Y_j | j \in \mathcal{N})$. Particular realisations of $\overline{X}$, $\overline{Y}$ will be denoted $\overline{x}$, $\overline{y}$, respectively (Random

variables are denoted in capital letters; realisations are denoted in ordinary letters).

As usual, a *0-run* of a sequence $\overline{x}$ of length $r > 0$ is a maximal subsequence $\{x_i, x_{i+1}, \ldots, x_{i+r-1}\}$ such that $\overline{x}$ equals zero on that set, i.e.,

$$\begin{aligned} & x_i = x_{i+1} = \ldots = x_{i+r-1} = 0, \\ & x_{i-1} = 1 \ \vee \ i = 0, \\ & x_{i+r} = 1. \end{aligned} \tag{3.1}$$

A *1-run* can be similarly defined, interchange 0 and 1. A *run* perse, is either a 0-run or a 1-run. By the maximality of a run, in any sequence 0-runs and 1-runs alternate. Any finite or semi-infinite binary sequence $\overline{x}$ can be uniquely described by its starting symbol x_0 and the sequence of the runlengths $\overline{r} = \{r_0, r_1, r_2, \ldots\}$ of its consecutive runs. The pair $(x_0, \overline{r}\,)$ is called the *runlength description* of the sequence $\overline{x}$. Consecutive runs are separated by transitions. For the sake of definiteness, if $x_i \neq x_{i-1}$ or $i = 0$, a transition is said to occur at time instant i (rather than at $i - 1$ or "$i - 1/2$").

Like [2], the time uncertainty is modelled by letting timing jitter shift the location of transitions in the output sequence with respect to the location of transitions in input sequence. In general, there is a one-to-one relationship between the locations of transitions and runlengths. The following assumption expresses that the channel does not loose any transitions (runs).

Assumption RS There is a one-to-one correspondence between $0 \rightarrow 1$ ($1 \rightarrow 0$) transitions in any finite input sequence, and $0 \rightarrow 1$ ($1 \rightarrow 0$) transitions in the corresponding output sequence.

Equivalently,

Assumption RS' There is a one-to-one correspondence between 0- (1-) runs in any finite input sequence, and the 0- (1-) runs in the corresponding output sequence.

From Assumption RS' it follows that the run index $n \in \mathcal{N}$ constitutes a common reference for the transmitter and receiver, which can serve as a clock if both operate in terms of runlengths. Such a binary timing

jitter channel (TJC) is denoted a TJC-RS (TJC with synchronisation on the level of runs, or 'Run Synchronisation' for short). The operation of a TJC-RS is readily defined using runlength descriptions of the input and output sequence. As any finite binary sequence has a runlength description, runlength descriptions can be used without any loss of generality. As we are interested only in the effects of timing jitter and not in amplitude uncertainty, it is assumed[1] that $Y_0 = X_0$. Denote the runlength sequences of $\overline{X}$, $\overline{Y}$ by $\overline{U}$ and $\overline{V}$, respectively. The effect of timing jitter is that $\overline{V}$ is a distorted version of $\overline{U}$. In order to define a TJC-RS, it remains to specify the probabilistic dependency of $\overline{V}$ on $\overline{U}$.

Note that the timing jitter equals the difference between receiver time and transmitter time. Therefore, the timing jitter S_n at the beginning of the n-th run (equivalently, at the n-th transition) is

$$S_n = \sum_{k=0}^{n-1} (V_k - U_k). \tag{3.2}$$

Obviously, in order for the order of time to be preserved by the channel and Assumption RS to hold, it is mandatory that[2]

$$V_k > 0, \text{ for all k.} \tag{3.3}$$

Baggen and Wolf consider a TJC-RS for which the $(S_n | n \in \mathcal{N})$ are defined to be independent random variables according to some given distribution P_S which lives on $\{s_{min}, s_{min}+1, \ldots, s_{max}\}$. The input sequences are restricted to be runlength-constraint in the sense that any run has length at least $d+1 > s_{max} - s_{min}$. Observe that the order of time is preserved and Assumption RS holds, since

$$V_n = S_{n+1} - S_n + U_n, \tag{3.4a}$$

$$V_n \geq s_{min} - s_{max} + d + 1 > 0. \tag{3.4b}$$

We will refer to this channel model as the BWTJC. Equation (3.4a) expresses that the transfer from U_n to V_n is governed by an additive

[1] As a consequence, for block length 1 any TJC can transmit 1 bit/transmission, which is the maximum possible.

[2] Channels for which V=0 is possible, are not run synchronous.

channel with noise component $D_n = S_{n+1} - S_n$. As the stochastic process $(D_n|n \in \mathcal{N})$ has memory, so far, only upper and lower bounds on the capacity function of the BWTJC have been obtained [2].

To the author the BWTJC suggested a different model, in which rather the $(D_n|n \in \mathcal{N})$ are defined as independent random variables according to some given distribution P_D. Then, the time jitter S_n becomes a random walk,

$$S_n = \sum_{k=0}^{n-1} D_n, \quad n \in \mathcal{N}. \tag{3.5}$$

Brownian motion, sampled at the transition times of the input sequence was put forward [10, 11] as a continuous time example of a time disparity process[3]. Because the runlengths are distorted by the difference of successive time disparities S_n,

$$V_n - U_n = S_{n+1} - S_n = D_n \tag{3.6}$$

definition of the timing jitter as a running sum of independent random innovation variables makes that successive runlengths are affected by statistically independent distortions. Thus, the transfer from U_n to V_n is governed by a dm OWC. This simplifies the analysis and allows for exact determination of the channel capacity. In accordance with communication practice and as exemplified by the sampled Brownian motion example, the statistics of the innovations can be allowed to depend on the actual input runlengths, without complicating the analysis. That is, for some distribution P^*:

$$\Pr\{V_n = v|U_n = u\} = P^*(v|u), \; n \in \mathcal{N}. \tag{3.7}$$

In our model, the binary memoryless increments (bmi) TJC-RS, the timing uncertainty introduced by the channel (without the effect of coding), increases as time grows to infinity, which is a marked difference with the Baggen-Wolf model, for which the timing jitter S_n remains bounded as n goes to infinity. Therefore, the BWTJC implicitly assumes some global time reference at the receiver which leaves

[3]Of course, the velocity of the receiver time with respect to the transmitter time should be constrained from being negative.

some local time uncertainty to be dealt with by the coding scheme. Our model does not assume an implicit time reference and models total time uncertainty. See Figure 1.

Since we are interested in nonterminating transmission using block codes, we fix the total number of input runs in any given code. As encoder and decoder are synchronized on the level of runlengths, this is a natural extension of block coding to the TJC. Determination of the capacity of the bmi TJC-RS is a special case of the capacity per unit cost, as defined by Verdu [15]. The capacity of the bmi TJC-RS equals the supremum of the mutual information of input runlength and output runlength, normalized by the expected input runlength. In [10], a slightly different definition of capacity is adopted that leads to a more self-contained presentation (average message bit error rate criterion instead of probability of decoding error).

Figure 3.1: Bmi TJC, time series

Figure 3.2: Bmi TJC-RS, runlength description

3.2 Definitions and preliminaries

As remarked in Section 1, the operation of bmi TJC-RS is most easily specified in terms of a runlength description $(X_0, \overline{U})$ of the input sequence $\overline{X}$ and a runlength description $(Y_0, \overline{V})$ of the output sequence $\overline{Y}$ of the channel. Throughout the paper, we will assume that the first bit is faithfully transmitted, i.e. $Y_0 = X_0$. Note that, generally, $\overline{X}$ and $\overline{Y}$ have different lengths but the same number N of runs. A bmi TJC-RS consists of two finite alphabets $\mathcal{U}$, $\mathcal{V}$, and a probability transition matrix $P^*(v|u)$. The alphabets $\mathcal{U}$ and $\mathcal{V}$ must be subsets of the set of positive natural numbers $\mathcal{P} = \{1, 2, \ldots\}$. The runs are indexed by $n \in \{0, 1, \ldots, N-1\}$.

The operation of the bmi TJC-RS can be considered in two equivalent ways:

- In terms of input sequence $\overline{X}$ and output sequence $\overline{Y}$,
- In terms of the runlength descriptions $(X_0, \overline{U})$ of the input and $(Y_0, \overline{V})$ of the output sequence.

The crucial matter is that, when considered in terms of runlength descriptions of the input, and output sequence, the bmi TJC-RS features in a well-known one-way communication system as defined by Shannon [1]. See Figure 2. (Prior to the communication of U_0, the first channel symbol X_0 of the input sequence is faithfully transmitted to the receiver.)

A code for bmi TJC-RS is defined by a tuple (N, M, T, P_e, e, d). These parameters denote the following:

N	number of runs in a codeword;
M	number of messages;
T	maximum number of bits transmitted;
P_e	average message error probability;
e	encoding function;
d	decoding function.

The source generates a message W, uniformly distributed over { 0 , ..., M - 1 }. A message is to be transmitted to the other side using at most T transmissions (input symbols). The total number of runs in

$\overline{X}$ and $\overline{Y}$ is fixed. The encoder is completely described by an encoding function e, that maps the message into the runlength description of the input sequence X,

$$(X_0, \overline{U}) = e(W). \tag{3.8}$$

All input runlength sequences with positive probability must satisfy the following time constraint

$$\sum_{n=0}^{N-1} u_n \leq T. \tag{3.9}$$

The decoder produces an estimate $\hat{W}$ of W, based on its knowledge of the sequence of received channel outputs $\overline{Y}$, that has runlength description $(Y_0, \overline{V})$, by means of a decoding function d:

$$\hat{W} = d(Y_0, \overline{V}), \ (Y_0 = X_0). \tag{3.10}$$

An (N, M, T, P_e, e, d) code for the bmi TJC-RS consists of an encoding and a decoding function such that the following average message error probability constraint is satisfied,

$$Pr\{\hat{W} \neq W\} = P_e. \tag{3.11}$$

A rate $R \geq 0$ is achievable for a bmi TJC-RS if for all $\varepsilon > 0$ there exists an T_0 such that for every $T \geq T_0$ there exists an (N, M, T, P_e, e, d) code with

$$r = \frac{log_2 M}{T} \geq R - \varepsilon, \tag{3.12a}$$

$$P_e \leq \varepsilon \tag{3.12b}$$

All logarithms in this paper are to the base 2. By definition, the set of all achievable rates is closed. The capacity of the bmi TJC-RS is defined as the maximum achievable rate.

3.3 Statement of result

Theorem 1 *The capacity of the bmi TJC-RS* $(\mathcal{U}, \mathcal{V}, P^*)$ *is given by the following single letter characterisation.*

$$C = \max\left\{R \mid 0 \leq R \leq \frac{I(U;V)}{E[U]}, \quad P_{UV} = P_U P^*_{V|U}\right\}, \tag{3.13}$$

$I(U;V)$ denotes the mutual information of the random variables U on $\mathcal{U}$ and V on $\mathcal{V}$, $E[.]$ denotes the expectation operator.

3.4 Proof of result

In terms of the runlength description of the problem, the capacity of the bmi TJC-RS with $X_0 \in \{0,1\}$ fixed at some definite value, corresponds to that of capacity per unit cost (Verdu [15]). The freedom of choice for X_0 can contribute to the rate of any scheme at most $1/T \leq 1/N$ bit/transmission, which vanishes asymptotically. Our result follows as a corrolary from Theorem 2 of [15].

3.5 Examples

To illustrate the use of the capacity formula, consider the following binary symmetric bmi TJC-RS,

$$\begin{aligned} \mathcal{U} &= \mathcal{V} = \{1,2\} \\ P^*(2|1) &= P^*(1|2) = p = 1-q \end{aligned} \tag{3.14}$$

for some fixed p, $0 \leq p \leq 1$. If $\alpha = P_U(1)$, then

$$C = \max\left\{ \gamma |\ \gamma = \frac{h(p+(q-p)\alpha) - h(p)}{2-\alpha},\ 0 \leq \alpha \leq 1 \right\} \tag{3.15}$$

This maximum can be evaluated for any p. Figure 3 shows how $I(U;V)$ $= h(p+(q-p)\alpha) - h(p)$ as a function of $E[U] = 2-\alpha$ is a shifted segment of the curve of the binary entropy function. The aforementioned maximum corresponds to the slope parameter of a tangient to this curve.

Consider a generalized bmi TJC-RS for which the input signal still is {0,1}-valued, but time is continuous. Let $X(t)$ denote this input random process. Transitions in $X(t)$ can happen at times $t \in \Re^+ \stackrel{def}{=} [0,\infty)$. Likewise, $(D_i|i \in \mathcal{N})$ and $(V_i|i \in \mathcal{N})$ may take on real values. Interestingly, if we assume that every interval $[i-0.5, i+0.5)$ contains at most one transition,

$$V_n \geq 1,\ n \in \mathcal{N}. \tag{3.16}$$

It makes no difference if we assume that the output $Y(t)$ is filtered and sampled,

$$Y_i = \int_{t=i-0.5}^{i+0.5} Y(t). \tag{3.17}$$

Because of constraint (3.16), $Y(t)$ is reconstructable from $(Y_i | i \in \mathcal{N})$. The capacity formula remains valid, however, with $\mathcal{U}, V \subset [1, \infty)$.

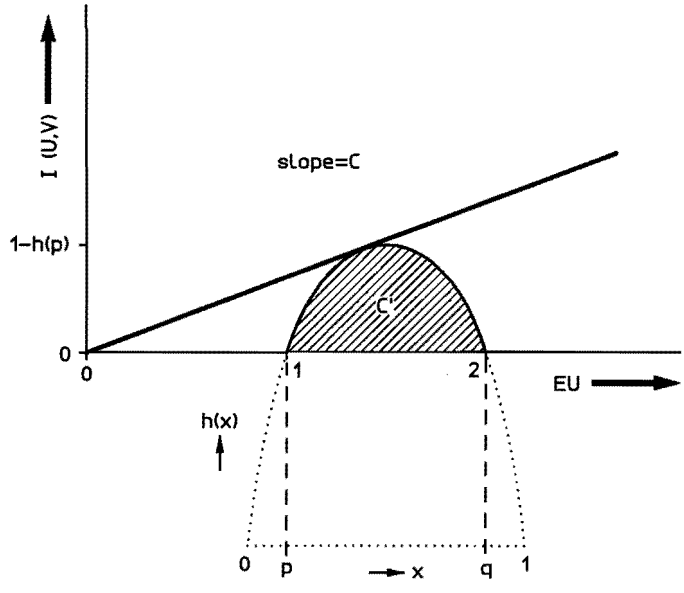


Figure 3.3: Illustration of capacity of example

3.6 Discussion

From our point of view, the Baggen-Wolf channel is a binary timing jitter channel with memory. As such, its capacity is more difficult to determine. Even if time goes to infinity and without the effects of coding, the distribution of the time disparity introduced by the channel

still has finite support. This is only possible if there is some (implicite) global time reference. Consider the replay of a disk or tape. The velocity of the replay can be modelled as random. This will give rise to a certain spectrum of the velocity. A discrete memoryless TJC makes the simplifying assumption that the velocity has a white spectrum (and more than that: is independently identically distributed). Thus, if we assume a Gaussian[4] distribution of the replay velocity, the receiver time disparity is a Brownian motion process. With discrete input runlengths, this Brownian motion process is sampled at random transition times of the input sequence, giving rise to a random walk process. A bmi TJC-RS for which $E[U] = E[V]$ is said to have *zero average jitter*, as the expected value of the timing jitter is zero (although the variance increases with time).

