Eindhoven University of Technology

MASTER

Shannon strategies for ternary deterministic two-way channels

van der Heijden, G.J.E.L.

Award date:
1987

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Electrical Engineering
Group: Information and Communication Theory

graduate report

SHANNON STRATEGIES FOR TERNARY DETERMINISTIC TWO-WAY CHANNELS

by

G.J.E.L. van der Heijden

coach : Ir. W.M.C.J. van Overveld

EINDHOVEN, The Netherlands, August 1987

SUMMARY

A two-way communication channel is a channel with two terminals, and each terminal has the availability over an input alphabet, and can observe symbols of its output alphabet. We limit ourselves to input and output alphabets consisting of three symbols. Furthermore we have the limitation of determinism, i.e. given a pair of input symbols, then for a certain channel, there is only one possible pair of output symbols. Though some study has been done w.r.t. these channels, there is little known about strategies, that are able to use the channel efficiently.

Because of the great amount of different classes (basically each class needs a different strategy) a general method has been developed in order to compute strategies for all of these channels. This method has been tried out on two channels and the results are rather good in comparison with the Shannon innerbounds of these channels.

Also with the help of so-called dominating part-patterns, it appeared to be possible to reduce the problem of finding good strategies substantially. Using this it is possible to both reduce the number of classes and simplify the problem of finding good strategies.

With the help of the developed computing method it is now possible to compute more strategies, and evaluate these, which is an obvious task to be done in the near future. Concerning the dominating part-patterns, there remained several theoretical problems unsolved, and it would be a great step forward if one would be able to solve these.

TABEL OF CONTENTS

ACKNOWLEDGEMENTS

The author wishes to thank all members of the group information and communication theory of the Eindhoven University of Technology, i.e. Mrs. Tine Bijl, Harry Creemers, ir. Ineke van Overveld, ir. Jan Rooyakkers, prof.dr.ir. J.P.M. Schalkwijk, ir. Tjalling Tjalkens, dr.ir. Han Vinck and dr.ir. Frans Willems, for their hospitality and support during my period of research.

Especially I wish to thank my prof.dr.ir. J.P.M. Schalkwijk and my coach ir. Ineke van Overveld for their useful suggestions and advices.

Because of his help on the subject of optimization I would like to thank dr.ir. J.L. de Jong of the department of mathematics and computing science.

Last, but definitely not the least I wish to thank my parents, because without the help they gave me, I was not able to write a single letter of this report.

List of symbols:

$\sim$: Equivalence relation for tdTWC's as defined in sect. 1.2.

A : Array representing all the rectangles in the unit square after some divisions by a strategy.

$A^{(i)}$: Area of a group of subrectangles; definitions see sect. 3.5.

B_k : Approximation of G_k, $B_k = H_k^{-1}$.

$c_j(\underline{x})$: Constraint function; general form $c_j(\underline{x}) \leq 0$.

dTWC : Deterministic two-way channel.

$\underline{d}_k$: Direction vector for the k-th iteration step of the optimization process.

$g_k = g(\underline{x}_k)$: Gradient of the rate function, in the point $\underline{x}_k$.

G : Capacity region.

G_i, G_o : Resp. the Shannon inner- and outerbound region.

$G_k = G(\underline{x}_k)$: Hessian matrix of the the rate function in the point $\underline{x}_k$.

$H^{(i)}$: Heigth of a group of subrectangles; definitions see sect. 3.5.

H_k : Approximation of G_k^{-1}, $H_k = B_k^{-1}$.

$L^{(i)}$: Length of a group of subrectangles; definitions see sect. 3.5.

n : Number of transmissions. In sect. 3.4 it means the number of thresholds, instead of N_n.

m_i $(i=1,2)$: Message to be sent by T_i.

N_n : Number of thresholds needed for an n-stage strategy.

nsq : Number of distinct input words, a terminal can transmit, for a fixed length strategy.

R_{12} : The signalling rate of the code from T_1 to T_2.

R_{21} : The signalling rate of the code from T_2 to T_1.

S_i : Set, containing all the rectangles, that are characterized by a given $\underline{x}_2$ and $\underline{y}_2$ sequence.

$S^{\delta}_{i.j}$: Representative of the equivalence class under the relation ~ (see also [4]).

T_i (i=1,2) : Terminal with nr. i.

tdTWC : Ternary deterministic two-way channel.

TWC : Two-way channel.

x_i : Threshold nr. i ($i \leq N_n$).

$\underline{x}_i$ (i=1,2) : Sequence of n input symbols sent by T_i.

$\underline{x}_k$: Sequence containing the threshold values after the k-th optimization iteration.

x_{ij} : j-th symbol of the input sequence $\underline{x}_i$.

$\underline{y}_i$ (i=1,2) : Sequence of output symbols received by T_i.

y_{ij} : j-th symbol of the output sequence $\underline{y}_i$.

$\underline{\gamma}_k$: Defined by : $\underline{g}_{k+1} - \underline{g}_k$.

$\underline{\delta}_k$: Defined by : $\underline{x}_{k+1} - \underline{x}_k$.

Chapter 1. Introduction

1.1 General description of the situation

A two-way communication channel can be represented as shown in fig. 1.1. It consists of two terminals T_1 and T_2 and a two-way communicating channel (Also called two-way channel or TWC). The channel has input alphabets $A_1 = \{0,1,\dots,|A_1| -1\}$ and $A_2 = \{0,1,\dots,|A_2| -1\}$ and output alphabets $B_1 = \{0,1,\dots,|B_1| -1\}$ and $B_2 = \{0,1,\dots,|B_2| -1\}$. Each terminal can transmit a symbol x_i (i=1,2) of its input alphabet and observe its channel output symbol y_i (i=1,2). The output symbols y_1,y_2 are determined by the transition probability distribution $P(y_1y_2|x_1x_2)$ of the TWC, if the channel is assumed to be memoryless, i.e. previous in- and outputs do not affect the transition probabilities.

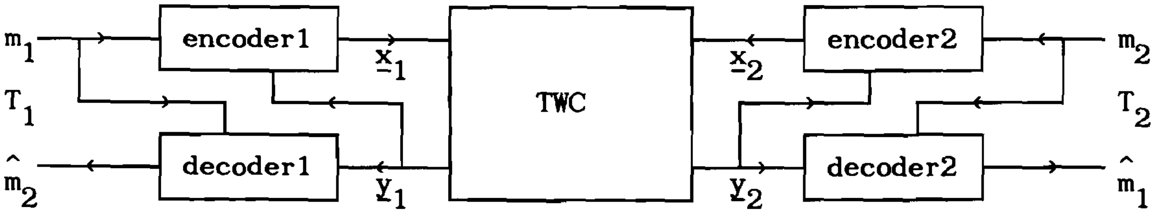


fig. 1.1 A two-way communicating system

The general coding scheme for this channel is based on two sources, each picking a message from their message sets $C_1 = \{0,1,\dots,M_1-1\}$, $C_2 = \{0,1,\dots,M_2-1\}$, with probabilities $p(m_1)$ and $p(m_2)$ that are mutually independent. T_1 starts sending an input symbol x_{11} according to m_1 and at the same time T_2 sends x_{21} according to m_2. They both receive an output symbol resp. y_{11} and y_{21} and next they can start sending their second symbol. The choice of these symbols does not have to be based on the chosen messages only, but it is also possible to make use of knowledge about the previous received outputs, thus creating a dependency of the channel input distribution $P(x_1x_2)$. Extending this idea for n transmissions leads to the following encoding scheme:

T_1	T_2
$x_{11}=f_1(m_1)$	$x_{21}=g_1(m_2)$
$x_{12}=f_2(m_1,y_{11})$	$x_{22}=g_2(m_2,y_{21})$
$\vdots$	$\vdots$
$x_{1n}=f_n(m_1,y_{11}..y_{1n-1})$	$x_{2n}=g_n(m_2,y_{21}..y_{2n-1})$

The decoding procedure for T_1 (T_2) has the duty to make an estimate of the message that was send by T_2 (T_1) by making use of the channel output sequence it has received and the knowledge about the transmitted inputs of T_1 (T_2). This leads to the following decoding scheme:

$$\hat{m}_2=\varphi(m_1,y_{11}..y_{1n}) \qquad \hat{m}_1=\psi(m_2,y_{21}..y_{2n})$$

Notice that the input sequence $x_{i1},...,x_{in}$ is fully determined by m_i and the outputsequence $y_{i1}..y_{in-1}$ (i=1,2).

The rates of these coding strategies are defined by:

$$R_{12}=\frac{H(m_1)}{n} \quad \text{and} \quad R_{21}=\frac{H(m_2)}{n} \qquad (1.1.1)$$

$$\text{with } H(m_i) = \sum_{m_{ij} \in C_i} -P(m_{ij})\cdot \log P(m_{ij}) \quad (i = 1,2)$$

Given a TWC and a strategy, it is possible to compute the rate and the average decoding error probabilities P_{e12} and P_{e21} of this strategy. We will say that a point (R_{12},R_{21}) belongs to the capacity region G of a given memoryless channel K if, given any $\epsilon > 0$, there exists a block code and decoding system for the channel with signalling rates R'_{12} and R'_{21} , satisfying $|R_{12} - R'_{12}| < \epsilon$ and $|R_{21} - R'_{21}| < \epsilon$ and such that the error probabilities satisfy $P_{e12} < \epsilon$ and $P_{e21} < \epsilon$.

It can easily be seen, that the capacity region is not affected by permutation of in- and output-symbols. So if there are two TWC's and one can be constructed from the other by permutation of symbols, then they both have the same capacity region. Exchanging the numbering of the terminals however, leads to the capacity region being reflected in the line $R_{12}=R_{21}$.

TWC's that have the property of $y_1=y_2$ are called T-channels. They can be depicted by a channel with two inputs and a single output, which can be observed by both terminals (see fig. 1.2).

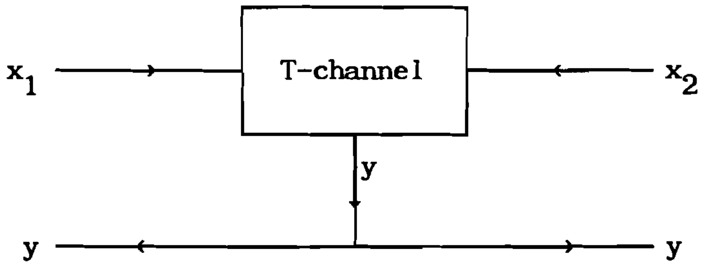


fig. 1.2 Model of a T-channel

We will restrict ourselves to deterministic TWC's (dTWC's); these are channels with $P(y_1y_2|x_1x_2)$ being either 0 or 1, i.e. the channel outputs y_1y_2 are uniquely determined by the channel-inputs x_1x_2.

In [8] Shannon derived bounds for the capacity region G. He proved that for every TWC the following holds:

$$\text{Let}\quad G_i = \text{co}\ \{\ (R_{12},R_{21})\ |\ 0 \le R_{12} \le I(X_1;Y_2|X_2),\ 0 \le R_{21} \le I(X_2;Y_1|X_1),\ P(x_1x_2) = P(x_1)P(x_2)\ \} \qquad (1.1.2a)$$

$$\text{and}\quad G_o = \{\ (R_{12},R_{21})\ |\ 0 \le R_{12} \le I(X_1;Y_2|X_2),\ 0 \le R_{21} \le I(X_2;Y_1|X_1),\ \text{arbitrary } P(x_1x_2)\ \} \qquad (1.1.2b)$$

With $I(X;Y|Z) = \sum_{x,y,z} P(xyz)\cdot\log \frac{P(x|yz)}{P(x|z)}$

and co meaning convex hull.

Then $G_i \subset G \subset G_o$.

Usually G_i is called the Shannon innerbound and G_o is called the Shannon outerbound.

1.2 Ternary dTWC's and partition patterns

A ternary dTWC (tdTWC) is a dTWC, with input- and output alphabets consisting of three symbols, e.g. $x_1, x_2, y_1, y_2 \in \{0,1,2\}$. Such a channel can be represented by two matrices (one for y_1 and one for y_2), with x_1 and x_2 as entries and y_1 or y_2 as values in the matrix.
An example of such a channel is given in fig. 1.3.

$$\begin{array}{cc|ccc} & & & x_2 & \\ & & 0 & 1 & 2 \\ \hline & 0 & 0 & 1 & 1 \\ x_1 & 1 & 2 & 2 & 1 \\ & 2 & 0 & 0 & 0 \\ & & & y_1 & \end{array} \qquad \begin{array}{cc|ccc} & & & x_2 & \\ & & 0 & 1 & 2 \\ \hline & 0 & 1 & 0 & 1 \\ x_1 & 1 & 2 & 0 & 1 \\ & 2 & 2 & 1 & 2 \\ & & & y_2 & \end{array}$$

fig. 1.3 Output matrices of a tdTWC

Gaal [3] and Jacobs [4] have already done some research on these channels, and a summary of their results is presented here.

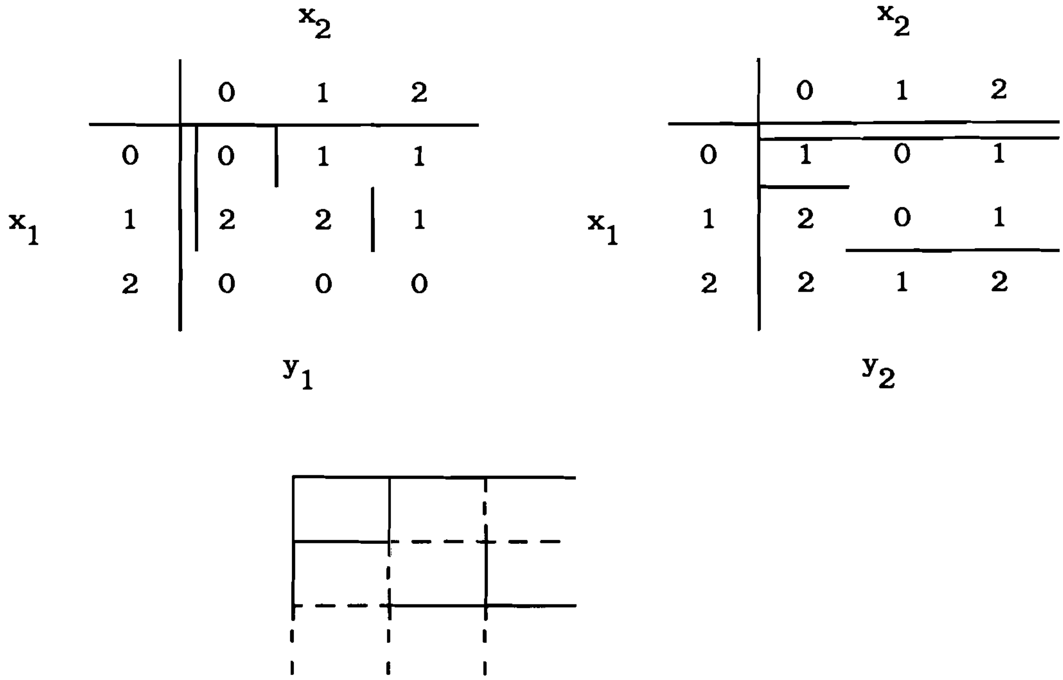
By counting the number of possibilities of the entries of the output matrices, we find that there are: $3^{2\cdot 9}$=387 420 489 of these channels. Since a lot of these channels have the same capacity region Gaal defined the following equivalence relation:

Two tdTWC's are equivalent under the relation ~ iff one output matrix can be derived from the other

1. by permutation of the alphabet of one or more of the inputs or outputs.
2. by exchanging the numbering of the terminals (1 and 2).
3. by combining operations 1. and 2.

Note that the channels which are equivalent under ~ have either identical capacity regions or reflected capacity regions with respect to the line $R_{12}=R_{21}$. Gaal was able to count the number of ternary deterministic T-channels. He found that there are 84 of these classes.

Another way of representing a tdTWC is the so-called partition pattern. The basic idea behind this concept is that a terminal (e.g. T_1) knowing its transmitted input (x_1) and the received output (y_1) is able to determine which inputs of the other terminal (x_2) were possibly sent. Given an input (e.g. x_1), there are three other inputs ($x_2 \in \{0,1,2\}$) that can cause an output y_1. If two different x_2 inputs cause two different y_1 (fixed x_1), then in the output matrix (see fig. 1.4) a line is drawn between these two outputs, where the inputs '0' and '2' are considered to be neighbours. When this is done for all outputs in both output matrices, we can omit the matrices. This gives us two line patterns (one with horizontal and one with vertical lines) that can be packed together. The result is called the partition pattern. Notice that permutations of symbols do not affect the partition pattern of a tdTWC; this can only be done by exchanging the numbering of the terminals or permutation of rows and/or columns. It is now much easier to see whether two channels are equivalent or not.



x_2

x_1	0	1	2
0	0	1	1
1	2	2	1
2	0	0	0

y_1

x_2

x_1	0	1	2
0	1	0	1
1	2	0	1
2	2	1	2

y_2

fig. 1.4 Construction of a partition pattern

With the help of these partition patterns Jacobs was able to count the number of equivalence classes of the tdTWC's. She found that there are 322 of these classes. In [4] a representative of each class is listed and also the Shannon inner- and outerbounds are plotted.

1.3 Motivation

Despite of the fact that the problem of equivalence classes of the tdTWC's may said to be solved, little is known about the achievable rates of the various channels. In many cases there appears to be a large gap between the Shannon innner- and outerbounds. A method of reducing this gap, is to look for strategies with a rate above the innerbound.

A second reason for trying to find good strategies is, that this method does not only result in a figure, which is said to be achievable. It also delivers a method that yields the promised ratepair.

There are basically two types of strategies known. The first type is the so-called Markov strategy; examples of such a strategy can be found in [3],[6] and [7]. The second type is the Shannon strategy (see e.g.[9]). Markov strategies,though sometimes yielding high rates, have the disadvantage that much insight in the behaviour of the channel is needed. Since there still are so many different classes of tdTWC's a general approach to this problem is unlikely to be found. Shannon strategies however, allow a more general approach, though the problem is still quite complex. Once some good Shannon strategies have been found for several tdTWC's, the results can be used to gain insight into these channels, and with this help it might be possible to construct good Markov strategies.

Chapter 2. Shannon strategies

2.1 Coding for one-way channels

In information theory messages sometimes are regarded as subintervals of the unity interval: [0,1) (see fig 2.1). Using this representation. the probability of a message being chosen is the length of the subinterval corresponding to that message. A code can thus be regarded as a scheme for partitioning in the [0,1)-interval.The received symbols allow the receiver to make divisions in the [0,1)-interval. After several symbols have been received, the unity-interval has been divided into a number of subintervals. If such a subinterval is a part-interval of one of the message-intervals, then the corresponding message is known to the receiver.

[——1——)[———2———)[—3—)[———4———)[———5———)
0 1

fig. 2.1 Division of the [0,1)-interval into 5 message-intervals

Let n be the number of transmissions, P_i the length of the i-th subinterval after n transmissions and M the total number of subintervals, then we have solved a certain amount of uncertainty in these n transmissions, the average quantity of uncertainty reduction/transmission will also be called the rate and can be computed using:

$$R = \frac{1}{n} \cdot \sum_{i=1}^{M} -P_i \cdot \log P_i \qquad (2.1.1)$$

2.2 Two-way channel strategies

If we are dealing with two users (T_1 and T_2), each having a message (resp. m_1 and m_2) to send, this can be represented as two subintervals from two different [0,1)-intervals, an interval for each user. This can be packed together into the two dimensional representation of the unit square (see fig. 2.2). Now the message-pair (m_1,m_2) can be represented as a subrectangle of the unit square. Similarly to the previous case coding can be seen as a method to divide the unit square. But now a division of the square can also be done by regarding the interval (message) a user wants to send, because this already determines the measurements of the wanted subrectangle in one direction.

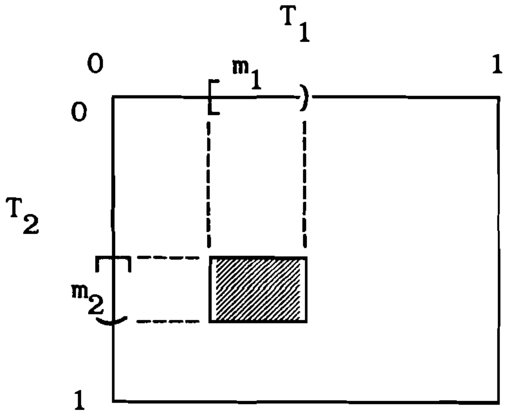


fig. 2.2 An example of a message rectangle

Example 1: The binary multiplying channel

The binary multiplying channel (BMC) is defined by:

$$x_1,x_2,y_1,y_2 \in \{0,1\} \qquad \text{and} \qquad y := y_1 = y_2 = x_1 \cdot x_2 \qquad (2.2.1)$$

The input probabilities for the first transmission, $P(x_{11}=0) = 1-P(x_{11}=1)$ and $P(x_{21}=0) = 1-P(x_{21}=1)$ are independent by the assumption of m_1 and m_2 being independent. In the square these probabilities are determined by the location of the thresholds t_1 and t_2, e.g. if the m_1 interval lies in $[0,t_1)$ then $x_{11}= 0$ otherwise $x_{11}= 1$. After one transmission the square divides as shown in fig. 2.3.