There is a straightforward generalization of the above capacity results to nonbinary memoryless increments (dmi) TJC-RS. For a dmi TJC-RS with A levels, $\log(A-1)$ must be added to the capacity per run. Another direct generalization is that to channels for which the output runlengths depend on the input runlengths and the channel symbol that makes up the runs: $P^*(v|u,x)$. Then, $I(U;V)$ is to be replaced by $I(U;V|X)$, with $P_X(0) = P_X(1) = 1/2$. For TJC that introduce amplitude uncertainties, there is a probability that successive runs by chance get the same amplitude, and thus merge into a single run. Such channels are no longer run synchronous.

Computation of the capacity of the bmi TJC-RS is a special case of computation of 'capacity per unit cost' as defined by Verdu [15]. Generalized Aritmoto-Blahut algorithms for the computation of capacity formulas $I(X;Y)/E[b(X)]$ exist in the literature (see [15]).

3.7 Conclusions

We conclude that a timing jitter channel that models timing jitter between transmitter and receiver as a random walk allows for exact determination of the channel capacity. This holds true even in the, from a practical point of view, very interesting case in which the innova-

[4]Of course, the velocity of receiver time with respect to transmitter time should be constrained from being negative.

tions of the timing jitter depend on the lengths of the input runs. In our approach, the Baggen-Wolf channel is a timing jitter channel with memory, whose capacity remains an open problem. Another interesting toppic for further research is obtained by the combination of time and amplitude uncertainty. Such channels are no longer run synchronous.

3.8 Acknowledgements

The author would like to acknowledge the very illuminating introduction to timing jitter by Stan Baggen and J. Wolf [2]. Furthermore, the author would like to thank Frans Willems of the Eindhoven University of Technology, the Netherlands, for interesting discussions on the subject as well as his reference to the recent work of Verdu [15]. The author would also like to thank his colleague Johan van Tilburg and an anonymous referee for useful comments on earlier versions of the manuscript.

Bibliography

[1] C. Shannon and W. Weaver, "*The Mathematical Theory of Communication,*"" Univ. Illinois, Press, 1949.

[2] S. Baggen and J. Wolf, "An Information Theoretic Approach to Timing Jitter,"" *11th Symp. on Inform. Theory in the Benelux*, pp. 174-180, Noordwijkerhout, the Netherlands, Oct. 1990.

[3] F. Willems, "Totally Asynchronous Slepian-Wolf Data Compression,"" *IEEE Trans. Inform. Theory*, Vol. IT-34, No. 1, pp. 35-44, Jan. 1988.

[4] S. Verdu, "Capacity Region of the Symbol-asynchronous Gaussian Multiple-Acc. Channel,"" *IEEE Trans. Inform. Theory*, Vol. IT-35, No. 4, pp. 733-51, July 1989.

[5] S. Verdu, "Multiple-access channels with memory with and without frame synchronism,"" *IEEE Trans. Inform. Theory*, Vol. IT-35, No. 3, pp. 605-19, May 1989.

[6] T. Berger and D. Tufts, "Optimum pulse amplitude modulation; Part II: Inclusion of timing jitter,"" *IEEE Trans. Inform. Theory*, Vol. IT-13, pp. 209-16, 1967.

[7] P. Vogel, "PAM-Systems Including Timing Jitter; A Game-Theoretical Approach,"" *Arch. Elektron. Uebertr.techn.*, Vol. 40, No. 3, pp. 1653-8, June 1986.

[8] Y. Tankik, "Comments on "Pulse Shape, Excess Bandwith, and Timing Sensitivity in PRS,"" *IEEE Trans. Commun.*, Vol COM-37, No. 8, Aug. 1989.

[9] G. Agrawal and T. Shen, "Power Penalty Due to Decision-Time Jitter in Optical Communication Systems,"" *Electron. Letters*, Vol. 22, No. 9, pp. 450-1, Apr. 1986.

[10] A. Hekstra, "*The capacity of the discrete memoryless timing jitter channel and its capacity in the case of weak synchronisation,*" Netherlands PTT Research, Report 975/90, November 1990.

[11] A. Hekstra, "The capacity of the discrete memoryless timing jitter channel and its capacity in the case of weak synchronisation,"" *12th Symp. on Inform. Theory in the Benelux*, pp. 9-16, Veldhoven, the Netherlands, May 1991.

[12] Gallager, "*Information Theory and Reliable Communiation,*"" Wiley, New York, 1968.

[13] R. Blahut, "*Principles and Practice of Information Theory,*"" Addison-Wesley, Amsterdam, 1987.

[14] I. Csiszar and J. Körner, "*Coding Theorems for discrete memoryless systems,*"" Academic Press, New York, 1981.

[15] S. Verdu, "On Channel Capacity per Unit Cost,"" *IEEE Trans. Inform. Theory*, Vol. IT-36, No. 5, 1990.

Chapter 4

An alternative to metric rescaling in Viterbi decoders

Abstract

In the Viterbi algorithm, the negative log-likelihood estimates, accumulated distances, or path metrics are unboundedly increasing functions of time. For implementation, all variables must be confined to a finite range. The following properties of the Viterbi algorithm can be exploited. 1) Path selection depends only on differences of metrics. 2) The difference between metrics is bounded. In the *rescaling scheme*, at each iteration the minimum metric is subtracted from all metrics. The use of *two's complement arithmetic* is proposed as an alternative to the rescaling method. Surprisingly this scheme avoids any kind of rescaling subtractions. Obvious advantages in implementation are hardware savings, and a speedup inside the metric update loop, which is critical to the decoder's computational throughput. Although the described technique appears to be known, it has not yet been published in the open literature.

[0]Published in IEEE Trans. on Commun., Vol. COM-37, pp. 1220-1222, Nov.1989.

4.1 Introduction

This correspondence presents an efficient implementation of the metrics in the well-known Viterbi algorithm (VA) [1]. Whereas the principles and performance of the Viterbi algorithm are very well described in the literature, we agree with Rader [3] that some of the tricks that ease its implementation are not widely known. The technique described in this paper has been discovered independently by others, but has not been published in the open literature. Other approaches to the implementation of the metrics that have been studied in the literature are: to rescale the metrics, as described below [7]- [9]; to sum only a certain number of most recent branch metrics [10] (a fully analog implementation is considered).

In its most general form, the VA may be viewed as a solution to the problem of maximum a posteriori probability estimation of the state sequence of a finite-state discrete-time Markov chain with memoryless output noise (Forney [2]). However, for our purposes it will suffice to consider the VA on a purely mechanical level. More elaborate descriptions can be found in references [1]-[6]. For implementation, a version of the VA needs to be found that is a finite state machine. In the VA there are two kinds of variables: sequence or path variables, and their metrics. Rader [3] presents an elegant solution to the path representation problem. This correspondence addresses the representation of metric information in the VA. It is shown that the input/output behavior of the VA is unaffected by the application of a modulo operator to all metric variables, when the range of the modulo operator is sufficiently large and approximately symmetric around zero. This modulo operator corresponds to the overflow mechanism in two's complement arithmetic and therefore has no hardware cost. Two properties of the VA will emerge: operation of the VA depends only on differences of metrics, and the difference between metrics is bounded. The well-known rescaling approach to the implementation of the VA is to subtract the minimum metric from all metrics, so that they all remain in a range defined by property II. The correctness of the two's complement and rescaling modifications of the VA is derived from the afore mentioned properties. Finally, the complexities of the resulting implementations are observed, and a comparison is made.

4.2 Description of the Viterbi Algorithm

Let $k \in N = \{1, 2, \cdots\}$ represent time, and let $s \in S, t \in S$ be states. The *path metric* of a state s at time k is denoted $M_s(k)$. The so called *branch metric* pertaining to the single step transition from a state s at time $k-1$ to a state t at time k is denoted $b_{st}(k)$. The branch metrics depend on the received sequence that is the input to the decoder. However, notationally this is omitted. The generic update equation of the VA reads

$$m_s(k) = min\{m_{st}(k) | s \in S\}, \quad t \in S, k \in N \tag{4.1a}$$

where $m_{st}(k)$ is the candidate metric for state t at time k associated with a transition from state s at time $k-1$ to state t at time k:

$$m_{st}(k) = m_s(k-1) + b_{st}(k), \quad s \in S, t \in S, k \in N. \tag{4.1b}$$

At the initial time $k = 0$, the values of the path metrics are zero. In typical applications, the range of the minimization in (1a) can be limited to a set of states s for which the transition to the state t is admissible. For instance, if states correspond to the contents of a shift register machine, state transitions must be producable by the pertaining shift operator. Formally, metrics of state transitions that are inadmissible can be thought of as infinite. The set of admissible single step state transitions (s,t)-from s to t, $s \in S, t \in S$ - is denoted T. For simplicity, T is assumed to be constant in time.

Along with the computation of the metrics, for each state, the algorithm keeps track of a path that leads to it. If a state s achieves the minimum candidate metric for a given state t and time k in (1a), s is called the *precursor* of t at time k. In principle, the algorithm has the values of the precursors, for all states and times, stored in memory. Going backwards like this, for each state t and time k, a sequence of states called the *survivor path* $p_t(k)$ of t is defined. A survivor path ends in the state t that corresponds to it and, theoretically extends all the way back to the initial time zero. Observe that if an arbitrary survivor path that ranges over a time span $\{0, \cdots, k\}$, is truncated to a path over $\{0, \cdots, i\}$, for some $i \leq k$, this again yields a survivor path. Another property of survivor paths –that is also readily verified– is that of all

the paths that end in a certain state, the survivor path of that state has the minimum metric. Here, the metric of a path is defined as the sum of the branch metrics of its single step state transitions. Theoretically, the *output* of the VA is a path with the minimum overall metric in this sense. It is the survivor path of a state with the minimum metric. Furthermore, in certain applications, the difference between the minimum path metric and other metrics is used as a reliability indicator in a synchronization loop.

4.3 Two Properties of the Vitervi Algorithm

The following two properties of the Viterbi Algorithm can be exploited at its implementation.

Property I: The output of the VA depends only on differences of metrics.

The selection of survivor paths, as governed by (1), involves only comparisons of candidate metrics $m_s(k-1) + b_{st}(k)$ and, hence depends only on differences of metrics. Reliability indicators that are not based on metric differences are left out of consideration.

Property II: The difference between metrics is bounded.

By assumption, S is a finite set. Let B be an upper bound for the absolute values of the finite branch metrics:

$$|b_{st}(k)| \leq B, \quad (s,t) \in T, k \in N. \tag{4.2}$$

From hereon, T is interchangeably considered as a subset of $S \times S$ and as a $0-1$ matrix on $S \times S$. A sufficient condition for Property II to hold is, that T, when raised to some finite power n, has all its entries strictly positive. Such a T could be called *irreducible and aperiodic.*

Proof: Let t_1 and t_2 be arbitrary states at time k. Furthermore, let p_1 be an abbreviation for $p_{t1}(k)$, the survivor path of t_1 at time k. Without loss of generality, $m_{t1}(k) \leq m_{t2}(k)$. In case k is less than n, q_2 is set equal to the survivor path of t_2 at time k. Otherwise, there

exists a state s at time $(k-n)$ on the survivor path of t_1. The segment of p_1 that starts at time zero and ends at time $(k-n)$ is denoted q (this is $p_s(k-n)$). By the assumption on T, there is an extension of q, along branches from t, to a path that ends in t_2 at time k. Provided k is greater than or equal to n, let this path be q_2. In either case, path q_2 ends in t_2, and therefore has a metric that is not less than $m_{t2}(k)$. Thus, the difference $(m_{t2}(k) - m_{t1}(k))$ is upper bounded by the metric of q_2 minus the metric of the survivor path of t_1. In an expansion of the metrics of q_2 and p_1 as sums of branch metrics, all but the last min $\{k, n\}$ pairs of terms cancel in their difference, so that with $m = 2n$,

$$|m_{t1}(k) - m_{t2}(k)| \leq mB, \quad t_1 \in S, t_2 \in S, k \in N. \tag{4.3}$$

In convolutional decoder applications, the entries of T^n are positive for n equal to the memory order of the code [6]. Tightness of the upper bound in (3) is important to all implementations that do not allow metric overflow. Often, the sharper bound $m = n$ can be attained. For instance, in hard-decision convolutional decoder applications this is the case. Assume that $b_{st}(k)$ is of the form $d(e(s,t), r)$ where s is the presumed state at time $k-1$, t is the presumed state at time k, $e(s,t)$ is the output symbol that corresponds to a transition from state s to state t, and r is the output symbol received at time k. It is sufficient that d satisfies the triangle inequality of a distance function for the sharper bound to hold. The proof above shows that the difference of any two "state metrics" is upper bounded by a sum of at most n terms, each term consisting of a difference of two branch metrics. A term is of the form $d(e(s_2,t_2),r) - d(e(s_1,t_1),r)$ where s_2, t_2 and s_1, t_i are successive states on q_2 and p_1, respectively. By the triangle inequality, the absolute value of each term is bounded above by $d(e(s_2,t_2), e(s_1,t_1))$. From this $d = n$ follows.

Metrics large in comparison to the minimum metric correspond to unlikely survivor paths. The degradation in decoding performance that results from not faithfully representing large metric differences is often small [5], [9]. Thus, it is sufficient for Property II to hold in a probabilistic sense.

4.4 The rescaling approach

By Property I, subtraction of a constant from a metric vector $(m_s(k)|s \in S)$ does not affect the output of the VA. Let the *rescaling functions* $r_k, k \in N$, and the minimum metric c_k be defined as

$$r_k(x) = x - c_k, \tag{4.4a}$$

$$c_k = \min\{m_s(k)|s \in S\}, \quad k \in N. \tag{4.4b}$$

In a rescaling implementation of the VA, the metrics $m_s(k)$ are replaced by rescaled versions $r_k(m_s(k))$. By Property II the rescaled metrics satisfy

$$0 \leq r_k(m_s(k)) \leq mB \quad \text{for all } s \in S, k \in N. \tag{4.5}$$

The *algorithmic equation* (1) turns into

$$r_k(m_t(k)) = \min\{r_{k-1}(m_s(k-1)) + b_{st}(k)|s \in S\} - (c_k - c_{k-1}). \tag{4.6}$$

Note from (6) that intermediate values up to $(m+1)B$ can occur. The most negative number that can occur is $-B$. Once $(c_k - c_{k-1})$ has been subtracted, (5) applies again. If the metrics are nonnegative, the required numerical range can be reduced accordingly. Otherwise, the numerical range will generally have to be chosen symmetrically. Then c_k can be an arbitrary metric.

The depth of the comparison tree for the determination of the minimum metric is proportional to $\log_2(|S|)$. This implies that in all but those cases in which there is no serious constraint on the computational throughput of the decoder, modifications of the above scheme are necessary. For instance, c_k can be redefined to be an arbitrary metric. The numerical range may increase. Some residual delay due to rescaling operations will be difficult to avoid. None of this is necessary.

4.5 Two's compl. arithmetic approach

The key idea to improvement is not to invest in avoiding overflow, as in the rescaling approach, but instead to accommodate overflow in such

a way that it does not affect the correctness of the results. *Two's complement arithmetic*, in c bits, refers to the additive group over

$$F_c := \{-2^{c-1}, 1-2^{c-1}, \cdots, 2^{c-1}-1\} \tag{4.7}$$

where addition is defined modulo 2^c. The modulo operator which reduces a number to an element of the interval F_c is denoted "mod2^c". Modification of the VA for the purpose of an implementation based on two's complement arithmetic entails replacing the metrics $m_s(k)$ by their residuals $m_s(k) \bmod 2^c, s \in S, k = 0, 1, \cdots$. As noted in Property I, the computation of the minimum in the right-hand side of (1) is performed by comparison of elements. It is the signs of all differences

$$m_s(k-1) + b_{st}(k) - (m_{s'}(k-1) + b_{s't}(k)), s \in S, s' \in S, t \in S, k \in N \tag{4.8}$$

that matters. Substitution of the modulo-2^c reduced metrics in the right-hand side of (1) will lead to the evaluation of the differences

$$m_s(k-1)+b_{st}(k)-(m_{s'}(k-1)+b_{s't}(k)) \bmod 2^c, s \in S, s' \in S, t \in S, k \in N. \tag{4.9}$$

By Property II, (8) does not exceed $(m+2)B((n+1)B$ if the branch metrics are nonnegative. Hence, if the numerical range is at least $\{-(m+2)B, \cdots, (m+2)B\}$, that is

$$2^{c-1} - 1 \geq (m+2)B \tag{4.10}$$

the reduced difference (9) equals the true difference (8). Then, Property I implies the correctness of the two's complement modification.

In principle, there is a worst case one-bit penalty in terms of data width over the rescaling option. However, such a penalty can be avoided by a small compromise in the range of the branch metrics or in the probability of error due to the occurrence of metric overflow. If the branch metrics are nonnegative, the required numerical range is actually the same as for the rescaling option.

4.6 Conclusions

The VA has the property that its behavior is left invariant when a modulo operator is applied to the metrics, provided that its range is

sufficiently large and approximately symmetric around zero. This modulo operator corresponds to the overflow mechanism in two's complement arithmetic and therefore has no hardware cost. In the worst case, the two's complement option will need a small compromise in the metric branch range or in the probability of overflow so as to preserve the width of the data paths. We conclude that the use of two's complement arithmetic to accomodate metric overflow in the VA offers significant advantages in implementation, in terms of design simplification and computational throughput.

Acknowledgement

The above results have been obtained during practical work [11] at Eindhoven University of Technology (EUT), the Netherlands. carried out in partial fulfillment of the requirements for the "Ingenieur" degree. The author would like to express his gratitude to Prof. J. Arnback and Mr. A. P. Verlijsdonk of the Telecommunictions Group, as well as Dr. J. Vinck of the Information and Conununication Theory Group of the School of Electrical Engineering at EUT for their stimulating supervision.

Finally, the author would like to thank Prof. T. Berger and Prof. C. Heegard of Comell University for their encouragement to publish this simple but useful result.

Bibliography

[1] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding adgorithm", *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[2] G. D. Forney, Jr., "The Viterbi algorithm", *Proc. IEEE*, vol. 61, pp. 268-278, Mar. 1973.

[3] C. M. Rader, "Memory management in a Viterbi decoder", *IEEE Trans. Commun.*, vol. COM-29, pp. 1399-1401, Oct. 1981.

[4] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding.* New York: McGraw-Hill, 1979.

[5] G.C. Clark and J.B. Cain, *Error Correction Coding for Digital Communication.* New York: Plenum, 1981.

[6] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*, London, England: Prentice-Hall, 1983.

[7] J. Conan, "An F8 microprocessor-based breadboard for the simulation of conununication links using rate 1/2 convolutional codes and Viterbi decoding." *IEEE Trans. Commun.*, vol. COM-31, pp. 165-171, Feb. 1983.

[8] Y. Yasuda et al., "Development of variable-rate Viterbi decoder and its performance characteristics." *Int. Conf. Digital Satellite Commun.*, Phoenix, AZ, pp. 19-23. IEEE, 1983.

[9] J. Cain, "CMOS VLSI implementation of $R = 1/2$, $K = 7$ decoder," *IEEE Nat. Aerospace Electron. Conf. NAECON 1984*, vol. 1, pp. 20-27.

[10] A. S. Acampora and R. P. Gilmore, "Analog Viterbi decoding for high speed digital satellite channels", *IEEE Trans. Commun.*, vol. COM-26. pp. 1463-1470, Nov. 1978.

[11] A. P. Hekstra, "Feasibility study of a Rate-1/2, 4096 KBit/s convolutional codec," Rep. Practical Training at the Communications Group tn the Depart. of Elec. Eng., Eindhoven Univ. Technol., The Netherlands, 1984.

Chapter 5

On the numerical range of the path metrics in a binary Viterbi decoder

Abstract

This paper addresses the numerical range of the path metric calculus in the well-known Viterbi algorithm (VA). Given a binary convolutional code, we present an exact expression for the maximum difference of (candidate) path metrics. We prove that the maximum difference corresponds to the case of noiseless reception of codewords, a trivial mode of operation. We show that for any convolutional code there is a constant B, such that states with path metrics exceeding the minimum path metric by more than B can never be on the output path of the VA. These so called 'looser states' are deleted from the trellis by means of a stopping rule. Comparison of tight bounds for the required numerical range shows that the stopping rule can at best half the required

[0]This paper has been submitted to IEEE Trans. on Inform. Theory, Sept. 1993. Parts of this paper were presented at the 13th Symp. on Inform. Theory in the Benelux, Enschede, the Netherlands, June 1-2, 1992, at the 1993 IEEE Intern. Symp. on Inform. Theory, San Antonio, USA, Jan. 17-22, 1993, and at the 14th Symp. on Inform. Theory in the Benelux, Veldhoven, The Netherlands, May 17-18, 1993, the 1994 IEEE Intern. Symp. on Inform. Theory, Trondheim, Norway, June 27-July 1, 1994.

numerical range.

Keywords

Viterbi algorithm, reduced complexity, path metrics, survivor paths.

I - INTRODUCTION

The path metrics, negative log-likelihood variables or accumulated distance functions in the Viterbi algorithm (VA) are increasing, unbounded functions of time. However, the survivor path selection in the VA depends only on differences of candidate path metrics. As usual, by a *state* we refer to a node in the trellis. A state is determined by an encoder state (contents of the encoder shift register) and the depth in the trellis. A *path* is a series of connected states in the trellis, that connects the initial 'all zeroes' state with some state at the final stage of the trellis. The term *path* is also used to indicate the sequence of output symbols that corresponds to a path as defined before. A *candidate path metric* (cpm) for a certain state S is the sum of the path metric of some state at an incoming branch to state S in the trellis plus the corresponding branch metric. Only these differences of candidate path metrics need to be faithfully reproduced in an implementation. By the VA, the path metric of a state equals the minimum of all its candidate path metrics. Due to the structure of the trellis, the maximum difference of candidate path metrics is bounded.

A possible implementation of the path metrics in the VA is to subtract, at each stage of the trellis, the minimum path metric from all path metrics. With this so called *'rescaling'* approach, the required numerical range must be large enough to represent the maximum difference of any two candidate path metrics for any two nodes at a given depth in the trellis. The latter maximum is denoted $\max\{\Delta cpm\}$. Consequently, the minimum candidate path metric is reduced to zero, and the maximum value that occurs is $\max\{\Delta cpm\}$. When processing units work with b-bits nonnegative integers $\{0, 1, \ldots, 2^b - 1\}$, for correct

operation of the decoder it is required that

$$2^b - 1 \geq \max\{\Delta cpm\}. \tag{5.1}$$

When the arithmetic units allow for positive and negative integers, $\{-2^{b-1}, -2^{b-1}+1, \ldots, 2^{b-1}-1\}$ we have that one additional bit is required,

$$2^{b-1} - 1 \geq \max\{\Delta cpm\}. \tag{5.2}$$

An alternative to rescaling is *modulo arithmetic* with a numerical range symmetric around zero e.g. two's complement representation [2]. Provided the maximum possible difference of any two candidate path metrics *for the same node*, $\max_1\{\Delta cpm\}$, fits inside the range of the modulo operator, the modulo reduced difference of the candidate path metrics equals the true difference and the correct survivor paths are selected. Denote the maximum difference of any two path metrics by $\max\{\Delta pm\}$. If the output path of the VA is traced back [3] from the state with minimal metric, also the difference of any two path metrics must fit inside the numerical range. In formula,

$$2^{b-1} - 1 \geq \max\{\max_1\{\Delta cpm\}, \max\{\Delta pm\}\}. \tag{5.3}$$

A clear advantage of modulo arithmetic is that it saves the computation time and hardware associated with the subtraction of the minimum path metric from all other path metrics. If a rescaling implementation uses only nonnegative integers, there is a one bit penalty in required data width for modulo arithmetic. When inequality occurs in

$$\max\{\max_1\{\Delta cpm\}, \max\{\Delta pm\}\} \leq \max\{\Delta cpm\}, \tag{5.4}$$

that penalty for modulo arithmetic may be reduced.

The results presented in this paper apply to all hard decision Viterbi decoders, i.e. Viterbi decoders that use the Hamming distance as metric function as well as to soft decision decoders that satisfy the following rather weak assumptions. For a rate $R = k/n$ binary convolutional code, we assume the following.

Assumptions

I - Symbol metrics are nonnegative integers.

II - If transmission is over a noiseless channel, the symbol metric equals either s_{max} or zero depending on whether or not the received bit equals the hypothesis bit.

III - A branch metric is the sum of n symbol metrics; a path metric is the sum of the branch metrics along the path.

With respect to Assumption I, note that what counts for the survivor path selection is the difference of the symbol metric given that a '0' was sent minus the symbol metric given that a '1' was sent. Figure 1 illustrates that this difference can be negative, even though the symbol metrics satisfy Assumption I. Assumption II merely states that the maximal (s_{max}) and minimal symbol metric (0) occur.

Throughout the paper, we assume a realization of the encoder as a parallel combination of k shift registers $i = 1, 2, \ldots, k$ for which the i-th shift register is m_i cells long [4]. Let M denote the total number of shift register cells in the encoder, i.e.

$$M = \sum_{i=1}^{k} m_i.$$

The number of states in the trellis equals 2^M. The memory order m is defined as the maximum among the m_i's [4]. Similarly, define m_- as the minimum length of any of the encoder shift registers. A *route* is a series of connected states in the trellis. Observe that any two states that are m stages apart in the trellis are connected by some route in the trellis. A route is a segment of a path.

Given a particular binary convolutional code, Theorem I gives exact expressions for $\max\{\Delta pm\}$, $\max_1\{\Delta cpm\}$ and $\max\{\Delta cpm\}$. These expressions are readily evaluated (the approach of [9] for nonbinary convolutional codes requires linear programming). By $\max_L\{\Delta pm\}$, $\max_{1,L}\{\Delta cpm\}$ and $\max_L\{\Delta cpm\}$ we indicate the maximum (candidate) path metric difference at depth $L = 0, 1, \ldots$ into the trellis. At $L = 0$, the trellis starts with the 'all zeroes' state. It turns out that as a function of L, the maxima first increase, then decrease, and finally settle at some level.

As the path metrics can be interpreted as log-likelihood functions, large path metric differences indicate a strong discrimination between candidate output sequences of the VA. Intuitively, we therefore expect large path metric differences to occur if the transmissions are over an ideal channel. We prove in Theorem I that the maximum (candidate) path metric differences occur in the case of noiseless reception of the 'all zeroes' codeword and comparison of the 'all zeroes' path as candidate survivor path for the 'all zeroes' state with a (candidate) survivor path of maximum Hamming weight. We show that the worst case (candidate) path metric differences occur at a depth of $m_- + 1 \leq L \leq m + 1$ in the trellis ($m_- \leq L \leq m$ for $\max\{\Delta pm\}$). With these simplifications, the aforementioned maxima are easy to determine.

In an application for which the VA works on finite blocks of data, the required numerical range is maximized over L in order to avoid a loss of performance. However, in applications in which the VA works on an, in principle, infinite data stream, it suffices to consider the limit values as L increases to ∞ in Theorem I. Evaluation of Theorem I for a given binary convolutional code indicates at what value of L, the maximum (candidate) path metrics settle at a fixed level. The level L_∞ at which this occurs can be bounded. We use the notation $\max_{1,\infty}\{\Delta cpm\}$, etc., to indicate the limit values.