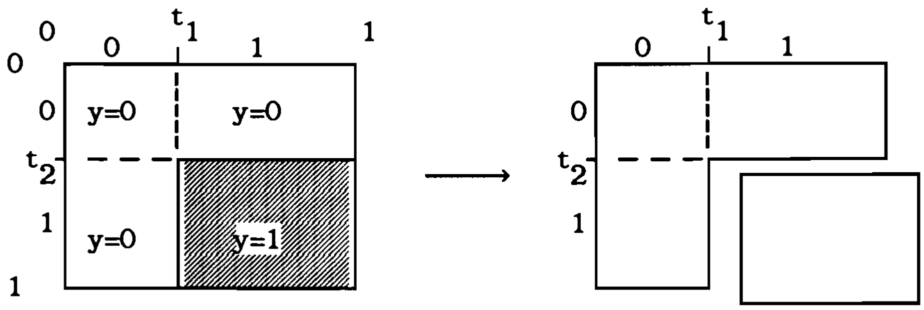


fig. 2.3 First transmission with the BMC

For the second transmission the probability distribution of the input symbols is not only determined by the message to be sent, but also by the received output. So $P(x_{12}=0|y_1=0, x_{11}=1)$ does not necessarily have to be equal to $P(x_{12}=0|y_1=1, x_{11}=1)$. This means that for each previously received outputsequence, the threshold values are allowed to be chosen independently within the range of the interval determined by the thresholds of the previous stage. Fig. 2.4 gives an example of how the various input probabilities for the second transmission could be chosen.

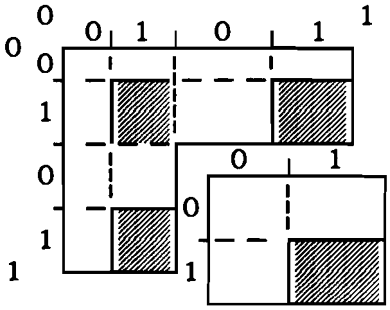


fig. 2.4 BMC division of the unit square after two transmissions

Thus continuing the unit square gets more and more divided into pieces until an area remains which is a subset of one of the message rectangles.

Since this channel is a T-channel, each user knows the output the other one receives. So they both know which subregion of the square the next transmission is related to. For non T-channels the situation is somewhat different. To illustrate this another example is given.

Example 2: a ternary non T-channel

Consider the ternary channel characterized by the partition pattern shown in fig. 1.4.

After one transmission the square apparently divides into two subparts, a square and an L-shape, as shown in fig. 2.5.

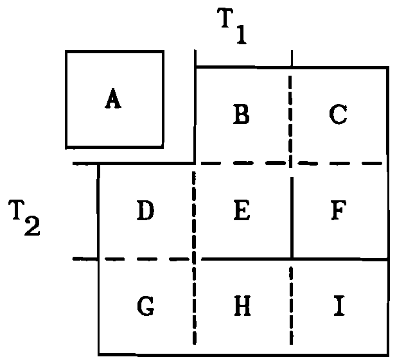


fig. 2.5 Square division when using a ternary non T-channel

When this is compared to example 2.1 it is seen, that for the BMC the square also divides into a square and an L-shape after the first transmission. But there is a difference: the L-shape of the second channel contains partitions. Due to these partitions it is possible to achieve a higer rate for this first transmission (compare $R := \frac{1}{2}(R_{12}+R_{21})$ Example 2.1 R=0.38925 trits, Example 2.2 R=0.52716 trits). A method of vizualizing this difference is to 'unfold' the L-shape (see fig. 2.6), i.e. given an input of T_1 we consider the outputs y_1 corresponding to the various inputs of T_2. If two different x_2's result the same y_1 then the corresponding subsquares are placed in the same column (see fig. 2.6 e.g. subelements C and F.) In case two different x_2's yield a different output, then the corresponding subsquares are placed in different columns(see fig. 2.6 e.g. subelements F and I). Each column corresponds to a distinguishable different situation for T_1 i.e. the probability distribution for the next symbol being transmitted does not have to be the same in two distinct columns. Of course a similar discussion with respect to T_2, leads to rows (instead of columns) for the various different situations.

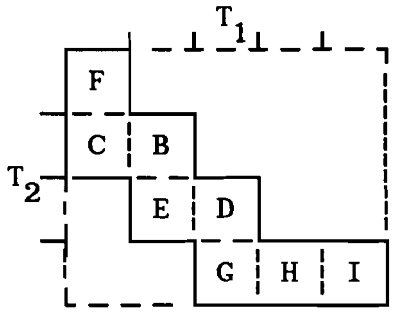


fig. 2.6 The unfolded L-shape of fig 2.4

The second transmission divides the unfolded L-shape into two subareas, corresponding to the subsquare and the L-shape of the first transmission. The 'L-shape part' can again be unfolded as shown in fig. 2.7.

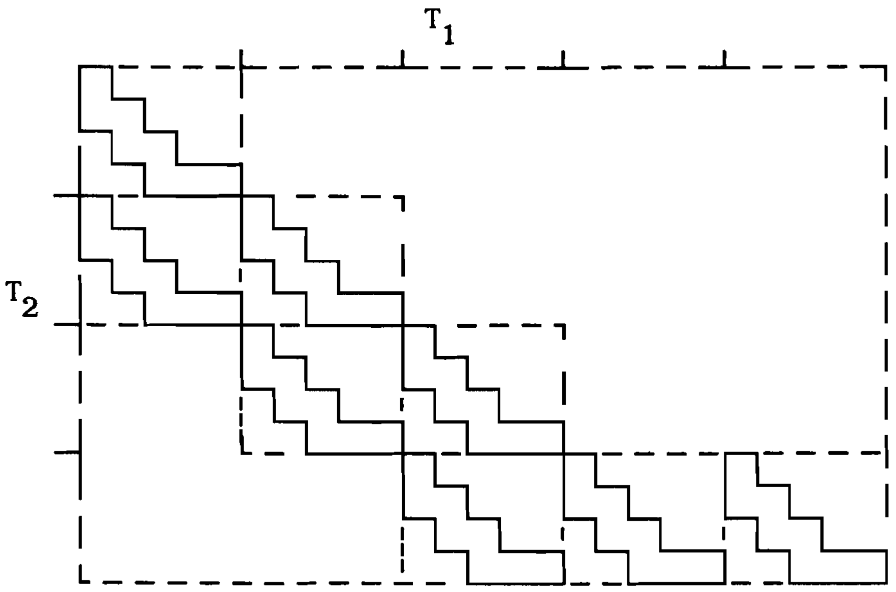


fig. 2.7 The unfolded shape after two transmissions

A Shannon strategy of n stages will sometimes be denoted as a K_n stategy. Similar to sect. 2.1, we will also call the average uncertainty reduction the rate of the strategy, and then the rates of a K_n strategy can be computed using:

$$R_{12}(n) = \frac{1}{n} I(m_1;\underline{Y}_2|m_2) = \frac{1}{n} H(\underline{Y}_2|m_2) \qquad (2.2.2a)$$

$$R_{21}(n) = \frac{1}{n} I(m_2;\underline{Y}_1|m_1) = \frac{1}{n} H(\underline{Y}_1|m_1) \qquad (2.2.2b)$$

with $\underline{Y}_i := (y_{i1},y_{i2},\ldots\ldots,y_{in})$ and $i = 1,2$

In [8] Shannon proved, that for $n \longrightarrow \infty$: $(R_{12}(n),R_{21}(n))$ can achieve every point of the capacity region, and conversely, a point outside this region cannot be reached.

Chapter 3. A method for computing good strategies

3.1 Introduction

When one wants to determine, whether a strategy is good, one has to calculate its rate, and compare this to the Shannon inner and outerbounds. The general way to find a good strategy, is to start with an arbitrary strategy, and evaluate its rate. Then slightly change the strategy and compute the rate, and repeat this process until a good strategy has been found. In this chapter a method of evaluating the rate is described, and also some good numerical optimization methods will be discussed.

3.2 The number of thresholds

In order to get some insight in the computing complexity of the problem we will calculate the amount of thresholds to be dealt with, when computing n stage strategies. With respect to the first transmission, 4 thresholds can be chosen independently. Considering the worst case situation, after one transmission there are 9 different states (3 different inputs, and for each input there are 3 distinguishable outputs). For each state a new set of 4 thresholds can be chosen independently. Let N_n denote the number of thresholds needed for an n stage strategy of some tdTWC, then:

$$N_2 \leq 4 + 4 \cdot 9 = 40 \qquad (3.2.1)$$

Each of these 9 states divides into 9 substates after the second transmission, and for each of these substates again 4 thresholds can be chosen. Thus continuing we get for N_n:

$$\begin{aligned} N_n &\leq 4 + 4\cdot 9 + 4\cdot 9^2 + \ldots\ldots\ 4\cdot 9^{n-1} \\ &= \tfrac{1}{2}\cdot(9^n - 1) \end{aligned} \qquad (3.2.2)$$

The exact value of N_n for a given partition pattern can be determined by observing that after a transmission T_1 recognizes, say d_1 different states and T_2 recognizes d_2 different states. Then:

$$\begin{aligned} N_n &= 2\cdot(d_1{}^0 + d_2{}^0) + 2\cdot(d_1{}^1 + d_2{}^2) + 2\cdot(d_1{}^2 + d_2{}^2) + \ldots\ldots \\ &\ldots + 2\cdot(d_1{}^{n-1} + d_2{}^{n-1}) = 2\cdot\left(\frac{d_1{}^n - 1}{d_1 - 1} + \frac{d_2{}^n - 1}{d_2 - 1}\right) \end{aligned} \qquad (3.2.3)$$

3.3 Computing the rate

The algorithm consist of three parts:

1. The over all initializing part: In this part the data structure is built up and initialized.
2. The evaluation set up part: Here the data structure is adapted for a given partition pattern.
3. The rate evaluation: Given a threshold setting the rate will be computed.