As stated before, large path metric differences indicate a strong discrimination between survivor paths. It turns out that in such a situation, for many states with 'large' path metrics it can be ruled out that the output path of the VA runs through those states. Here, a path metric of a state is called 'large' if the path metric exceeds the minimum path metric at the given depth in the trellis by more than some constant B that depends on the given binary convolutional code. Theorem II formulates a stopping rule that deletes all states with a 'large' metric, as for any path that runs through such a state and for any received sequence, a detour exists via the state that has minimal path metric such that the resulting path has a smaller metric than the original path. Hence, the required numerical range can be reduced. A worst case example for a 'large' path metric occurs with noisefree reception of the 'all zeroes' codeword up to a certain time, followed by the reception of some other codeword that continues from some nonzero state and most quickly returns to the 'all zeroes' path in terms

of Hamming distance.

The stopping rule requires the computation of the difference of the path metrics with the minimum path metric, and is most logically combined with the rescaling method. Application of the principle behind the stopping rule to modulo arithmetic and trellis codes under further study. Theorem III gives an exact analysis of the numerical range of the path metrics for the VA with stopping rule. The approach of Theorem I is extended to include the B-constraint. The results of Theorem I and III are precise, and require an evaluation that depends on the generators of the binary convolutional code used. Theorem IV gives an upper bound on $\max_{\infty}\{\Delta pm\}$ in terms of n, m, k and d_{free}. This upper bound is a generalisation of a recent result by Alston and Chau [1].

II - STATEMENT OF THEOREMS, EXAMPLES AND LEMMAS

Throughout the paper, the Viterbi decoder satisfies Assumptions I-III and we consider only binary convolutional codes. The linearity of the convolutional encoder is essential in our proofs. The trellis starts from the initial encoder state 'all zeroes'. A state in a trellis of a given finite length is called *final*, if it is located at the final stage of the trellis.

The *output path* of the VA is a path that has the minimal overall metric value for the entire codeword length considered. By $pm(S)$ we denote the *path metric* of state S. The *Hamming metric* $Hm_L(S)$ of an encoder state S at depth L, is the path metric of that state, if

- the decoder were of the hard decision type,
- the Hamming distance is used as metric function, and
- the received sequence were 'all zeroes'.

In other words, the Hamming metric is the Hamming weight of the 'lightest' path from the 'all zeroes' starting state of the trellis. Here, 'lightest' refers to minimal Hamming weight. A *candidate Hamming metric* for a state, is a candidate path metric, if the decoder were of the

hard decision type, etc. A candidate Hamming metric for a state S is the sum of a Hamming metric of a state at an incoming branch to state S in the trellis plus the Hamming weight of that branch. Because the Hamming metrics are just path metrics, they are readily determined with the VA.

Theorem I

For a Viterbi decoder that satisfies Assumptions I-III, we have

- $\max_L\{\Delta pm\}$ *equals* s_{max} *times the maximum Hamming metric of a final state in a trellis* L *branches deep,* $\max_L\{Hm\}$,
- $\max_{1,L}\{\Delta cpm\}$ *equals* s_{max} *times the maximum candidate Hamming metric for the final 'all zeroes' state in a trellis* L *branches deep,* $\max_{s=0,L}\{cHm\}$,
- $\max_L\{\Delta cpm\}$ *equals* s_{max} *times the maximum of all candidate Hamming metrics of all final states in a trellis* L *branches deep,* $\max_L\{cHm\}$.

More information about the distribution of the path metrics is obtained from

$$|pm(S) - pm(T)| \leq s_{max} Hm_L(S - T), \tag{5.5}$$

for arbitrary states S *and* T *at depth* L. *For maximization over the depth* L *into the trellis, w.l.o.g.* $m_- \leq L \leq m$ *for* $\max\{\Delta pm\}$, *and* $m_- + 1 \leq L \leq m + 1$ *for* $\max_1\{\Delta cpm\}$ *and* $\max\{\Delta cpm\}$.

The paths provided by Theorem I can actually occur during the operation of the Viterbi decoder. By Assumption II, in case of noiseless reception of the 'all zeroes' codeword, the path metric of a state equals s_{max} times its Hamming metric. Hence, in the proof of Theorem I, we only need to show that no larger (candidate) path metric differences than specified can occur. The folllowing example illustrates that the three maxima in Theorem I can differ (the convolutional code is not a good code to use in practise).

Example I Consider the $R = 1/2$, memory order $m = 2$, binary convolutional code with generators (notation of [4]), and hard decision decoding ($s_{max} = 1$),

$$g^{(1)} = (1\ 0\ 1),\ g^{(2)} = (1\ 0\ 0).$$

Figure 2 illustrates the evaluation of $\max\{\Delta pm\} = 4$, $\max_1\{\Delta cpm\} = 3$, $\max\{\Delta cpm\} = 5$.

For block based applications of the VA, one only needs to evaluate the (candidate) Hamming metrics at depths of $L \leq m+1$ in the trellis. For stream based applications, values of $L \geq m+1$ give an upper bound for the required numerical range. Consider e.g. the evaluation of $\max_\infty\{Hm\}$. The Hamming metric of any encoder state at depths $L \geq m$ in the trellis can only decrease with the depth as a trellis of length L contains a subtrellis of length $(L-1)$, prefixed with a 'all zeroes' branch. Denote by L_∞ the depth in the trellis at which the (candidate) Hamming metrics have settled at their final value. The value of L_∞ can be upper bounded as follows. No survivor path visites a certain encoder state more than once. Therefore, the total number of iterations necessary is at most 2^M, the number of states in the trellis. Out of the 2^M states, a fraction 2^{-n} has an outgoing branch labelled with 'all zeroes'. The Hamming weight of any survivor path is upper bounded by $\max_m\{Hm\}$. Therefore,

$$L_\infty \leq 2^{(M-n)} + \max_m\{Hm\} + 1 \leq 2^{(M-n)} + nm + 1. \qquad (5.6)$$

Conclude that all expressions in Theorem I are computable within $2^{(M-n)}$ iterations of the VA.

Corollary I

$$\max_{L+1}\{\Delta cpm\} - ns_{max} \leq \max_L\{\Delta pm\} \leq \max_{L+1}\{\Delta cpm\} - s_{max} \qquad (5.7a)$$

$$\max\{\Delta cpm\} - ns_{max} \leq \max\{\Delta pm\} \leq \max\{\Delta cpm\} - s_{max} \qquad (5.7b)$$

$$\max_\infty\{\Delta cpm\} - ns_{max} \leq \max_\infty\{\Delta pm\} \leq \max_\infty\{\Delta cpm\} - s_{max} \qquad (5.7c)$$

Proof of Corollary I

To show the left hand side of (5.7a), use the identities of Theorem I, and take any length-$(L+1)$ candidate survivor path that has maximum Hamming weight. Omission of the last branch, yields a survivor path and reduces the Hamming weight by at most n. The resulting weight cannot exceed $\max_L\{Hm\}$. The right hand side of (5.7a) follows because any survivor path of maximum Hamming weight can be extended to a length-$(L+1)$ candidate survivor path. Each state by assumption has at least one outgoing branch labelled with a nonzero output symbol. Equations (5.7b, 5.7c) follow from (5.7a) by maximization over L, and let L go to ∞, resp. □

By an *excursion* we mean the encoder output that corresponds to a series of states that begins and ends with an 'all zeroes' state, and has no such states in between. Let $d_{free,L}$ denote the free distance of the binary convolutional code attained over at most L branches. For $L > m_-$, $d_{free,L}$ exists and is nonincreasing. If the output path of the VA is traced back from an arbitrary state instead of the state with minimum path metric [3], the quantities (5.8b, 5.8c) together with (5.3) prescribe the required numerical range for modulo arithmetic.

Corollary II

If $k = 1, R = 1/n$, *then*

$$\max_{1,L}\{\Delta cpm\} = d_{free,L}, \tag{5.8a}$$

$$\max_{1,\infty}\{\Delta cpm\} = d_{free}, \tag{5.8b}$$

$$\max_{1}\{\Delta cpm\} = d_{free,m_-+1} \tag{5.8c}$$

Proof of Corollary II

An excursion X of at most L branches of minimal Hamming weight, viz. $d_{free,L}$, ends in the final 'all zeroes' state. In general, $d_{free,L}$ can be (slightly) larger than d_{free} [1]. Because $k = 1$, there are only two candidate survivor paths for the final 'all zeroes' state, viz. the 'all zeroes' path and the path that has weight $d_{free,L}$. Therefore, the expression

for $\max_{1,L}\{\Delta cpm\}$ given in Theorem I and $d_{free,L}$ are equal. □

Example II

Consider the $R = 2/3$, memory order 1, binary convolutional code with generators [4],

$$g_1^{(1)} = (1\ 1),\ g_1^{(2)} = (0\ 1),\ g_1^{(3)} = (1\ 1),$$

$$g_2^{(1)} = (0\ 1),\ g_2^{(2)} = (1\ 0),\ g_2^{(3)} = (1\ 0).$$

Figure 3 depicts the evaluation of $\max_1\{\Delta cpm\} = 5$. However, $d_{free,m+1} = d_{free} = 3$. Example I already showes that $\max\{\Delta cpm\}$ can differ from $d_{free,m+1}$.

Theorem II-A 'Stopping rule'

A state S at depth L in the trellis with a path metric that exceeds the path metric of a state T by $s_{max}G(S-T)$ or more cannot be on the output path of the VA, where $G(S)$ denotes the minimal Hamming weight of any connection of state S to the 'all zeroes' path,

$$\begin{aligned} G(S) \;=\; & \min\{w_H(v_S) \mid v_S : \text{'}v_S \text{ starts in } S, \\ & \text{ends in an 'all zeroes'state'}\}. \end{aligned} \tag{5.9}$$

In particular state T can be chosen to have minimal path metric. Stopping rule A deletes all states S for which the path metric exceeds the minimum path metric of some state $T = MIN$ by $s_{max}G(S-MIN)$ or more. By stopping rule A^+ we mean that all pairs S,T are considered, not just $T = MIN$. In the proof of Theorem II-B we show that with stopping rule A the path metric differences at depth L in the trellis cannot exceed B_L, where

$$B_L = \max\{\min\{s_{max}G(S) - 1, s_{max}Hm_L(S)\}|S\}. \tag{5.10}$$

In order to simplify application of the stopping rule, all states S for which the path metric exceeds the minimum path metric by more than B_L are deleted. This is called stopping rule B.

With respect to the tightness of the bounds in Theorem III, it is of relevance whether for instance at an arbitrary depth into the trellis all path metrics can be equal. With hard decision decoders, in general, this is not the case. For the sake of definiteness, we introduce an erasure symbol that, when received, adds zero the discrimination between paths in the trellis.

Assumption IV

There is a channel symbol '*' that represents an erasure. That is, the symbol metric of '*' given that a '0' was sent equals the symbol metric given that a '1' was sent.

Theorem II-B '(Simplified) stopping rule'

A state S at depth L in the trellis for which the path metric exceeds the minimum path metric by more than B_L cannot be on the output path of the VA. For decoders with an erasure symbol (Assumption IV) B_L can not be defined any smaller than (5.10) up to a "round off" margin of $(s_{max} - 1)$.

Application of the stopping rule requires the computation of the minimum path metric and comparison of all path metrics with the minimum metric. Therefore, as stated before, the stopping rule is most logically combined with the rescaling method. Because $Hm_L(S), L \geq m$ is non-increasing in L, B_L is also non-increasing in L. For stream applications B_∞ is used, for block applications the larger value

$$B \stackrel{def}{=} \max\{B_L | m_- \leq L \leq m\} \tag{5.11}$$

applies.

Let $Hm(S)$ denote $Hm_L(S)$ at $L = L_\infty$. Define $G(S)$ is the Hamming metric of S at depth "∞" of the time-reversed convolutional code. Note that $G(S)$ can be computed with the VA[1]. Recall

$$\begin{aligned} Hm_L(S) \;=\; & \min\{w_H(u_S) | u_S : \text{'}u_S \text{ starts in} \\ & \text{an 'all zeroes' state, ends in } S, \text{ length } L\text{'}\}. \end{aligned} \tag{5.12}$$

[1]See also Proof 2 of Theorem II-B.

For a given S, the concatenation of the optimal u_S, v_S in (5.9,5.12) forms an excursion $X(S)$ through S of minimal Hamming weight ('lightest excursion'). Vice versa, given a lightest excursion X(S) through S the section $X_{pre}(S)$ from the start of X up to S has weight $Hm(S)$, and the section $X_{post}(S)$ from S to the end of X has weight $G(S)$. For the sake of uniqueness of definition, if excursions have equal weight, the order is defined in some arbitrary manner, e.g. lexicographically. B_∞ is redefined in terms of $X(.)$ as,

$$B_\infty = \max\{\min\{s_{max} w_H(X_{post}(S)) - 1, s_{max} w_H(X_{pre}(S))\}|S\}. \quad (5.13)$$

For the computation of B_L, the length of $X_{pre}(S)$ is constrained to L.

The set of excursions $\{X(S)|S\}$ is put in increasing order, $\mathcal{X} = \{X_i|i\}$. Each X_i has the property that it passes through some state for which X_i is the excursion of least Hamming weight to go through that state. By definition, X_1 has weight d_{free}, and for all states on X_1 we have that $X(S)$ equals X_1. Similarly, it can be verified that for all states on X_j that are not on any X_i $(i < j)$, we have $X(S) = X_j$. Denote by W the maximal Hamming weight of an excursion in $\mathcal{X}$,

$$W = \max\{w_H(X)|X \in \mathcal{X}\}.$$

Let W_L be similarly defined, with a constrained on the length of $X_{pre}(S)$ to at most L.

Lemma I

For binary convolutional codes for which any route of m_- branches has Hamming weight at least n, it holds that

$$\max_{s=0,\infty}\{cHm\} \leq W. \quad (5.14)$$

For any binary convolutional code, we have

$$(s_{max} d_{free,L} - 1)/2 \; - s_{max} n + 1 \leq B_L \leq (s_{max} W_L - 1)/2, \quad (5.15a)$$

$$(s_{max} d_{free} - 1)/2 \; - s_{max} n + 1 \leq B_\infty \leq (s_{max} W - 1)/2. \quad (5.15b)$$

Proof of Lemma I

The assumption implies that $G(S)$ of a state S that has an outgoing branch to an 'all zeroes' state equals the weight of that branch. Thus, a candidate path metric for the 'all zeroes' state corresponds to a lightest excursion and (5.14) follows. Equation (5.15b) follows from (5.15a) for $L = L_\infty$. The right hand side of (5.15a) is a consequence of (5.13) for general L instead of L_∞. To show the left hand side of (5.15a) observe that there is a state S on $X_1 = X(S)$, which has Hamming weight at least d_{free}, such that

$$\begin{aligned} s_{max} w_H(X_{pre}(S)) &\leq s_{max} w_H(X_{post}(S)) - 1, \\ s_{max} w_H(X_{pre}(S)) &\geq (s_{max} d_{free} - 1)/2 - s_{max} n + 1, \end{aligned}$$

□

Examples III and IV illustrate the computation of B and B_∞, and the use of the stopping rule. We remark beforehand, that for a certain value of B, path metric differences larger than B (but not larger than $B + n$) can occur before new looser states are deleted.