3.3.1 The basic data structure

After one transmission the unit square is divided into 9 subrectangles. Each of these rectangles is characterized by its (x_1,x_2) input combination. In general for an n stage strategy there are nsq:=3^n input combinations for each terminal. This can be represented by an nsq•nsq array A. Each element of A can be characterized by the input sequences $\underline{x}_1$ and $\underline{x}_2$, and contains the output sequences $\underline{y}_1$ and $\underline{y}_2$ corresponding to $\underline{x}_1\underline{x}_2$. Fig. 3.1 shows a graphical representation of this array for K_2 (In app. B the situation for K_3 is shown). From this picture it is seen that each rectangle (except for the ones on the border) is surrounded by 4 thresholds. In this part of the algorithm it is assumed, that we are dealing with the maximum number of thresholds. We can number these thresholds the way it is done in fig. 3.1.

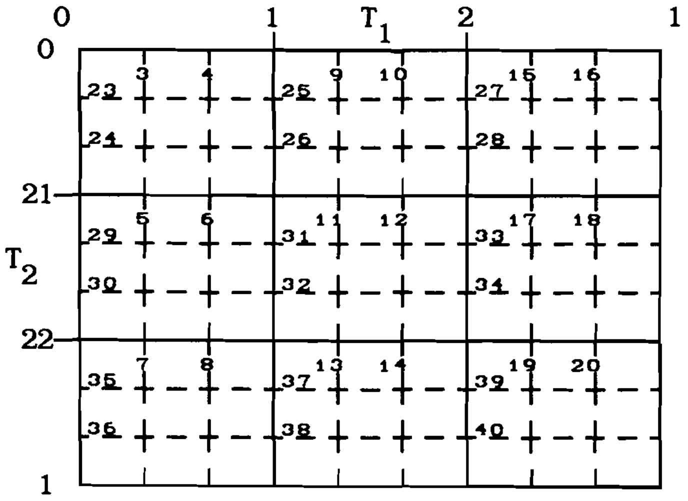


fig. 3.1 Numbering the thresholds for K_2 strategies

The final part of this general initialization is the assignment to each matrix element of the numbers of the thresholds it is surrounded by. Summarizing we have an nsq•nsq matrix, where each element contains six values: 2 output sequences and 4 threshold numbers.

3.3.2 The evaluation set-up

In this part the rate evaluator is prepared for a given partition pattern. After reading a partition pattern the output matrices for y_1 and y_2 are constructed. With the help of these matrices the $\underline{y}_1$ and $\underline{y}_2$ elements of the A array are easily calculated. As shown in the previous chapter, given an input sequence, a terminal can only choose different threshold values if it observes different output sequences. After one transmission we can look for states which cannot be distinguished by a user, e.g. T_1. If there are such states, then the values of the corresponding thresholds, determined by T_1, must be the same. E.g. consider the channel of fig. 1.4 and assume that the thresholds have been numbered, the way it is done in fig. 3.1. Suppose T_1 sends a '0', then T_1 cannot distinguish the '1' symbol of T_2 from the '2' symbol. Therefore in fig. 3.1 the thresholds with the numbers 5 and 7 must have the same value. So there is a set of thresholds, which must have the same value. We will call one element of this set independent, and we will say the other elements to be coupled to it; an independent element can take every value, but a coupled element must have the value of the element it is coupled to. By comparing output sequences, sets of coupled thresholds can be listed. We will choose the one with the smallest index to be the independent one and couple all the others to it. This can be stated in a list of N_n elements, with on the i-th place the number i if threshold i is independent, and else the index of the threshold, threshold i is coupled to. Note that this number is always smaller than i. Now for each element of A the values of the surrounding thresholds have been adapted to the regarded partition pattern.

3.3.3 The calculation of the rate

The information structure we have built up will now be used in this part of the algorithm in order to compute a rate achieved by a given thresholdsetting (this is an array $x_1,\dots,x_n$ with $0 \leq x_i \leq 1$ and $i \in \{1,\dots,N_n\}$ and is not be confused with x symbols used for the TWC inputs).

The first problem dealt with is the fact that the constraints $0 \leq x_i \leq 1$ are not sufficient. Each threshold (see fig. 3.1) is limited on both sides by either the edge of the unit square or another threshold. To solve this a coordinate transform is performed on the threshold values. Because the constraint $x_{2k-1} \leq x_{2k}$ $(k=1,\dots,N_n/2)$ always has to be satisfied (see fig. 3.1), the coordinate transform starts with the replacement:

$$x_{2k-1} := x_{2k-1} \cdot x_{2k} \qquad (k=1,\dots,N_n/2) \qquad (3.3.1)$$

If each pair of thresholds (x_{2k-1}, x_{2k}) has to stay within the interval $[x_a, x_b]$ (e.g. In fig. 3.1 the thresholds x_{11} and x_{12} have to stay within the interval $[x_1, x_2]$), then they are transformed using the following expression:

$$x_i := (x_b - x_a) \cdot x_i + x_a \qquad (3.3.2)$$

If this is done properly i.e. one starts with the thresholds of the first stage, then the second etc. until the the n-th stage, all thresholds will have the right value.

Since an element of A corresponds to a rectangle in the unit square, that remains after several transmissions, and since we are now able to determine the exact values of the thresholds, that surround this rectangle, the measurements of the rectangles can be calculated easily.

Now only formula (2.2.2) has to be adapted to our data structure. Considering only R_{12} (the derivation and interpretation for R_{21} is very similar), we have

$$\begin{aligned} R_{12} &= \frac{1}{n}\, H(\underline{Y}_2|m_2) = \frac{1}{n}\, [H(y_{21}|x_{21}) + H(y_{22}|y_{21}x_{22}x_{21}) + \cdots \\ &\qquad \cdots + H(y_{2n}|y_{2n-1}\cdot\cdot y_{21}x_{2n}\cdots x_{21})] \\ &= \frac{1}{n}\, [\sum_{\underline{x}_2,\underline{y}_2} -P(\underline{x}_2,\underline{y}_2)\cdot(\ \log P(y_{21}|x_{21}) + \\ &\qquad +\log P(y_{22}|y_{21}x_{22}x_{21}) + \cdots\cdot \\ &\qquad \cdots\cdot\cdot +\log P(y_{2n}|y_{2n-1}\cdot\cdot y_{21}x_{2n}\cdots x_{21})\ )] \\ &= \frac{1}{n}\, [\sum_{\underline{x}_2,\underline{y}_2} -P(\underline{x}_2,\underline{y}_2)\cdot\log\ (P(y_{21}|x_{21})\cdot \\ &\qquad P(y_{22}|y_{21}x_{22}x_{21})\cdot\cdots \\ &\qquad \cdots\cdot P(y_{2n}|y_{2n-1}\cdot\cdot y_{21}x_{2n}\cdots x_{21})\ )] \end{aligned}$$

(3.3.3)

When we interpret this expression geometrically, then it is easily seen, that

$$P(\underline{x}_2,\underline{y}_2) = \sum_{\underline{x}_1} P(\underline{x}_1,\underline{x}_2,\underline{y}_2)$$

is the sum of the areas of the rectangles, determined by a given $\underline{x}_2$ and $\underline{y}_2$.

The product in the log argument of (3.3.3) seems rather complicated but has a very simple geometrical interpretation. For the first transmission $P(y_{21}|x_{21})$ is the sum of the length of all the subrectangles which have the same x_{21} and y_{21}. $P(y_{22}|y_{21}x_{22}x_{21})$ is the sum of the length of all the subrectangles with a given $y_{22}y_{21}x_{22}x_{21}$, but it has been normalized with respect to $P(y_{21}|x_{21})$. So $P(y_{21}|x_{21})\cdot P(y_{22}|y_{21}x_{22}x_{21})$ actually is the sum of the length of all the subrectangles with given $y_{22}y_{21}x_{22}x_{21}$. Extending this result for n transmissions gives us the following algorithm for the computation of R_{12}.

Initialize the variable R by putting it to '0'.

Take a fixed value for $\underline{x}_2$; this corresponds to only regarding one row of the A matrix.

Gather all elements of this row with the same $\underline{y}_2$.

Because the values of the surrounding thresholds are known, the areas, lengths and heights of these elements can be computed easily. Calculate the sum of the areas and lengths of the gathered elements. Call these resp. Q and L.

Calculate $-Q \cdot \log L$ and add this to R.

Repeat this process for all possible $\underline{x}_2, \underline{y}_2$ combinations.
R now contains the value of R_{12} and for one direction the rate is known.

The calculation of R_{21} can be done in an analogue way.

3.4 Optimization

3.4.1 Introduction

In the previous section we have seen, that given a partition pattern and a threshold distribution we are able to calculate the rate function. In order to construct a good strategy we have to find values for the thresholds that yield high rates. This problem is the commonly known problem of finding an optimum of a function of n variables over a restricted area. In literature many solutions to this problem are described, from which one has been selected. We will discuss the chosen method in this section.

3.4.2 Statement of the problem

Given:

$$R := R_{12} + \lambda R_{21} = f(x_1, \ldots\ldots, x_n) \quad \text{with} \quad 0 \leq x_i \leq 1 \quad \text{and} \quad i \in \{1, \ldots, n\} \qquad (3.4.1)$$

In this chapter we will only consider the case for $\lambda=1$.
Find the maximum of $f(\underline{x})$.

Concerning the gradient $\underline{g}(\underline{x}):=(\frac{\partial f}{\partial x_1},\ldots\ldots,\frac{\partial f}{\partial x_n})$:

If $x_i \in (0,1)$, then f is partially differentiable with respect to x_i, but if $x_i=0$ or $x_i=1$ then $\partial f/\partial x_i$ can become positive or negative infinite.

Since in literature (e.g. [2],[5] or [10]) usually the analogous problem of finding the local minimum is described, we will assume, that we are looking for the minimum, when discussing optimization methods. Note that this problem is basically the same, because finding a maximum of R is equivalent with finding a minimum of -R. Also by convention we will assume the constraints $0 \leq x_j \leq 1$ to be written as $c_i(\underline{x}) \geq 0$ $(i \in \{1,\ldots,2n\})$.

3.4.3 The general algorithm

If there were no constraints the general algorithm would consist of two parts:

1. The line search algorithm: Given a direction vector $\underline{d}$ and a start vector $\underline{x}$; find an α such that $f(\underline{x} + \alpha\, \underline{d})$ is minimum.
2. Determination of the direction vector $\underline{d}$.

These two steps are repeated until no further minimization can be done. A good choice for $\underline{d}$ can substantially reduce the number of iterations.