Example III

The code of Example I is reused (see Figure 2). A hard decision decoder is used, $s_{max} = 1$. From the trellis observe that, in terms of encoder output,

$$\begin{aligned} X(10) = X(01) = X_1 &= (11\ 00\ 10), \\ X(11) = X_2 &= (11\ 11\ 10\ 10). \end{aligned}$$

Note that

$$\begin{aligned} Hm(10) = Hm(01) &= 2, \quad G(10) = G(01) = 1, \\ Hm(11) &= 4, \quad G(11) = 2. \end{aligned}$$

The values of B and B_∞ work out to 1. Moreover, B cannot be defined any smaller (e.g., reception of 01 01 00 10 10). The numerical range with rescaling is reduced to $B + 1 = 2$ for the A-rule, and to $B + 2 = 3$ for the B-rule (see Example V). Compare this with $\max\{\Delta cpm\} = \max_\infty\{\Delta cpm\} = 5$. If the 'all zeroes sequence' is received, the stopping rule deletes all states except the 'all zeroes' states.

Example IV

The $m = 4$, $R = 1/2$ binary convolutional code with maximal $d_{free} = 7$ is taken from [4], $s_{max} = 1$ and the generator functions are defined as

$$g^{(1)} = (1\ 0\ 0\ 1\ 1),\ g^{(2)} = (1\ 1\ 1\ 0\ 1).$$

By inspection of the trellis, the following excursions are obtained (in terms of encoder output):

$$\begin{aligned} X_1 &= (11\ 01\ 01\ 10\ 11), \\ X_2 &= (11\ 10\ 00\ 00\ 00\ 10\ 10\ 11), \\ X_3 &= (11\ 01\ 10\ 00\ 00\ 10\ 00\ 01\ 11), \\ X_4 &= (11\ 01\ 10\ 00\ 00\ 01\ 01\ 00\ 01\ 11), \\ X_5 &= (11\ 01\ 01\ 01\ 10\ 01\ 10\ 11). \end{aligned}$$

The Hamming weight w_i of path X_i is

$$w_1 = 7,\ w_2 = 7,\ w_3 = 8,\ w_4 = 9,\ w_5 = 10.$$

Each X_i has the following states S for which $X(S) = X_i$ (decimal notation of the states),

$$X_1 :\ 1,2,4,8,\ X_2 :\ 5,6,11,12,\ X_3 :\ 3,7,10,13,14,\ X_4 :\ 15,\ X_5 :\ 9.$$

Table 1 lists $Hm(S)$ and $G(S)$. Observe that $B_\infty = 4$ and B_∞ equals the maximum of the $(G(S) - 1)$ values. Hence, B_L is constant and B equals B_∞. The maximum difference between candidate path metrics which occurs before looser states are deleted equals $B + 1 = 5$ for the A-rule, and $B + 2 = 6$ for the B-rule (see Example V). Without a stopping rule, the maximum difference between candidate path metrics equals $\max\{cHm\} = 8$ for block applications and $\max_\infty\{cHm\} = 7$ for stream applications (see Theorem I).

Denote by $\max_{A,L}\{\Delta cpm\}$ the maximum difference between candidate path metrics for states at depth L when stopping rule A is used. Similarly, $\max_{B,L}\{\Delta cpm\}$ for stopping rule B. A trivial bound is given by

$$\max_{A,L}\{\Delta cpm\} \leq \max_{B,L}\{\Delta cpm\} \leq B + n. \tag{5.16}$$

S*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hm(S)	5	4	5	3	3	3	5	2	5	4	3	3	4	4	5
G(S)	2	3	3	4	4	4	3	5	5	4	4	4	4	4	4
$W_H(X(S))$	7	7	8	7	7	7	8	7	10	8	7	7	8	8	9

* In decimal notation

Table 5.1: Numerical results of Example IV

In general, a somewhat tighter result than $B + n$ is possible. Let $I(S)$ be the maximum weight of any outgoing branch of the state S, i.e.,

$$I(S) = \max\{w_H(e(S,T))|\ '(S,T) \text{ a branch in the trellis'}\}.$$

Theorem III

For any binary convolutional code that satisfies Assumptions I-III, and S a state at depth $(L-1)$ in the trellis, it holds that

$$\max_{A,L}\{\Delta cpm\} \leq \max\{\min\{s_{max}G(S)-1, s_{max}Hm_{L-1}(S)\}+s_{max}I(S)|S\}, \tag{5.17a}$$

$$\max_{B,L}\{\Delta cpm\} \leq \max\{\min\{B_{L-1}, s_{max}Hm_{L-1}(S)\} + s_{max}I(S)|S\}. \tag{5.17b}$$

If a soft decision decoder has an 'erasure symbol' (Assumption IV), these bounds are tight within a "round off" margin of $(s_{max}-1)$.

Example V

For Examples III, IV with the B-rule, it can be verified by inspection that the values obtained from (5.16) are already tight, and Theorem III confirms these results. In case of Example III, note that

$$I(01) = w_H(01) = 1,\ I(10) = w_H(11) = 2,\ I(11) = w_H(01) = 1.$$

As a result,

$$\max_{A,L}\{\Delta cpm\} \leq 2.$$

With Example IV, only $I(9) = 1$ is of importance. Hence,

$$\max_{A,L}\{\Delta cpm\} \leq 5.$$

For the B-rule these values are tight as well.

The following Lemma II puts a limit on the reduction in numerical range that can be achieved by the stopping rules. It implies that for $R = 1/n$ binary convolutional codes, the penalty for modulo arithmetic without stopping rule compared to rescaling with stopping rule is at most one bit (two bits, if rescaling can be implemented with non-negative numbers).

Lemma II

The following lower bound applies to the numerical range of the VA with stopping rule.

$$\max_{B,L}\{\Delta cpm\} \geq \max_{A,L}\{\Delta cpm\} \geq (s_{max} d_{free} - 1)/2. \tag{5.18}$$

Proof of Lemma II

There is a branch (S, T) on $X_1 = X(S) = X(T)$ such that for the states S, T,

$$s_{max} w_H(X_{pre}(S)) \leq s_{max} w_H(X_{post}(S)) - 1,$$

$$s_{max} w_H(X_{pre}(T)) \geq (s_{max} d_{free} - 1)/2.$$

Assume the 'all zeroes' sequence is received. Observe that state S is not a looser state. The candidate path metric of the branch (S, T) equals

$$\begin{aligned} s_{max} Hm_{L-1}(S) + s_{max} w_H(e(S,T)) &= s_{max} w_H(X_{pre}(T)) \\ &\geq (s_{max} d_{free} - 1)/2 \quad \square \end{aligned}$$

As demonstrated by Examples I and II there is no simple general relation between d_{free} and the required numerical range. Theorem IV presents an upper bound on $\max_\infty\{\Delta pm\}$ in terms of the parameters

(n, k, m, M, d_{free}) of the binary convolutional code. This upper bound generalizes a recent result by Alston and Chau [1]. They proved that for decoders of binary $R = 1/n$ codes, under certain assumptions on the metric function [1], it holds that

$$\max_{\infty}\{\Delta pm\} \leq s_{max}\lfloor n(m+1) - d_{free}/2 \rfloor. \tag{5.19}$$

Define ΔM as the difference between the maximum possible number of memory cells in the encoder, km, and the true number of memory cells M. Note that $\Delta M = 0$ for $k = 1$.

Theorem IV 'Generalized Alston & Chau bound'

Under Assumptions I-III, for any (n, k, m, M, d_{free}) binary convolutional code, it holds that

$$\begin{aligned} &\max_{\infty}\{\Delta pm\} \leq \\ &\quad s_{max} \min\{\ \lfloor n(m+\delta) - d_{free}(1 - 2^{-(\Delta M + k\delta)}) \rfloor \mid \delta = 0, 1, \ldots\ \}. \end{aligned} \tag{5.20}$$

Example VI

For the code of Example I, the upper bound in Theorem IV evaluates to 4, which is a tight result. For the code of Example IV, we have that $\max_{\infty}\{\Delta pm\}$ equals 5. However, Theorem IV gives an upper bound of 6. Hence, Theorem IV is not tight in general.

The question arises: *Can $\delta > 1$ strengthen the bound for a particular (n, k, m, M, d_{free})?* For the sake of simplicity set $k = 1$ and let

$$f(\delta) = n(m + \delta) - d_{free}(1 - 2^{-\delta}).$$

The δ that minimizes f equals

$$\delta^* = \lfloor \log_2(d_{free}/n) \rfloor. \tag{5.21}$$

For the sake of simplicity, assume (d_{free}/n) is a power of two. Then, we have that

$$\begin{aligned} f(\delta^*) &= n(m + 1 + \log_2(d_{free}/n)) - d_{free} \\ &= nm - d_{free} + n\ o(d_{free}). \end{aligned} \tag{5.22}$$

The A & C bound equals $nm - d_{free}/2 + n$. From the table of $n = 2$, optimal d_{free} codes [4], we see that for $2 \leq m \leq 16$, the free distance lies in the range $m + 3 \leq d_{free} \leq m + 4$. For $m = 16$, the table reads $d_{free} = 20$ and the (A & C) bound equals $f(1) = 24$. The new bound, with $\delta^* = 3$ yields $f(3) = 20$.

III - THE NUMERICAL RANGE REQUIRED FOR THE VA

In this section, Theorem I is proved. This theorem provides exact expressions for $\max_L\{\Delta pm\}$, $\max_{1,L}\{\Delta cpm\}$ and $\max_L\{\Delta cpm\}$. These expressions are easy to evaluate. We already pointed out that if the all zeroes sequence is received, the path metrics equal s_{max} times the Hamming metrics by Assumption II. Therefore, we only need to show that no larger (candidate) path metric differences than stated in Theorem I can occur.

Proof of Theorem I

First, we show that no larger values for $\max_L\{\Delta cpm\}$ than specified can occur. Hereafter, the other two maxima follow from small modifications of the arguments below.

See Figure 4. At a given depth L in the trellis, let *MIN* be the state that has minimal candidate path metric. The candidate survivor path Q of *MIN* consists of the survivor path of some state *min*, followed by the branch (min, MIN). By $e(Q)$ we refer to the encoder output symbols that correspond to the series of states Q, and similarly for other paths. In the context of metrics, Hamming weights, etc., the term 'path' always refers to a path in terms of encoder output symbols, i.e., $e(Q)$ instead of Q. Let *MAX* be the state that has maximum candidate path metric at depth L. The candidate survivor path for *MAX* consists of the survivor path of some state *max*, followed by the branch (max, MAX). (*MAX* and *MIN* may coincide.)

Note, that the survivor path of *max* has the minimal metric of all paths that end in *max*. The difference of the given candidate path metrics of *MAX* and *MIN* is upper bounded by the given metric function d

of an arbitrary path P that leads to max and MAX, minus the metric of Q. The choice of P is detailed below. As discussed in [2], a term in the latter difference is of the form

$$d(r, e(s_1, t_1)) - d(r, e(s_2, t_2)), \tag{5.23}$$

where

r	=	received channel symbols (an n-vector),
s_1, t_1	=	successive states on P,
s_2, t_2	=	successive states on Q,
$e(s,t)$	=	encoder output n-vector that corresponds to the state transition (s,t),
$d(x,y)$	=	metric function of the (series of) channel symbol(s) x and the hypothesis bit(s) y.

The difference of the branch metrics of the paths P and Q at a certain depth in the trellis can be upper bounded as follows.

$$\begin{aligned} d(r, e(s_1, t_1)) - d(r, e(s_2, t_2)) &\leq s_{max} d_H(e(s_1, t_1), e(s_2, t_2)) \\ &= s_{max} w_H(e(s_1, t_1) - e(s_2, t_2)) \end{aligned} \tag{5.24}$$

Summation over the length of the paths of (5.24) yields the following upper bound on the maximum path metric difference,

$$\max\{\Delta pm\} \leq s_{max} w_H(e(P) - e(Q)). \tag{5.25}$$

By the *linearity* of the encoder function $e(.)$, we have

$$e(P) - e(Q) = e(P - Q) = e(0, \ldots, (max - min), (MAX - MIN)) \tag{5.26}$$

Hence, (5.25) is equivalent to

$$\max\{\Delta pm\} \leq s_{max} w_H(e(P - Q)). \tag{5.27}$$

That is, the maximum candidate path metric difference of any two states is upper bounded by s_{max} times the Hamming weight of a path L branches long.

Next, consider the question whether this path can be assumed to consist of a survivor path followed by a connecting branch. Observe

that the path P can be chosen freely out of all possible paths ending in max, followed by the branch (max, MAX). There is a affine correspondence between a path P and the resulting difference path $U = (P - Q)$, that ends with the branch $(max - min, MAX - MIN)$. Thus, there is a choice of P makes U equal to the survivor path of $(max - min)$ followed by $(max - min, MAX - MIN)$

Consider $\max\{cHm\}$. Survivor paths of length $L < m_- + 1$ can always be prefixed with 'all zeroes' branches to reach length $m_- + 1$. The Hamming metric stays the same and one again obtains survivor paths because no selection has been made yet. Thus, maximization for $L = m_- + 1$ upper bounds the result for $L < m_- + 1$. Now assume $L > m + 1$. Consider a candidate survivor path Z that ends with some branch (opt, OPT). Let pre be the 'all zeroes' state at depth $(L - L'), L' \geq m + 1$. Extend the 'all zeroes' path that leads to pre to a path that ends with (opt, OPT). By the VA, the resulting path W has a larger Hamming metric than the original candidate survivor path Z. W can be stripped of its leading $(L - L')$ 'all zeroes' branches, so that it assumes length L'. As a result,

$$\max_L\{cHm\} \leq \max_{L'}\{cHm\}, \quad L \geq L' \geq m + 1. \tag{5.28}$$

This proves Theorem I, for $\max\{\Delta cpm\}$.

Next, consider $\max_1\{\Delta cpm\}$. The distinction between the proof of the previous situation is that differences of candidate path metrics for the same state are considered. Thus, MAX and MIN coincide, and $(MAX - MIN)$ equals the 'all zeroes' state. With this restriction, the rest of the proof still holds true. In case of $\max\{\Delta pm\}$, instead of survivor paths followed by a connecting branch, only survivor paths are considered. In the latter proof, equality (5.5) emerges as a combination of equivalents of (5.25) and (5.26) for survivor paths P and Q. Again, the proof is similar and is left to the reader. □

IV - SELECTION OF STATES THAT CAN BE ON THE OUTPUT PATH

Figure 5 illustrates the key idea behind the stopping rule. With reference to Corollary II, consider the worst case situation with $k = 1$ for $\max_{1,\infty}\{\Delta cpm\} = d_{free}$. That is, a VA with Hamming metric, comparison of the 'all zeroes' candidate survivor path with an excursion X of weight d_{free}, when the 'all zeroes' sequence is received. Take a state S at some depth t in the trellis that is about half-way on X. Assume that the Hamming weight of the section $[0, \ldots, S]$ of X, i.e., the Hamming metric $Hm(S)$ of S slightly exceeds $d_{free}/2$. *Can the output path of the VA run via state S?* Even if the data received after stage t would exactly match the section $[S, \ldots, 0]$ of X, the metric of the path X still exceeds $d_{free}/2$, whereas the 'all zeroes' path ends with a metric less then $d_{free}/2$. 'Looser state' S cannot be on the output path of the VA. Deletion of looser states could half $\max\{\Delta cpm\}$.

We give two proofs of Theorem II-A. The first proof is more constructive and detailed in nature. The second proof, which invokes Theorem I, is more conceptual.

Proof 1 of Theorem II-A

The formal principle behind the stopping rule is illustrated by Figure 6. Consider a state S and T at depth L into the trellis. Assume that the path metric $pm(S)$ of S exceeds the path metric of T by $s_{max}G(S - T)$ or more. Let p_S be the survivor path of S. Take an extention q_S of p_S that starts from state S to what could in principle be the output path $P = (p_S|q_S)$ of the VA. The metric of P is

$$m_P = pm(S) + d(r, q_S), \tag{5.29}$$

where r is the received sequence that starts from depth t, and d is the metric function as defined in the proof of Theorem I. Out of all paths that emanate from T at depth t and merge with q_S at some later stage of the trellis, let q_{min} be the path for which $d_H(q_{min}, q_S)$ is minimal. The metric of the concatenation Q of the survivor path of T with q_{min} is

$$m_Q = pm(T) + d(r, q_{min}). \tag{5.30}$$

Observe that for m_P to be smaller than m_Q, q_S is the best possible value of r. Let v be the difference path $v = q_S - q_{min}$ that connects the state $(S - T)$ to the 'all zeroes' path. By the triangle inequality,

$$\begin{aligned} m_P - m_Q &= pm(S) - pm(T) + d(r, q_S) - d(r, q_{min}) \\ &\geq pm(S) - pm(T) - s_{max} d_H(q_S, q_{min}) \\ &= pm(S) - pm(T) - s_{max} w_H(v) \\ &= pm(S) - pm(T) - s_{max} G(S - T) \\ &\geq 0. \end{aligned} \tag{5.31}$$

If the path metric difference equals $s_{max}G(S - T)$ then the output path of the VA may be non-unique, and the path which goes via T is preferred. This gives a further reduction in numerical range.

For block applications, special considerations apply to the end of the trellis. A finite length of the trellis can limit the length of the extension q_S, q_{min}. On the other hand, in that case it suffices for q_S, q_{min} to reach the final stage of the trellis. They do not have to merge. Accordingly, define $G_M(S)$ to be

$$G_M(S) = \min\{w_H(v_S) | v_S : \text{'}v_S \text{ starts in } S, \text{ has length } M\text{'} \}, \tag{5.32}$$

so that we have

$$\begin{aligned} G_M(S) &\leq \min\{w_H(v_S) | \text{'}v_S \text{ begins in } S, \text{ has length } L = \infty\text{'}\} \\ &= \min\{w_H(v) | \text{'}v \text{ begins in } S, \, v \text{ merges with 'zeroes' path}\} \\ &= G(S) \end{aligned} \tag{5.33}$$

Thus, at the end of the trellis $s_{max}G(S)$ is an overestimate of the gain that can be realized. Hence, looser states for a certain possible gain $s_{max}G$, certainly are looser states when the true possible gain is only $s_{max}G_M$. □

Proof 2 of Theorem II-A

If at a certain depth L in the trellis, the path metric of a state S exceeds that of a state T, then this difference must be compensated for by a difference in metrics ("gain") of the *extensions* of the survivor

paths of S and T to the end of the trellis. Otherwise, the output path of the VA cannot run via S. By (5.5) applied to the time reversed convolutional code the difference between the metrics of the extensions is bounded by $s_{max}Hm_{rev}(S-T)$, where Hm_{rev} denotes the Hamming metric function of the time reversed convolutional code.

In fact, (5.5) needs to be modified to the situation in which all encoder states are allowed as initial states of the time reversed code because all encoder states are possible at the end of the trellis. As a consequence, not just 'all zeroes' but all initial states are allowed in the evaluation of $Hm_{rev}(S-T)$. This change in initial condition makes that $Hm_{rev,L}(S)$ is non-decreasing in L. At the end of the trellis, i.e., at the beginning of the time reversed trellis, $Hm_{rev}(S-T)$ is an overestimate of the possible gain $Hm_{rev}(S-T)$. The gain function $G(S)$ is an overestimate of $Hm_{rev}(S)$, because the modification of initial condition can only decrease the Hamming metrics. In case the difference of path metrics of S and T equals $s_{max}G(S-T)$, the path that goes via T is preferred. Finally, conclude that if S is to be on the output path of the VA, then the path metric of S should be less than $s_{max}G(S-T)$. □

Examples III, IV show that the stopping rule can reduce the numerical range of the rescaling option. Analysis of the effect of the stopping rule on the numerical range is the subject of Theorem III.

Proof of Theorem II-B

Consider the maximum difference between path metrics when the stopping rule of Theorem II-A has been applied. We show that this numerical range is at most B_L. Therefore, deletion of all states that have a path metric that exceeds the minimum path metric by more than B_L is a valid alternative stopping rule.

By Theorem I, for arbitrary states S and T at depth L, we have

$$pm(S) - pm(T) \leq s_{max}Hm_L(S-T). \tag{5.34}$$

The stopping rule of Theorem II-A deletes S unless

$$pm(S) - pm(T) \leq s_{max}G(S-T) - 1. \tag{5.35}$$

Combination of (5.34) and (5.35) yields

$$pm(S) - pm(T) \leq \min\{s_{max}G(S-T) - 1, s_{max}Hm_L(S-T)\}. \quad (5.36)$$

Maximization over $A = (S - T)$ proofs that no larger difference than B_L is possible.

Let S^* be the state that achieves the maximum in (5.10). If $s_{max}Hm(S^*)$ is less than or equal to $(s_{max}G(S^*) - 1)$, then we suppose that the 'all zeroes' sequence is received noisefree and a path metric difference of B_L occurs between S^* and the 'all zeroes' state. If $s_{max}Hm(S^*)$ exceeds $(s_{max}G(S^*) - 1)$, assume that 'all zeroes' are received noisefree until the survivor path of S^* has accumulated a Hamming metric of $s_{max}(G(S^*) - 1)$. Assume that the remaining 1's in the survivor path of S are erased. Each erasure reduces by s_{max} the path metric difference between S^* and the 'all zeroes' state, which originally was $s_{max}Hm(S^*)$ and now becomes $s_{max}(G(S^*) - 1)$. Hence, for soft decision decoders with an erasure symbol the value of B_L is within $(s_{max} - 1)$ of optimality. □

V - REQUIRED NUMERICAL RANGE FOR RESCALING W. STOPPING RULE

Proof of Theorem III

Let (s, S) and (t, T) be branches, with S and T states at depth L in the trellis (s, t at depth $(L - 1)$). Because state s was not deleted by the stopping rule, it holds that

$$\begin{array}{lcll} pm(s) - pm(t) & \leq & s_{max}G(s-t) - 1, & \text{for stopping rule } A, \\ pm(s) - pm(t) & \leq & B_{L-1}, & \text{for stopping rule } B. \end{array} \quad (5.37)$$

Similarly as in the proof of Theorem I, with $u = s - t$, $U = S - T$, it follows that

$$\begin{array}{rcl} pm(S) - pm(T) & \leq & pm(s) - pm(t) + d(r, e(s, S)) - d(r, e(t, T)) \\ & \leq & pm(s) - pm(t) + s_{max}d_H(e(s, S), e(t, T)) \\ & \leq & pm(s) - pm(t) + s_{max}w_H(e(u, U)) \end{array} \quad (5.38)$$

Combination with (5.5), (5.37) and maximization over (u, U) yields the right hand side of (5.17a,b) as an upper bound.

To show that the values specified by (5.17) can actually occur in practise for decoders that have an erasure symbol, assume that the received symbols that correspond to the branch (s, S) are received noise-free (Assumption II). Combination with the result of Theorem II-B shows that within a "round off" margin of $(s_{max} - 1)$ the bounds of (5.17) are tight under Assumption IV. □

VI - BOUND ON $\max_\infty\{\Delta pm\}$ IN TERMS OF d_{free}

The generalized A & C bound (5.21) reduces to the elementary bound

$$\max_\infty\{\Delta pm\} \leq s_{max} mn,$$

for $\delta = 0$ and $\Delta M = 0$ (k=1). The A & C bound is obtained for $\delta = 1$ for rate $R = 1/n$ codes. First, the A & C case is proved. Finally, a packing lemma (Lemma III) establises Theorem IV for $k\delta + \Delta M > 1$. Note that Theorem IV easily generalizes to $d_{free,m+\delta}$ [1] instead of d_{free}, i.e., the free distance attained over $L = m + \delta$ branches.

Proof of Theorem IV for $\delta = 1$, $k = 1$, 'A & C bound'

By Theorem I it suffices to find an upper bound for the maximum Hamming metric in a trellis L branches deep. Denote the state with maximum Hamming metric 'OPT'.

For $L = m + 1$, there are at least two different paths that start in the 'all zeroes' state and end in OPT. By definition, the Hamming distance between any two different paths that start in the same state and end in the same state is at least d_{free}. Thus, the two paths have distance at least d_{free}. Call these paths P_1 and P_2. Define Q_1 and Q_2 as the bitwise complement of P_1 and P_2, respectively. Of course, Q_1 and Q_2 also have minimum distance d_{free}:

$$w_H(P_1) = (m + 1)n - w_H(Q_1)$$

$$w_H(P_2) = (m+1)n - w_H(Q_2).$$

By the *triangle inequality* for the Hamming distance function d_H, we have that

$$w_H(Q_1) + w_H(Q_2) = d_H(Q_1, 0) + d_H(Q_2, 0) \geq d_H(Q_1, Q_2) \geq d_{free}. \tag{5.39}$$

By Lemma I, it holds that

$$\begin{aligned} \max_{\infty}\{\Delta pm\} &\leq s_{max} \min\{w_H(P_1), w_H(P_2)\} \\ &= s_{max} \lfloor (m+1)n - \max\{w_H(Q_1), w_H(Q_2)\} \rfloor. \end{aligned} \tag{5.40}$$

Since a maximum is never less than the average, from (5.39) it follows that

$$\max\{w_H(Q_1), w_H(Q_2)\} \geq d_{free}/2. \tag{5.41}$$

Substitution of (5.41) in (5.40) yields the Alston and Chau (A & C) result, i.e.,

$$\max_{\infty}\{\Delta pm\} \leq s_{max} \lfloor (m+1)n - d_{free}/2 \rfloor. \qquad \square \tag{5.42}$$

In our proof of the A & C bound the substitution of (5.41) in (5.40) is the crucial step. Given any set of two binary vectors $\{Q_1, Q_2\}$ with minimum distance d_{free}, at least one of the vectors has Hamming weight at least $d_{free}/2$. Essentially, this amounts to a *packing problem.* It is impossible to pack 2 points with minimum distance d_{free} in a sphere of radius less than $d_{free}/2$. For the i-th shift register, m_i bits are determined by the final state OPT, and $\delta + (m - m_i)$ bits can be chosen freely. In total there are

$$k\delta + km - M = k\delta + \Delta M$$

degrees of freedom. If $2^{\delta k + \Delta M} > 2$ points are to be packed, the radius of the containing sphere has to be larger. This increases the "$d_{free}/2$" term of the A & C bound. However, this increase can be compensated for by an commensurable increase in the "$(m+1)n$" term to $(m+\delta)n$. As we shown in Example VI, for codes with a large enough (d_{free}/n)-ratio a significant net improvement over the A & C bound is possible.

The set[2] of $2^{\delta k+\Delta M}$ paths which connect the initial 'all zeroes' state and 'OPT' forms a affine space. As the encoding function is linear, both the state sequences and the encoder output of the paths form a linear space. Because a convolutional code is uniquely decodable, the paths have the same starting state and the same final state, the encoder output sequences of the paths are all different. Any pair of paths has distance at least d_{free} (or $d_{free,m+\delta}$, etc.). Finally, the (n, k, m, M, d_{free}) convolutional code supplies an affine subspace of $2^{\delta k+\Delta M}$ binary vectors $\{Q_i|i\}$ with minimal Hamming distance d_{free}. Our generalization of the A & C bound concentrates on the solution of the following problem.

Packing Problem

Given an affine subspace of 2^H vectors $\{Q_1, Q_2, \ldots, Q_{2^H}\}$, $Q_i \in GF(2^N)$ with minimum Hamming distance D: What is the smallest possible radius $R^*(H, D, N)$ of a sphere with center zero that contains all vectors?

With the following choice of the parameters H, D, M: $H = k\delta + \Delta M$, $D = d_{free}$, $N = (m + \delta)n$, our proof of the A & C bound generalizes and shows that

$$\max_{\infty}\{\Delta pm\} \leq \tag{5.43}$$
$$s_{max} \min\{\lfloor (m+\delta)n - R^*(k\delta + \Delta M, d_{free}, (m+\delta)n)\rfloor \mid \delta \geq 0\}.$$

Lemma III provides an analytical lower bound for R^*. Substitution of this bound in (5.44) proves Theorem IV.