Because the constraints in our problem are not complex, (the constraints are inequalities of first order and in only one variable) they are not too difficult to deal with. Given a point $\underline{x}$ then there can be constraints for which the equality holds, e.g. $x_i=0$ for some i. These constraints are called active. Constraints, satisfying the inequality will thus be called passive. The algorithm commonly used when dealing with inequality constraints is first to look whether a constraint is active or not. If it is active, then the constraint is kept active, which implies that during the next iteration steps the equality will hold. If it is passive, then the choice of the search direction will not be influenced by the constraint, unless the constraint tends to be violated. In that case the constraint is made active, and will be kept active like all the other active constraints. After several iterations, the minimization algorithm will find a minimum under the condition of the active constraints. Following Lagrange we now have for the constrained optimum:

$$\sum_{j \in S} \lambda_j \cdot \nabla c_j(\underline{x}) - \nabla f(\underline{x}) = 0 \quad \text{(for some } \lambda_j\text{)} \qquad (3.4.2)$$

S is the set containing the indices of the active constraints.

Notice that in our case $\nabla c_j(\underline{x})$ is a vector with zeros on all positions except one. On that position it has the value +1 or -1.

If there is some $\lambda_j < 0$ then we can minimize $f(\underline{x})$ further by making $c_j(\underline{x})$ passive. So we deactivate this constraint, find a new minimum and look for another constraint that can be deactivated. This is repeated until there are no more active constraints $c_j(\underline{x})$ with $\lambda_j < 0$.

Summarizing we get the following algorithm:

0. Generate a starting point $\underline{x}_o$ within the given domain; k:=0.
1. Determine which constraints are active and compute their gradients.
2. Evaluate $f(\underline{x}_k)$ and $g(\underline{x}_k)$.
3. Calculate a new and valid search direction $\underline{d}_k$.
4. Determine the value of α_{max}, i.e. the smallest value of α, where $\underline{x}_k + \alpha\ \underline{d}_k$ violates a constraint.
5. Perform a line minimization, i.e. find an $\alpha \in [0,\alpha_{max}]$ for which $f(\underline{x}_k + \alpha\ \underline{d}_k)$ is minimum.
6. If $\alpha = \alpha_{max}$ then activate corresponding constraint.
7. If the constrained optimum has not yet been found then increase k by 1 and goto 2.
8. If a constraint can be deactivated, then make it passive, increase kby 1 and goto 2.
9. Local minimum has been found.

In the next sections we will discuss the methods of line minimization and calculating the $\underline{d}$-vector.

3.4.4 The determination of the direction vector

One of the most obvious methods for choosing a direction vector is the method of steepest descent, i.e. choose for $\underline{d}_k$:

$$\underline{d}_k = -g(\underline{x}_k) \qquad (3.4.3)$$

This method, though working well in the first stages of the iteration process, becomes very bad when $\underline{x}_k$ is in the neighbourhood of the local minimum $\underline{x}^*$.

A better method is achieved by using the second order derivatives. For a function of n variables, the second order derivatives can be put in n•n matrix G, called the Hessian, with on position i,j: $\frac{\partial^2}{\partial x_i \, \partial x_j} f(\underline{x})$.

This Hessian is a symmetric matrix, and in a minimum it has to be positive definite.

Given a point $\underline{x}_k$ then, with the help of the Taylor series, we get for $\underline{x}$ in the neighbourhood of $\underline{x}_k$:

$$f(\underline{x}) \approx f_k + g_k^T \cdot (\underline{x} - \underline{x}_k) + \frac{1}{2} \cdot (\underline{x} - \underline{x}_k)^T \cdot G_k \cdot (\underline{x} - \underline{x}_k) =: y(\underline{x})$$

$$\begin{aligned} \text{with} \quad & G_k := G(\underline{x}_k) \\ & g_k := g(\underline{x}_k) \qquad\qquad (3.4.4) \\ & f_k := f(\underline{x}_k) \end{aligned}$$

Assume $\underline{x}_k$ to be near the minimum (G_k is positive definite), then the minimum of $y(\underline{x})$ is a good approximation for the minimum of $f(\underline{x})$. The minimum of $y(\underline{x}_k)$ is found by solving:

$$\begin{aligned} & G_k \cdot (\underline{x} - \underline{x}_k) + g_k = 0 \quad \Leftrightarrow \\ & \underline{x} = \underline{x}_k - G_k^{-1} \cdot g_k \qquad\qquad (3.4.5) \end{aligned}$$

Which gives us the iterative formula:

$$\underline{x}_{k+1} = \underline{x}_k - G_k^{-1} \cdot g_k \qquad\qquad (3.4.6)$$

Note that in this case no line search is needed, but the method demands $\underline{x}_k$ close to the minimum. Better convergence results can be obtained by considering $-G_k^{-1} \cdot g_k$ as a direction vector, and then perform a line search. This method is usually called the Newton method (see [2]), and though it has very good convergence properties, it often is (and also in our case) very impractical, since it makes use of the Hessian, which is sometimes hard to compute. And even if one is able to compute G_k, problems could arise in determining G_k^{-1}.

In order to deal with these problems, the so-called quasi-Newton methods have been developed. The principle of these methods is to try to find out information about the Hessian during the search process and to use this information for determining a new direction of search. A matrix H is used, which in the beginning is equal to the identity matrix, and is updated after every line minimization. An updating formula for H is wanted , such that H_k approximates G^{-1} for increasing k. So there is no need for knowledge of the Hessian, neither a complicated matrix inversion has to be performed. There are several different updating formulas for H_k known and the two mostly used (see [2] or [10]) will be discussed here.

First a few definitions:

H_k is the approximation of G^{-1} after k line optimizations.

$\underline{\delta}_k := \underline{x}_{k+1} - \underline{x}_k$, and

$\underline{\gamma}_k := g_{k+1} - g_k$

The Taylor series gives:

$$\underline{\gamma}_k = G_k \cdot \underline{\delta}_k + \ldots\ldots \qquad (3.4.7)$$

If we restrict ourselves to quadratic functions, then the higher order terms of (3.4.7) can be neglected. Since $\underline{\delta}_k$ and $\underline{\gamma}_k$ are only known after the line search that gives $\underline{x}_{k+1}$ (we need H_k for determining the direction of the line search), it is difficult for H_k to satisfy $H_k \cdot \underline{\gamma}_k = \underline{\delta}_k$. Thus we want H_{k+1} to correctly relate $\underline{\gamma}_k$ and $\underline{\delta}_k$, i.e.

$$H_{k+1} \cdot \underline{\gamma}_k = \underline{\delta}_k \qquad (3.4.8)$$

This is sometimes called the quasi Newton condition.

The general formula for updating H is:

$$H_{k+1} = H_k + M_k \qquad (3.4.9)$$

A quasi Newton method is generally classified by the rank of the M_k-matrix. The methods, we will discuss are rank 2 methods. Let $\underline{u}$ and $\underline{v}$ be column vectors of length n, then a rank 2 method can be written as:

$$H_{k+1} = H_k + a\underline{u}\underline{u}^T + b\underline{v}\underline{v}T \qquad (3.4.10)$$

Still the quasi Newton condition (3.4.8) has to be satisfied:

$$\underline{\delta}_k = H_k\underline{\gamma}_k + a\underline{u}\underline{u}^T\underline{\gamma}_k + b\underline{v}\underline{v}^T\underline{\gamma}_k \qquad (3.4.11)$$

Since $\underline{u}$ and $\underline{v}$ are not uniquely determined by this equation, we are more or less free to pick something. An obvious choice is to choose $\underline{u}=\underline{\delta}_k$ and $\underline{v}=H_k\underline{\gamma}_k$ then a and b are determined by $a\underline{u}^T\underline{\gamma}_k=1$ and $b\underline{v}^T\underline{\gamma}_k=-1$. Realize that H_k is symmetric, then we have:

$$H_{k+1} = H_k + \frac{\underline{\delta}_k\underline{\delta}_k^T}{\underline{\delta}_k^T\underline{\gamma}_k} - \frac{H_k\underline{\gamma}_k\underline{\gamma}_k^T H_k}{\underline{\gamma}_k^T H_k\underline{\gamma}_k} \tag{3.4.12}$$

This updating formula is known as the DFP-formula (Davidon, Fletcher, Powell). If we define $B_k=H_k^{-1}$ the corresponding updating formula for B_{k+1} is given by:

$$B_{k+1} = B_k + \frac{\underline{\gamma}_k\underline{\gamma}_k^T}{\underline{\gamma}_k^T\underline{\delta}_k} - \frac{B_k\underline{\delta}_k\underline{\delta}_k^T B_k}{\underline{\delta}_k^T B_k\underline{\delta}_k} + \underline{w}_k\underline{w}_k^T$$

$$\text{with} \quad \underline{w}_k := (\underline{\delta}_k^T B_k\underline{\delta}_k)^{1/2}\cdot\left(\frac{\underline{\gamma}_k}{\underline{\gamma}_k^T\underline{\delta}_k} - \frac{B_k\underline{\delta}_k}{\underline{\delta}_k^T B_k\underline{\delta}_k}\right) \tag{3.4.13}$$

And the quasi Newton condition (3.4.8) can be rewritten into

$$B_{k+1}\underline{\delta}_k=\underline{\gamma}_k \tag{3.4.14}$$

If we now exchange $\underline{\delta}_k$ with $\underline{\gamma}_k$, and H_k with B_k then (3.4.13) changes into:

$$H_{k+1} = H_k + \frac{\underline{\delta}_k\underline{\delta}_k^T}{\underline{\delta}_k^T\underline{\gamma}_k} - \frac{H_k\underline{\gamma}_k\underline{\gamma}_k^T H_k}{\underline{\gamma}_k^T H_k\underline{\gamma}_k} + \underline{w}_k\underline{w}_k^T$$

$$\text{with} \quad \underline{w}_k := (\underline{\gamma}_k^T H_k\underline{\gamma}_k)^{1/2}\cdot\left(\frac{\underline{\delta}_k}{\underline{\delta}_k^T\underline{\gamma}_k} - \frac{H_k\underline{\gamma}_k}{\underline{\gamma}_k^T H_k\underline{\gamma}_k}\right) \tag{3.4.15}$$

Applying this exchange in (3.4.14) we see that (3.4.15) satisfies the quasi Newton condition. (3.4.15) is usually called the BFGS-formula (Broyden, Fletcher, Goldfarb, Shanno). Due to its relationship with (3.4.12) it is also called the complementary or dual DFP-formula. Because of the duality the DFP-formula can be derived from the BFGS-formula in a similar way as described above.