Lemma III 'Packing Lemma'

The maximum Hamming weight R^* of an H-dimensional affine subspace of $GF(2^N)$ with minimum Hamming distance D is at least $(1 - 2^{-H})D$.

[2]For codes with $k > 1$ and $\Delta M > 0$, the value $\delta = 0$ is allowed.

Proof of Lemma III

Lemma III is first proved for the case $H = 2$. Consider four binary vectors $\{Q_1, Q_2, Q_3, Q_4\}$ that form an affine subspace. In a field of characteristic two, if four points form a linear subspace, the sum of the vectors equals the 'all zeroes' vector. This property is invariant under a translation of the points over a constant vector. Thus, the same property holds for an affine subspace. Denote by $(abcd)$, $a, b, c, d \in \{0, 1\}$ the entries of the vectors at an index $i = 1, 2, \ldots, N$. The affine subspace property is in fact a bitwise property. The number of ones, i.e., $(a + b + c + d)$ is even. N_{abcd} denotes the number of times the combination $(abcd)$ occurs.

The packing problem can be restated in terms of the N_{abcd}-variables, as an integer linear programming problem (ILP) (e.g., $\sum N_{10**} = N_{1000} + N_{1001} + N_{1010} + N_{1011}$).

$$N = \sum N_{****}, \tag{5.44a}$$

$$\begin{aligned} D \leq d_{12} &= \sum N_{10**} + \sum N_{01**}, \\ \cdots &= \cdots \\ D \leq d_{34} &= \sum N_{**10} + \sum N_{**01}, \end{aligned} \tag{5.44b}$$

$$\begin{aligned} w_1 &= \sum N_{1***}, \\ \cdots &= \cdots \\ w_4 &= \sum N_{***1} \end{aligned} \tag{5.44c}$$

$$\begin{aligned} \max\_w &\geq w_1 \\ \cdots &\geq \cdots \\ \max\_w &\geq w_4 \end{aligned} \tag{5.44d}$$

$$R^* = \min\{ \max\_w \mid N_{abcd}, \text{ (5.45a-d) } \}. \tag{5.44e}$$

The conditions (5.45a-d) are necessary conditions.

The ILP (5.45) can be relaxed to a real linear programming problem (LP) when only the average weight of the vectors and the average pairwise distance are constrained.

$$N = \sum N_{****}, \tag{5.45a}$$

$$D \leq \sum avg\_d_{****} N_{****}, \tag{5.45b}$$

$$w = \sum avg\_w_{****} N_{****}, \tag{5.45c}$$

$$R = \min\{\ w \mid N_{abcd},\ (5.46\text{a-c})\ \}, \tag{5.45d}$$

where $avg\_w_{abcd} = (1/4)w_H(abcd)$, and avg_d_{abcd}, denotes the average of the six pairwise Hamming distances between the four bits. For instance, $avg\_d_{0011} = 4/6 = 2/3$. Observe that avg_w_{abcd} as well as avg_d_{abcd} depend only on the Hamming weight of $(abcd)$.

Let

$$T_o = N_{0000}, \tag{5.46a}$$

$$T_2 = N_{1100} + N_{1010} + N_{1001} + N_{0110} + N_{0101} + N_{0011}, \tag{5.46b}$$

$$T_4 = N_{1111}. \tag{5.46c}$$

Substitution of (5.47) into (5.46) yields

$$N = T_o + T_2 + T_4, \tag{5.47a}$$

$$D \leq 2T_2/3, \tag{5.47b}$$

$$w = T_2/2 + T_4. \tag{5.47c}$$

The solution of the LP is $R = 3D/4$. Indeed, this solution is independent of the dimension N of the space. However, as a consequence of equation (5.48a) N must exceed $3D/2$, or the LP (5.48) has no solution. Observe that the LP (5.48) is a relaxation of the original formulation of the packing problem (5.45), so if (5.48) has no solution, in fact, the packing problem has no solution. Hence, the proof of our result produces as a Corollary a lower bound for the dimension N of the space for the packing problem to have a solution at all.

The main facts used in the proof for $H = 2$ are

- avg_w_{abcd} and avg_d_{abcd} depend only on the number of ones in the section $(abcd)$ of the affine subspace,
- the number of ones in a section of an affine subspace of dimension H equals 2^{H-1}, with the exception of the trivial 'all zeroes' and 'all ones' sections.

These basic facts hold true regardless of the value of H. Let $P = 2^{H-1}$. For general H, the analogue of (5.48) works out to

$$D \leq \frac{PT_P}{2P-1}, \tag{5.48a}$$

$$w = T_P/2 + T_{2P}, \tag{5.48b}$$

$$R^* = \min\{\ w \mid T_P,\ T_{2P},\ (5.49a, b)\ \}. \tag{5.48c}$$

Here T_P denotes the sum of all $N_{x_1 \ldots x_{2P}}$ for which $(x_1 \ldots x_{2P})$ forms a section of an affine subspace and the Hamming weight of $(x_1 \ldots x_{2P})$ equals P. The LP (5.49) is solved for $R^* = (1 - 2^{-H})D$, which proves Lemma III in full generality. Similarly to the case $H = 2$, a necessary lower bound for the dimension N of the space for the packing problem to have a non-empty solution at all follows, viz. $N \geq (2 - 1/P)D$. □

Proof of Theorem IV

For $\delta = 0$ and $\Delta M = 0$, the bound (5.44) reduces to the elementary upper bound $s_{max}mn$. Therefore, without loss of generality, it is assumed that $\Delta M > 0$ for $\delta = 0$. Thus, in any case the number of points in the affine subspace is nonzero because $k\delta + \Delta M > 0$. Theorem IV follows by application of Lemma III to (5.44). □

History of Results and Acknowledgement

While reading [1], we discovered a more general approach as stated in Theorem I and Theorem IV. Subsequently, we learned of the existence of a NASA internal report in which results very similar to our Theorem I had apparently been found [8][3] Theorems I, IV were presented at the 1992 Symposium on Information Theory in the Benelux [5], and

[3]During the revision of the manuscript (Sept. 1993) a copy of [8] is not yet available to us.

at the 1993 IEEE International Symposium on Information Theory. Theorem I shows that the maximum differences between (candidate) path metrics occurs in a situation that is trivial for decoder operation, viz. noiseless reception of codewords. This led to the discovery of the stopping rule (Theorem II, III). In an extension of the original paper for the 1993 Benelux Symposium ([6]) the stopping rule and its analysis were added.

The author would like to thank his colleagues Johan van Tilburg and Frank Muller for their careful reviews of this manuscript.

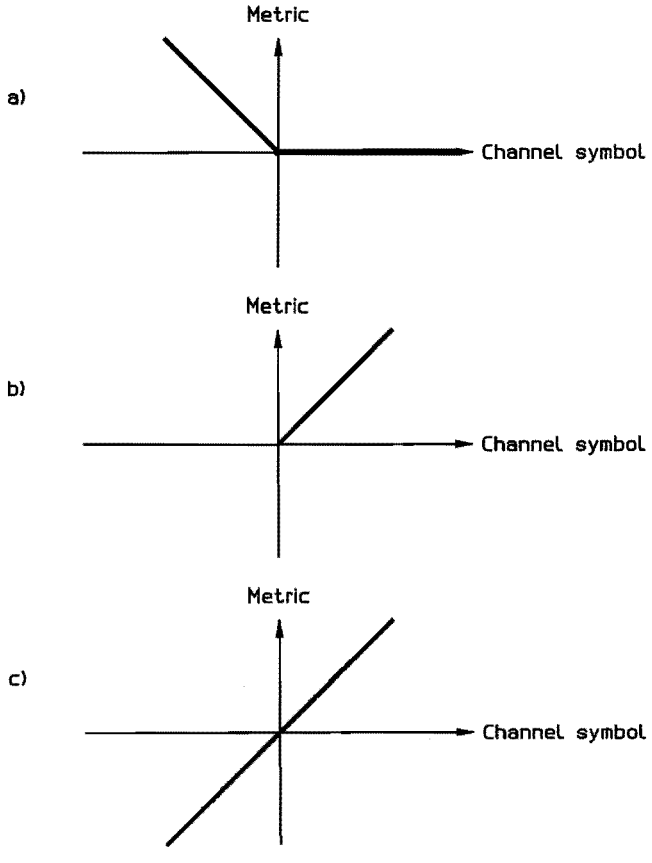


Figure 5.1: Symbol metrics
a - given that a '0' was sent
b - given that a '1' was sent
c - difference between b) and a)

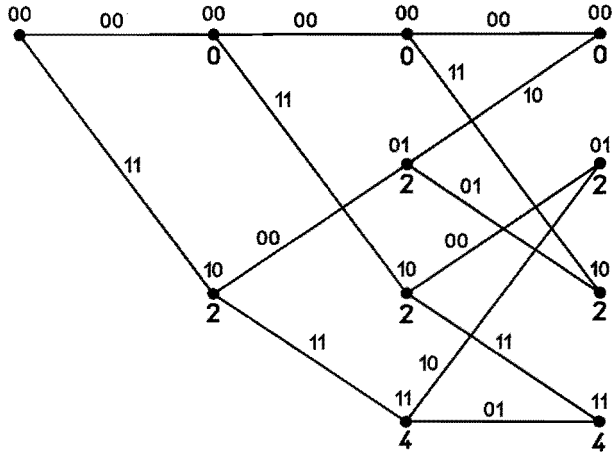


Figure 5.2: Trellis of R=1/2 code. 3 stages, 4 states

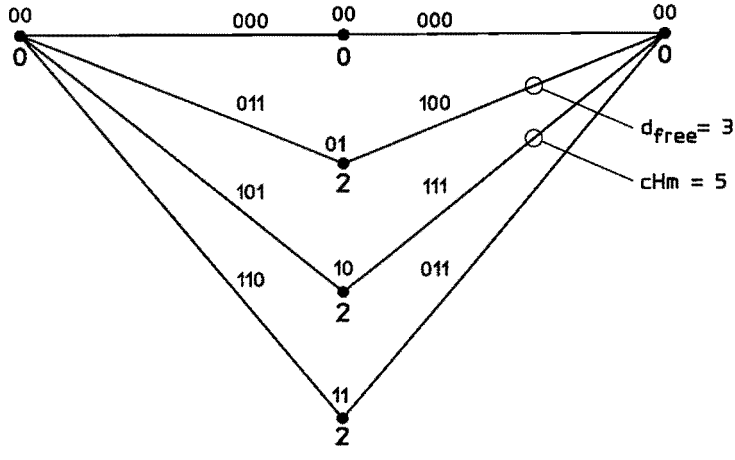


Figure 5.3: Part of trellis of R=2/3 code. 2 stages, 4 states

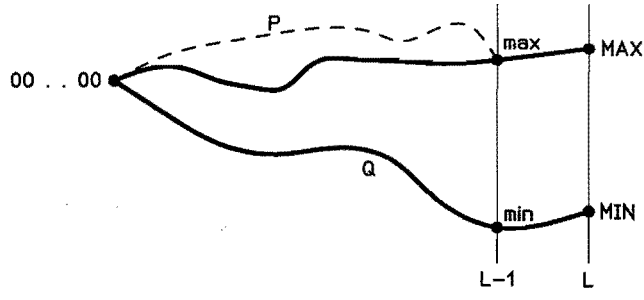


Figure 5.4: Converse for $\max\{\Delta cpm\}$

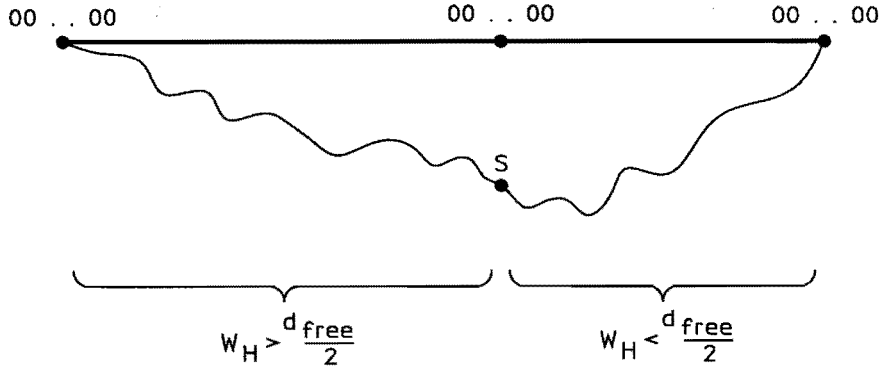


Figure 5.5: Key idea behind stopping rule

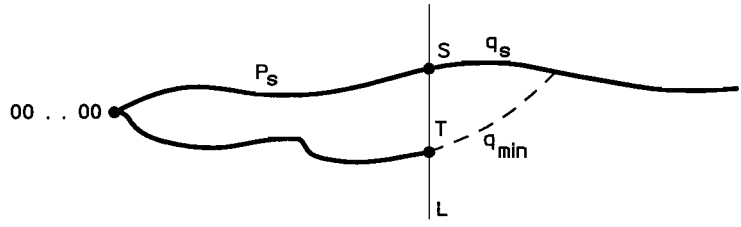


Figure 5.6: Competition between paths

Bibliography

[1] M.D. Alston, P.M. Chau, '*An improved analytical bound on the maximum difference of path metrics in Viterbi decoders of binary tree convolutional codes*', submitted to IEEE Trans. on Commun, Vol. COM-40, 1992.

[2] A.P. Hekstra, '*An alternative to metric rescaling in Viterbi decoders,*' IEEE Trans. Commun., Vol. COM-37, pp. 1220-1222, Nov. 1989.

[3] C.M. Rader, '*Memory management in a Viterbi decoder,*' IEEE Trans. Commun., Vol. COM-29, pp. 1399-1401, Oct. 1981.

[4] S. Lin, D.J. Costello, '*Error control coding: fundamentals and applications*', New Jersey: Prentice Hall, 1983.

[5] A.P. Hekstra, '*On the Maximum Difference between Path Metrics in a Viterbi Decoder*', 13th Symposium on Information Theory in the Benelux, Enschede, the Netherlands, June 1-2, 1992.