Now we have two possibilities for updating H, and the problem which one to choose. From tests that have been done with these two expressions (see [2]), it is known, that for accurate line searches they have about the same performance. However for inaccurate searches BFGS has proved to be much better than DFP. In fact DFP gets convergence problems if the line searches are not accurate enough, while the performance of BFGS hardly decreases. Therefore BFGS is a suitable method for the optimization part of the algorithm, the more so as it has proved to be good in comparison with other known methods, which are not mentioned here.

3.4.5 The line minimization algorithm

After the determination of $\underline{d}_k$ and α_{max} the algorithm is now to find a value $\hat{\alpha}$, $0 \leq \hat{\alpha} \leq \alpha_{max}$, such that,

$$f(\underline{x}_k + \hat{\alpha}\, \underline{d}_k) = \min_{\alpha} f(\underline{x}_k + \alpha\, \underline{d}_k) \qquad (3.4.16)$$

At the start of the line minimization $f(\underline{x}_k)$ is known. Similar to the Newton method i.e. $\underline{x}_{k+1} - \underline{x}_k = -H_k \underline{g}_k$, the value $\alpha=1$ is a good estimate for the first step.

If $f(\underline{x}_k) \leq f(\underline{x}_k+\underline{d}_k)$, then the value of α is doubled and again f is evaluated. The process of doubling the stepsize is repeated until either $\alpha \geq \alpha_{max}$ or $f(\underline{x}_k+\alpha\,\underline{d}_k) < f(\underline{x}_k+2\,\alpha\,\underline{d}_k)$. If $\alpha \geq a_{max}$ then α is made equal to α_{max}. If $f(\underline{x}_k+\alpha_{max}\underline{d}_k)$ still is less than or equal to the previous evaluated function value, then the minimum is assumed to be on the border and the corresponding constraint has to be made active.

If $f(\underline{x}_k+\alpha\,\underline{d}_k) < f(\underline{x}_k+2\,\alpha\,\underline{d}_k)$, then three value $\alpha_1 < \alpha_2 < \alpha_3$ are known satisfying:

$$f(\underline{x}_k+\alpha_2\underline{d}_k) \leq \min\,\{f(\underline{x}_k+\alpha_1\underline{d}_k), f(\underline{x}_k+\alpha_3\underline{d}_k)\} \qquad (3.4.17)$$

In fig. 3.2 this situation is plotted.

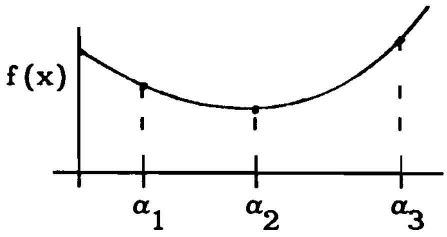


fig. 3.2 Closing in the minimum

So an interval is known that contains the line minimum and we will now try to reduce this interval. This is done by means of quadratic interpolation, i.e. a parabola is constructed through the three known points and its minimum is computed using:

$$\tilde{\alpha} = \frac{(\alpha_1+\alpha_2)\cdot mc + (\alpha_2+\alpha_3)\cdot ma}{2\cdot(ma+mc)}$$

$$\text{with} \quad ma = (\alpha_2-\alpha_3)\cdot(h(\alpha_1)-h(\alpha_2)),$$

$$mc = (\alpha_1-\alpha_2)\cdot(h(\alpha_3)-h(\alpha_2)),$$

$$\text{and} \quad h(\alpha)=f(\underline{x}_k+\alpha\underline{d}_k) \qquad (3.4.18)$$

One of the exterior points α_1 or α_3 will now be discarded, such that the three remaining points satisfy the condition (3.4.17), thus reducing the interval. This process is repeated several times, and the smallest functionvalue discovered is assumed to be the minimum.

In case of $f(\underline{x}_k) < f(\underline{x}_k + \underline{d}_k)$ the stepsize is halved and this repeated until three values are found satisfying (3.4.17), or until the step is smaller than some user specified minimum value. In the first case the parabolic interpolation procedure can be started, in the second case a constrained minimum is assumed to be found.

3.4.6 Dealing with constraints

As described the general way of dealing with constraints is to ignore them when they are passive and to keep them active, once they have become active. This implies that active constraints influence the choice of a $\underline{d}_k$. Several methods have been developed for calculating a $\underline{d}_k$ satisfying the active constraints. Out of these methods, Goldfarb's method [10] has been chosen, though some adaptions have been made. Goldfarb's method makes a $\underline{d}_k$ satisfying its constraints by modifying H_k. It distinguishes 3 cases:

1. There has been no change in the constraint set. In that case the BFGS-formula (3.4.15) is used to calculate H_{k+1} from H_k.
2. One of the passive constraints has become active e.g. $x_i=0$, so now the number of variables in the function is reduced by one, and thus we have to lower the rank of H_k by 1. This is done by filling the

i-th row and the i-th column of H_k with zeros, thus getting H_{k+1}. Now $H_{k+1}\underline{g}_{k+1}=\underline{d}_{k+1}$ will have a zero on the i-th position.

3. If one of the active constraints is allowed to become passive e.g. $x_i=0$, then the element h_{ii} of H (which was zero) will be changed into a 1, thus allowing $\underline{d}_k$ to have a non-zero component in direction i. Sometimes however it may happen, that $\partial f/\partial x_i$ is infinite, and numerical problems could arise when updating H. Therefore H is re-initialized and the search is continued.

3.5 The gradient evaluation

3.5.1 Introduction

The optimization method discussed in the previous chapter needed information about the gradient. The computation of this vector can be done either numerically or analytically. Because of the possibility of a partial derivative to become infinite, the numerical determination might get into troubles. Therefore an analytical solution is preferable, despite of the fact that it is a hard problem to solve.

3.5.2 Inventarisation of the problem

Because we defined $R(\underline{x}) := R_{12}(\underline{x}) + R_{21}(\underline{x})$, for a partial derivative:

$$\frac{\partial R(\underline{x})}{\partial x_i} = \frac{\partial R_{12}(\underline{x})}{\partial x_i} + \frac{\partial R_{21}(\underline{x})}{\partial x_i} \qquad (3.5.1)$$

So it is possible to consider R_{12} and R_{21} separately, when calculating $\partial R/\partial x_i$. Because the method of computation for the two gradients is very similar, only the evaluation of R_{12} gradient will be discussed here. As seen in section 3.3, we can make sets S_i where each set contains elements of A that are characterized by the same $\underline{x}_2, \underline{y}_2$ vectors. Let A_i and L_i denote the sum of resp. the areas and the lengths of the elements of such a set S_i, then the rate can be written as:

$$R_{12} = \sum_{\text{all } S_i} - A_i \cdot \log L_i \qquad (3.5.2)$$

But $A_i = H_i \cdot L_i$ (H_i is the height of that group of rectangles.) substituted in (3.5.2) gives:

$$R_{12} = \sum_{\text{all } S_i} -H_i \cdot L_i \cdot \log L_i \qquad (3.5.3)$$

The height of the rectangles is determined by the values of the horizontal thresholds they are surrounded by. The two vertical thresholds determine the length. We will call the horizontal thresholds 'linear', because if R_{12} was regarded as a function of a horizontal threshold x_i, $R_{12}(x_i)$ would be a function of first degree. The vertical thresholds appear as 'a log a' in the expression for $R_{12}(\underline{x})$, therefore they will be called 'entropy thresholds'. Since linear thresholds and entropy thresholds contribute to $R_{12}(\underline{x})$ in a different way, for each case a different method is necessary.

3.5.3 The coordinate transform

In section 3.3.3 we saw, that before $R_{12}(\underline{x})$ was really calculated, the thresholds were subjected to a coordinate transform, which did not only change the values of x_i's, but also related values of thresholds to one another. So a value of x_i influences the values of several other thresholds. Therefore we also have to consider the effect of the coordinate transform on the partial derivatives. First the even thresholds are discussed. If a threshold x_{2k} had to satisfy the constraint $x_a \leq x_{2k} \leq x_b$, then it was passed through an expression that projected it from the [0,1] interval onto the interval $[x_a,x_b]$. Suppose x_{2k} is a threshold of the n-th transmission stage. Then there are groups of thresholds related to the following stages (>n) that directly or indirectly have to satisfy $x_a \leq x \leq x_{2k}$ (call this a left threshold) or $x_{2k} \leq x \leq x_b$ (right threshold). Let I_i be an interval determined by left thresholds, then the length of $I_i =: |I_i|$ is a linear function of x_{2k}. The length of an interval J_j, determined by right thresholds is a linear function of $(1-x_{2k})$.

Summarizing it can be said, that during the coordinate transform x_{2k} does not affect values of thresholds related to previous stages. x_{2k} is projected from the [0,1] interval onto the interval $[x_a,x_b]$ (x_a and x_b are thresholds of previous stages, and they have already been passed through the transform). Finally the following holds:

$$\sum_i |I_i| = x_{2k}\cdot(x_b-x_a)$$
$$\sum_j |J_j| = (1-x_{2k})\cdot(x_b-x_a) \qquad (3.5.4)$$

The only difference for an odd threshold is, that it satisfies another right constraint i.e. $x_a \leq x_{2k-1} \leq x_{2k}$, but its influence on thresholds of following stages is very similar to that of its even neighbour.

3.5.4 Partial derivatives to an entropy threshold

Let x_{2k} be an entropy threshold, whose partial derivative we are interested in. Remember the data structure, that mainly consisted of the matrix A. According to the reasoning in the previous section the elements of A, that do not lie in the area between the thresholds x_a and x_b, do not have any relationship with x_{2k}. Consider a row of elements between x_a and x_b, whose values are influenced by x_{2k}. In section 3.3 we discussed, how the rate was computed. For the R_{12} situation all elements of A within the same row and having the same $\underline{y}_2$ output were regarded. Their areas and lengths were added and these were substituted in the formula for $R_{12}(\underline{x})$. If we consider these elements with respect to their relationship with x_{2k}, then we can distinguish 3 cases:

1. Elements that are outside the $[x_a, x_b]$ interval. So their lengths and areas are not affected by the value of x_{2k}.
2. Elements within the $[x_a, x_b]$ interval, to the left of x_{2k}. Their measurements are a linear function of x_{2k}.
3. elements within the $[x_a, x_b]$ interval, but to the right of x_{2k}. Their lengths and areas are a linear function of $(1-x_{2k})$.

Call the sum of the areas and lengths of the elements of group 1 resp. $A^{(1)}$ and $L^{(1)}$, the sum of the areas and the lengths of the elements of group 2 resp. $x_{2k}A^{(2)}$ and $x_{2k}L^{(2)}$ ($A^{(2)}$ and $L^{(2)}$ do not depend on x_{2k}) and the sum of the areas and the lengths of the elements of group 3 resp $(1-x_{2k})\cdot A^{(3)}$ and $(1-x_{2k})\cdot L^{(3)}$ ($A^{(3)}$ and $L^{(3)}$ also do not depend on x_{2k}). From section 3.3 we know that the contribution to R_{12} is given by:

$$-(A^{(1)}+x_{2k}A^{(2)}+(1-x_{2k})\cdot A^{(3)}) \cdot \log (L^{(1)}+x_{2k}L^{(2)}+(1-x_{2k})\cdot L^{(3)}) \qquad (3.5.5)$$

Since the discussed elements are characterized by the same $\underline{x}_2$ and $\underline{y}_2$ vectors, they cannot be distinguished by T_2 i.e. these elements are surrounded by the same horizontal thresholds, so their height is the same, say H. So

$$A^{(i)} = H\cdot L^{(i)} \qquad (i = 1,2,3) \qquad (3.5.6)$$

Substituting this in (3.5.5) we get:

$$-H*(L^{(1)}+x_{2k}L^{(2)}+(1-x_{2k})\cdot L^{(3)}) * \log (L^{(1)}+x_{2k}L^{(2)}+(1-x_{2k})\cdot L^{(3)}) \qquad (3.5.7)$$

Differentiating this expression to x_{2k} gives:

$$-H\cdot(L^{(2)}-L^{(3)})-H\cdot(L^{(2)}-L^{(3)})*\log (L^{(1)}+x_{2k}L^{(2)}+(1-x_{2k})\cdot L^{(3)}) \qquad (3.5.8)$$

Observe that the sum of all the areas influenced by x_{2k} does not depend on x_{2k}, so it can be considered as a constant. Let i only have the values of the S_i's with elements in the descibed area.

$$C := \sum_i x_{2k}A_i^{(2)} +(1-x_{2k})\cdot A_i^{(3)} \qquad (3.5.9)$$

Differentiating to x_{2k} gives:

$$\sum_i A_i^{(2)}-A_i^{(3)} = 0 \qquad (3.5.10)$$

Formula (3.5.8) has to be added for all possible $\underline{x}_2,\underline{y}_2$ combinations, thus giving the value of the partial derivative. So:

$$\frac{\partial R_{12}}{\partial x_{2k}} = \sum_i \{-H_i\cdot(L_i^{(2)}-L_i^{(3)}) - H_i\cdot(L_i^{(2)}-L_i^{(3)})\cdot\log\,(L_i^{(1)}+x_{2k}L_i^{(2)}+(1-x_{2k})L_i^{(3)})\}$$

$$(3.5.11)$$

With (3.5.6) and (3.5.10) we get

$$\frac{\partial R_{12}}{\partial x_{2k}} = \sum_i -H_i\cdot(L_i^{(2)}-L_i^{(3)})\cdot\log\,(L_i^{(1)}+x_{2k}L_i^{(2)}+(1-x_{2k})L_i^{(3)})$$

$$(3.5.12)$$

The possibility of (3.5.12) to become infinite is easily seen: If for some i the log argument is zero then (3.5.12) $\partial R_{12}/\partial x_{2k}$ is positive infinite when $x_{2k}=L_{i1}=L_{i3}=0$ and $L_{i2}> 0$. $\partial R_{12}/\partial x_{2k}$ is negative infinite when $x_{2k}=1$, $L_{i1}=L_{i2}=0$ and $L_{i3}> 0$.

For odd thresholds the solution is very similar, but the right boundary is now x_{2k} instead of x_b, assuming we differentiate to x_{2k-1}.

3.5.5 Partial derivative to a linear threshold

Just as in the previous section, we will first consider an even threshold, x_{2k} and again we observe that the elements of A, influenced by x_{2k}, are situated between two thresholds of previous transmission stages: x_a on top and x_b at the bottom. Also remember that if two elements are characterized by the same $\underline{x}_2,\underline{y}_2$ they are surrounded by the same linear thresholds. Analogous to the previous section we can denote the height of an element between x_a and x_{2k} by $x_{2k}H^{(1)}$, with $H^{(1)}$ independent of x_{2k}. The height of an element between X_{2k} and x_b can thus be denoted by $(1-x_{2k})\cdot H^{(2)}$ ($H^{(2)}$ independent of x_{2k}). If we call the sum of the lengths of all the elements characterized by the same $\underline{x}_2,\underline{y}_2$: L, then we can denote a term for R_{12} concerning elements above x_{2k} by:

$$-x_{2k}H_i^{(1)}L_i\cdot\log L_i \qquad (3.5.13a)$$

For the elements below x_{2k} we have:

$$-(1-x_{2k})\cdot H_j^{(2)}L_j\cdot\log L_j \qquad (3.5.13b)$$

Differentiating these terms to x_{2k} gives:

$$-H_i^{(1)} L_i \cdot \log L_i \qquad (3.5.14a)$$

and

$$+H_j^{(2)} L_j \cdot \log L_j \qquad (3.5.14b)$$

Adding these expressions for all elements concerned

$$\frac{\partial R_{12}}{\partial x_{2k}} = \sum_i -H_i^{(1)} L_i \cdot \log L_i \ - \ \sum_j -H_j^{(2)} L_j \cdot \log L_j \qquad (3.5.15)$$

Again for odd thresholds the problem is very similar, except for the fact that the interval now is restricted to $[x_a, x_{2k}]$.

Chapter 4. Shannonstrategies for two tdTWC's

4.1 Introduction

With the help of the algorithms described in the previous chapter, we are able to calculate good Shannon-strategies for any tdTWC. Because of the exponential growth of the number of thresholds by an increasing number of transmissions, we limit ourselves to K_3-strategies. Such strategies have been computed for two tdTWC's; these are $S_{3.3}$ and $S^b_{10.10}$ following the naming of [4]. Fig. 4.1 shows the partition patterns of these two channels.

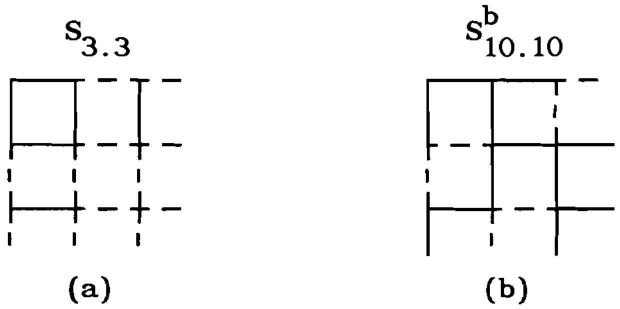


fig. 4.1 Partition patterns of $S_{3.3}$ and $S^b_{10.10}$

The reasons, why these two channels are chosen, are:

$S_{3.3}$ is chosen, because it is a non T-channel and since all the strategies designed until now concern only T-channels or channels which are equivalent to T-channels (see [4]), it is interesting to see how a strategy for a non T-channel works. The second reason is, that there is a great difference between the Shannon inner and outerbounds of this channel. The third reason is, that it is a true ternary dTWC (R_{12} and

R_{21} separately can achieve 1 trit/transmission), and it has the least number of solid lines in its partition pattern, compared to other true tdTWC's. $S^b_{10.10}$ has been chosen, because of its peculiar Shannonbounds (See [4] app. B) and because it seems that it will divide in a very symmetric way, since the partition pattern consists of three L-shapes, and each L-shape again is built up from 3 square elements.

In the next sections it is shown, how the computed strategies divide the square. App. B contains the computed values of the thresholds.
The strategies, though achieving a far better rate than the Shannon innerbound, cannot be claimed to be optimum, because of the many local optima the rate-function has. This makes it nearly impossible to prove that the best strategy has been found, at least, when it is calculated using a computer program.
The strategies plotted here are the best found after several searches have been performed, and each search is started with random thresholdsettings i.e. each threshold is randomly chosen from the [0,1) interval.

4.2 The K_3-strategy for $S_{3.3}$

Given a strategy that yields rate (R_{12},R_{21}), the total rate of the strategy R_{tot} is defined by:

$$R_{tot} := \frac{1}{2} (R_{12} + R_{21}) \qquad (4.2.1)$$

Table 4.1 shows the input distribution that yields innerboundrate for this tdTWC.

x_1	$P(x_1)$	x_2	$P(x_2)$
0	0.63870	0	0.63870
1	0.17165	1	0.17165
2	0.17165	2	0.17165

$R_{21} = 0.52672$ $R_{12} = 0.52672$

$R_{tot} = 0.52672$

table 4.1 Input probabilities for innerbound rate of $S_{3.3}$

In fig. 4.2 the square division of the first transmission is plotted (solid lines; the dashed lines correspond to the dashed lines in the partition pattern). It is seen that the square is divided into a rectangle and a shape, that will be called the folded 'cross shape'. In the rectangle the partitions for the second transmission are also drawn (dash-dot-lines), and since the rectangle of this second transmission is divided by an innerbound division in the third transmission, it is marked with 'IB'.

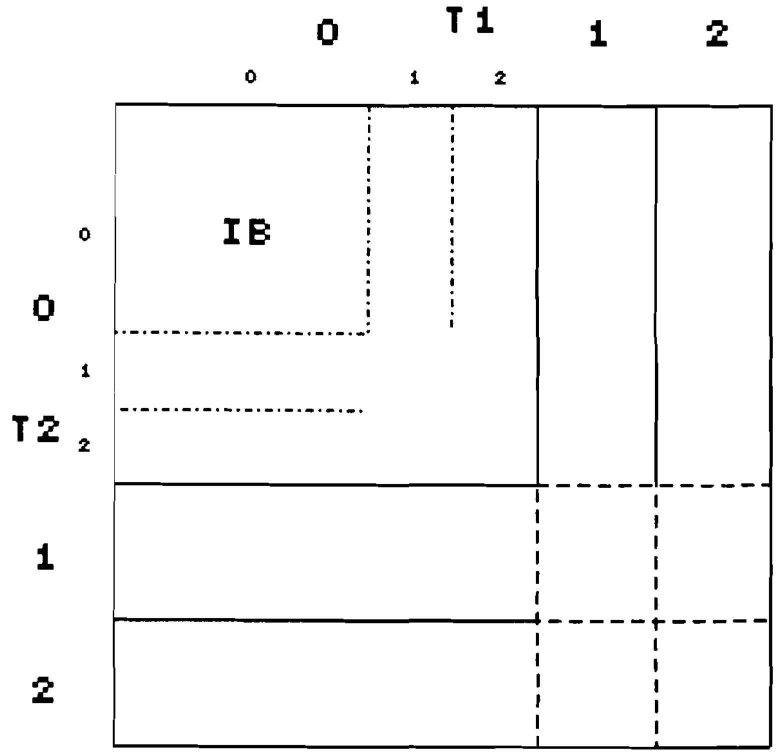


Fig. 4.2 The division of the unit square by $S_{3.3}$

We will now consider the folded cross shape that remained in the rectangle after two transmissions. In fig. 4.3 this shape is unfolded and it is seen, how this shape is divided in the third transmision (the dash-dot-lines show this division; the dashed lines correspond to the dashed lines in the partition pattern). It is very interesting to see, that only rectangular shapes remain. thus we have found an indication for a Markov strategy for this channel, which consists of only two states. These two states are:

1. The division of a rectangle into a rectangle and a folded cross shape.
2. The division of the cross shape in just rectangles in a way similar to the way it is done in fig. 4.3.