[6] A.P. Hekstra, '*Reduction of the Numerical Range of the of the Path Metrics in a Viterbi Decoder*', 14th Symposium on Information Theory in the Benelux, Veldhoven, the Netherlands, May 17-18, 1993.

[7] A.P. Hekstra, '*On the Maximum Difference between Path Metrics in a Viterbi Decoder*', 1993 IEEE Internat. Symp. on Information Theory, San Antonio, Jan. 17-22, 1993.

[8] D. Gilhousen, I. Jacobs, A. Viterbi, "Coding systems study for high data rate telemetry links," NASA Report CR-114278, Jan. 1971.

[9] P.H. Siegel, C.B. Shung, T.D. Howell, H.K. Thaper, '*Exact bounds for Viterbi detector path metric differences*', Proc. of IEEE Int. Conf. Acoust., Speech and Signal Proc., pp. 1093-1096, May 13-16, 1991.

Excerpt (Abstract in Dutch)

Anno 1994, circa een halve eeuw na Shannon's stichting van het vakgebied, is informatietheorie een nogsteeds zeer levendig, en uiterst veelzijdig vakgebied. Deze dissertatie presenteert bijdragen op een aantal deelgebieden der informatietheorie,

- een bovengrens voor het capaciteitsgebied van het tweeweg-kanaal met enkelvoudige uitgang,
- een analyse van de maximale processordichtheid in een programmeerbare chip, waarbij n paren processoren via n disjuncte paden moeten kunnen communiceren,
- definitie van een kanaal met onzekere tijdrelatie tussen zender en ontvanger, en een bepaling van het capaciteitsgebied van dit kanaal,
- een implementatiemethode voor de padmetrics in het Viterbi algoritme gebruik makende van two's complement arithmetic,
- een analyse van het numeriek bereik van de padmetrics in het Viterbi algoritme, alsmede een techniek voor de reductie van dit bereik met maximaal een factor twee.

Hoofdstuk 1 behandelt een bovengrens voor het capaciteitsgebied van het tweewegkanaal met enkelvoudige uitgang. op basis van een zogenaamde afhankelijksheidsbalans. Wanneer een tweeweg-kanaal uit twee onafhankelijke 'eenweg-kanalen in beide richtingen bestaat dan zijn in het optimale geval de ingangssignalen aan beide zijden van het gecombineerde kanaal statistisch onafhankelijk. In het algemeen, kan

het gebruik van statistische afhankelijkheid tussen de beide ingangssignalen de bereikbare datasnelheden vergroten. De ingangssignalen zijn echter afgeleid van de berichten, en de berichten zijn statistisch onafhankelijk gekozen. De eerste ingangssignalen van een strategie zullen derhalve altijd onafhankelijk zijn. Voor volgende ingangssignalen hoeft dat niet te gelden, maar wat wel bewezen kan worden is dat gemiddeld, de afhankelijkheid die de ingangssignalen bezitten niet groter kan zijn dan de a posteriori afhankelijkheid van de ingangssignalen gegeven de uitgangssignalen. Waar het op neer komt is dat afhankelijkheid die gebruikt wordt eerst opgebouwd moet worden. Bovendien kan het introduceren van een extra parallelkanaal de capaciteit van het oorspronkelijke kanaal alleen maar vergroten. Een dergelijk parallelkanaal kan zo gekozen worden dat het a posteriori afhankelijkheid afbreekt, zonder dat het wezenlijke informatie weggeeft en derhalve het capaciteitsgebied daadwerkelijk vergroot.

Hoofdstuk 2 verhandelt over wat met wat gevoel voor analogie parkeerplaatstheorie genoemd kan worden. Gegeven een rechthoek die is onderverdeeld in rechthoekige cellen, allemaal van gelijke grootte, kunnen cellen gebruikt worden als processorcellen of als communicatiecellen. Het streven is een maximale dichtheid van processorcellen te bereiken, gegeven dat elk paar processorcellen verbonden kan worden via een pad dat uit louter communicatiecellen bestaat. De maximale dichtheid van processorcellen blijkt 2/3 te zijn. Wanneer we de processorcellen als autos beschouwen, en de communicatiecellen als lege cellen, en een processor cell vervangen door een 'uitgangs cel', kan iedere auto naar de uitgang. Vice versa, op een parkeerplaats, wanneer alle autos naar een uitgangscel kunnen bewegen over een pad van vrije cellen, indien de autos bidirectional kunnen bewegen, kan dan iedere auto via de uitgang naar iedere andere auto bewegen. In die zin is dan ieder paar autos met elkaar te verbinden. Meer algemeen is de vraag, wat, gegeven n paren van processoren die verbonden moeten worden via disjuncte paden, de maximale dichtheid van processorcellen kan zijn? Het blijkt dat het optimale verloop van de processordichtheid $O(n^{-2})$ is.

Hoofdstuk 3 gaat over kanalen met een onzekere tijdrelatie. Shannon's resultaat voor de capaciteit van het 'eenweg-kanaal vooronderstelt dat zender en ontvanger de beschikking hebben over een perfecte klok, en derhalve in staat stelt perfect 'een op 'een de ingangs- en uit-

gangswaarden van het kanaal te bemonsteren. In het algemeen, transporteren kanalen signalen niet alleen in de spati"ele maar ook in de tijddimensie. In het voetspoor van Baggen en Wolf, beschouwen we een belangrijke klasse van kanalen nl. opslagmedia zoals compact disc en optische of magnetische tape. Indien de snelheid waarmee een dergelijk medium wordt afgespeeld niet exact gelijk is aan de snelheid waarmee is opgenomen, ontstaat onzekerheid in de tijdrelatie tussen zender en ontvanger. Neem aan dat het kanaal een tweewaardige ingang en uitgang heeft. De lengte van een pulstrein oftewel 'run' van 'eenen of nullen kan dan worden uitgerekt of verkort onder invloed van de onjuiste bemonsteringssnelheid. Indien wordt aangenomen dat slechts de lengtes van pulstreinen wordt be"iinvloedt, maar dat niet hele pulstreinen van 'eenen of nullen verloren gaan waardoor naburige pulstreinen van nullen, resp. 'eenen zouden versmelten, zijn zender en ontvanger synchroon op het nivo van het pulstrein-volgnummer. Door het probleem te herformuleren in termen van pulstreinen als ingang en uitgang van het kanaal, kan de capaciteit van dit kanaal bepaald worden met behulp van een theorema van Verdu over de capaciteit per eenheid van kosten.

Hoofdstuk 4 heeft als onderwerp de implementatie van het Viterbi algoritme (VA). Het resultaat is geldig voor (nagenoeg) iedere toepassing van het Viterbi algoritme, niet alleen voor het decoderen van lineaire convolutiecodes. De padmetrics in het VA accumuleren de afstand tussen de ontvangen reeks kanaalsymbolen en de 'survivor'-paden in de trellis. Naarmate het aantal opgetreden kanaalfouten toeneemt, groeien de padmetrics. In een implementatie heeft het de voorkeur indien de padmetrics kunnen worden ge"implementeerd met een eindig aantal bits. Derhalve kunnen de volgende eigenschappen van het VA worden uitgebuit. 1. De selectie van de survivor-paden hangt alleen af van verschillen van padmetrics. 2. Het maximale verschil tussen twee padmetrics is eindig en kan worden begrensd met behulp van eigenschappen van de trellis. Een bekende implementatiemethode die gebruik maakt van deze twee eigenschappen is de zogenaamde 'rescaling' implementatie. Daarbij wordt na iedere iteratie van het VA de minimale padmetric van alle padmetrics afgetrokken. Zodoende blijven alle padmetrics in het bereik van nul tot en met het maximale padmetric verschil. Deze methode heeft als nadeel dat aftrekmiddelen noodzakelijk zijn in de iteratie van het VA. Dit kost hardware en rekentijd.

Het bepalen van de minimale padmetric kan evt. vermeden worden. Voorgesteld wordt een alternatieve methode die het gebruik van aftrekmiddelen overbodig maakt. Indien de padmetrics worden gerepresenteerd modulo een constante zodanig dat het bereik van de modulo operator ongeveer symmetrisch is om nul zoals bij 'two's complement arithmetic', treden weliswaar overflows op, maar indien de constante groot genoeg gekozen wordt, leidt dit niet tot algoritmische fouten. Het modulo gereduceerde verschil van twee padmetrics is namelijk gelijk aan het originele verschil mits dit originele verschil binnen het bereik van de modulo operator ligt.

In het verlengde van hoofstuk 4, wordt in hoofdstuk 5 het maximale verschil tussen twee padmetrics nader geanalyseerd voor het geval van toepassing op lineaire convolutie codes. Het resultaat is geldig voor hard decision decoding en voor een klasse van soft decision metrics. Bewezen kan worden dat het ontvangen van het codewoord bestaande uit allemaal nullen de meest kritische situatie is voor het verschil tussen padmetrics. Tevens kan bewezen worden dat dit verschil als functie van de diepte in de trellis eerst toeneemt en vervolgens (licht) afneemt. Voor de implementatie is het van belang het maximale verschil tussen candidaatwaarden voor de padmetrics te bepalen in de VA iteratie. Deze verschillen zijn namelijk groter dan die tussen de padmetrics zelf. Voor wat betreft de maximale verschillen tussen kandidaatwaarden voor de padmetrics dient een onderscheid gemaakt te worden tussen rescaling en modulo aritmetiek implementaties. Bij modulo aritmetiek hoeven namelijk alleen de verschillen tussen twee kandidaatwaarden voor eenzelfde padmetric d.w.z. voor eenzelfde node in de trellis beschouwd te worden. Hierdoor is bij modulo aritmetiek het numeriek bereik van de padmetric-waarden (iets) kleiner. Een ander resultaat is dat de maximale verschillen tussen padmetrics gereduceerd kunnen worden door een selectiecriterium. Relatief grote padmetrics komen overeen met relatief onwaarschijnlijke survivorpaden. Het kan bewezen worden dat sommige van die paden nooit het output pad van het VA kunnen zijn, omdat er altijd een ander pad is dat een kleinere padmetric heeft. Dit leidt tot een reductie van het numeriek bereik van de padmetrics en van de kandidaatwaarden voor de padmetrics. De winst bedraagt op zijn hoogst ca. een factor twee.

Curriculum Vitae

Andries P. Hekstra was born on August 9, 1961 in Breda, The Netherlands.

In May 1985 he received his M.S.E.E. degree (Cum Laude) in Electrical Engineering from the Eindhoven University of Technology. The first chapter of this dissertation is based on his graduation project.

In June 1985 he became a young graduate trainee at the European Space Operation Centre of the European Space Agency in Darmstadt, Germany, working on simulation of telecommunication systems and the design of a spread spectrum combined command and control and ranging system. Since the Fall of 1986 he was a Ph.D. student as well as a teaching and research assistent at Cornell University, Ithaca, USA. The second chapter is based on research performed during this period. September 1990 he joined the Visual Communications Research group of PTT Research, Leidschendam, the Netherlands.

At PTT Research, his work centers around information theory: data compression, error correction both for telecommuncation and postal applications, channel coding, network protocols. He is involved in various European projects, such as RACE 1018 HIVITS, RACE 2045 DISTIMA, EUREKA 625 VADIS.

Stellingen

behorende bij het proefschrift

Capacity and Coding in Digital Communications

door

Andries P. Hekstra

22 december 1994

1. De hoeveelheid afhankelijkheid zoals gemeten met de (conditionele) mutuele informatie vormt een effectief aangrijpingspunt voor een bovengrens aan het capaciteitsgebied van het twee-weg kanaal met enkelvoudige uitgang.
2. De maximale dichtheid van processoren in een programmable gate array met vertex connectivity zodanig dat elke n paren processorcellen via n disjuncte paden van communicatiecellen kunnen communiceren is $O(n^{-2})$.
3. Het capaciteitsgebied van een belangrijke klasse van kanalen met onzekere tijdrelatie tussen zender en ontvanger laat zich exact bepalen met behulp van een beschrijving in termen van pulstrein-lengtes.
4. Two's complement aritmetiek leidt tot een efficiënte implementatie van de padmetrics in het Viterbi algoritme.
5. Het numeriek bereik van de padmetrics in het Viterbi algoritme, alsmede van kandidaatwaarden hiervoor, laat zich exact analyseren voor decoders van lineaire convolutiecodes. Dit numeriek bereik kan gereduceerd worden met een eenvoudige selectieregel.
6. Het fragmentarisch contact met andere vakgebieden bevestigt de student in zijn studiekeus.
7. Een voordeel van het commerciële denken van deze tijd is dat het mensen dwingt positief over te komen.
8. Gegegeven het periodieke karakter van veel processen en ervaringen in het leven, veroorzaakt door de eindigheid van variatie en de alomtegenwoordigheid van terugkoppelmechanismen, is de vraag gerechtigd of het leven niet eenvoudiger in een frequentiedomein begrepen zou kunnen worden.
9. Hoewel de eerste en tweede wet van Newton ingang gevonden hebben in het algemeen spraakgebruik (actie-reactie wet: "wie kaatst moet de bal verwachten", aantrekkingswet: "aantrekking tussen mensen met vergelijkbare interesses"), rijst de vraag of ook Einstein's theorie niet overdraagbaar is naar interpersoonlijke relaties. Massa, geïnterpreteerd als weerstand tegen veranderingen zou, volgens Einstein, een in zichzelf gesloten vorm van energie zijn, die vrijgemaakt kan worden. De energie die vrijkomt (bijv. bij het krijgen van een inzicht) zou evenredig zijn met de omgezette massa oftewel de vrijgemaakte weerstand tegen veranderingen.
10. (Analogie van Einstein's wet van constante licht oftewel waarneming-

snelheid.) Aangezien zowel waarneming persé als tijdsbeleving direkt gekoppeld zijn aan het ervaren van een verandering in de bewustzijnstoestand van de waarnemer is de subjectieve waarnemingssnelheid, hoewel onquantificeerbaar, voor alle waarnemers (bijv. een schildpad en een vogel) gelijk.

11. Snelheidsbeperkende afbuigingen van een rijbaan in de verticale dimensie hebben de voorkeur boven horizontale afbuigingen omdat deze laatste methode verschillende soorten verkeersdeelnemers met elkaar in conflict brengt.
12. Voor het spreekwoord "iemand een oor aannaaien" zijn nieuwe toepassingen mogelijk in het kader van het voorzien van alle koeienoren met genummerde plastic labels en in het kader van het Dolby ProLogic systeem met vijf luidsprekers.