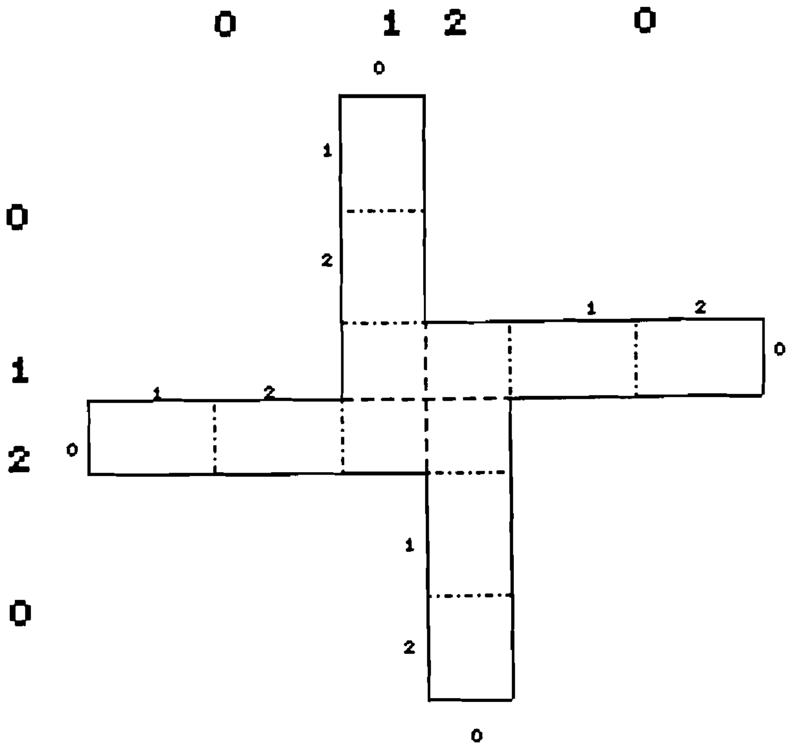


fig. 4.3 The division of the cross shape of the second transmission

We now have to study the division of the cross shape, that remained after the first transmission. In fig. 4.4 it can be seen, that except for the left arm, it looks very similar to the division shown in fig. 4.3 (dash-dot-lines correspond to the second transmission; dashed lines correspond to dashed lines in the partition pattern). Again the rectangles, that will be divided with an innerbound transmission in the third transmission are marked with IB. The rectangles marked with IB' will be divided in a way very close to the innerbound (the difference appears in the third significant decimal; all IB' divisions are identical). Notice that these rectangles are attached to a part of the left arm by the 'stroke' at the bottom of the square in the center.

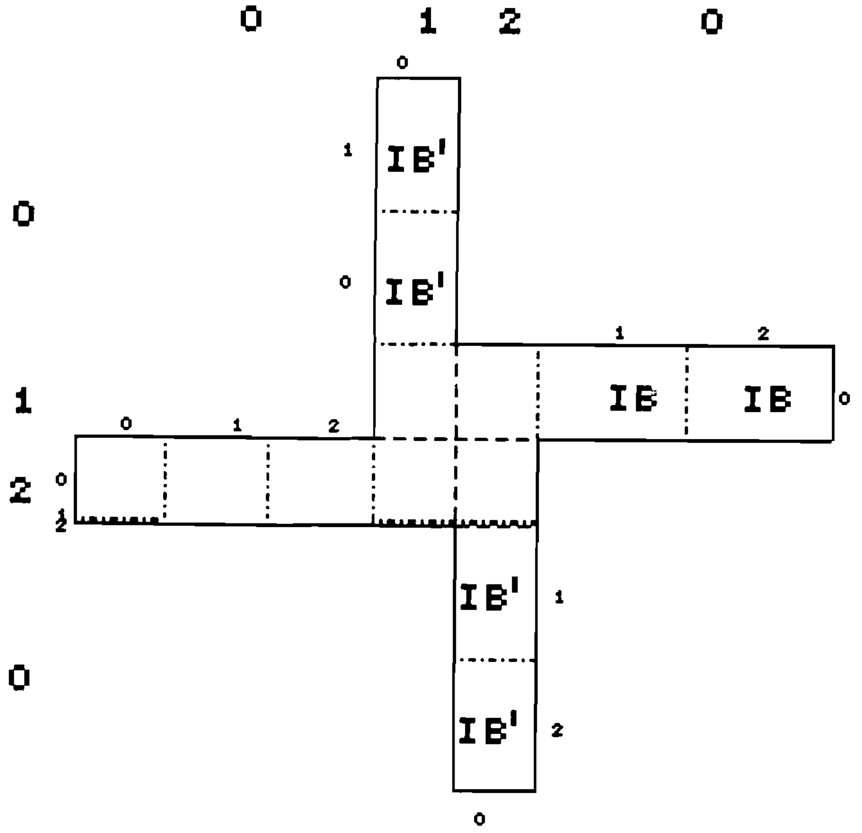


fig 4.4 Division of the cross shape of the first transmission

We now only have to consider two subshapes; these are the center square (except the small bottom area), that is connected to the '00' input square in the left part of the left arm. This is shown in fig.4.5. The other part is the remaining area of the left arm in connection with the 'stroke' of the center square and the rectangles marked with IB'. Fig. 4.6 shows this division, but the IB' rectangles have been omitted in order to get a better intelligible picture. Therefore the line at the right bottom is dashed. In both pictures the actual division is drawn with the dash-dot-line, divisions of previous transmissions are drawn with solid lines, and the dashed lines correspond to the dashed lines in the partition pattern and they also are a result of previous transmissions.

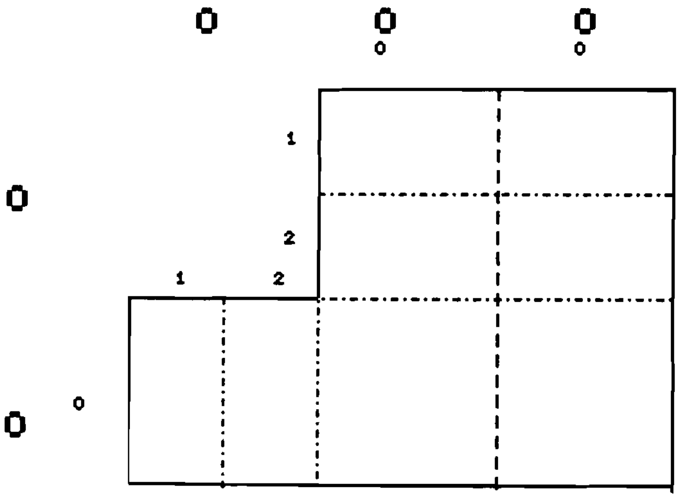


fig. 4.5 Division of the the centersquare with the '00' area

Notice, that the area of fig. 4.5 is entirely divided into rectangles.

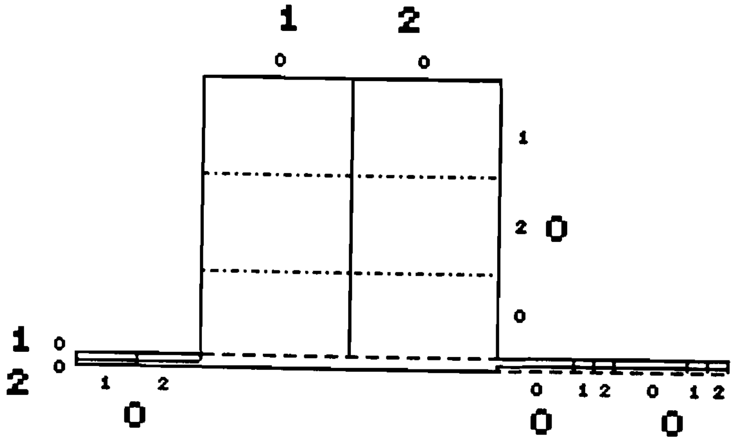


fig. 4.6 Division of the shape in the left arm of fig 4.4

The rates of the shown strategy are:

R_{12} = 0.52036 trits/transm.

R_{21} = 0.58406 trits/transm.

R_{tot}= 0.55221 trits/transm.

Comparing this to the innerbound (R_{tot} = 0.52672), we have an improvement not to be neglected, though it is still far below the outerbound (R_{tot} = 0.63093).

4.3 The K_3-strategy for $S^b_{10.10}$

The innerbound rate of $S^b_{10.10}$ is 0.57938 and can be achieved when all probabilities of the input symbols of both terminals are equal to 1/3. Fig. 4.7 shows the first two transmissions of the K_3-strategy. It is seen, that in the first transmission the square is divided in three almost congruent L-shapes (see the solid lines; the dashed lines correspond to the dashed lines in the partition pattern; the dash-dot-lines are used for the second transmission). We will number these three L-shapes as follows: L-shape nr.1 corresponds to the upper left-, middle left- and middle right element of fig.4.7, L-shape nr. 2 corresponds to the center-, lower middle- and lower left element of 4.7; the remaining L-shape has nr. 3. In the figures 4.8, 4.9 and 4.10 it is seen how these three L-shapes are divided by this K_3-strategy (N.B. some inputs have been exchanged in order to have a better view on the coherent areas). The dash-dot-lines symbolize the division of the third transmission; the solid lines correspond to divisions by previous transmissions; concerning the dashed lines of the partition pattern, the dashed lines in the figures correspond to the first transmission and the dotted lines correspond to the second transmission. The regions marked with 'IB' have an innerbound division in the third transmission. Because of the fact, that the L-shape in fig. 4.9 is divided in a way different from the other two L-shapes (these two L-shapes are divided in exactly the same way), it is reasonable to assume that this is not an optimum. It could be that both divisions yield the same rate, but this is very unlikely. A better rate could be achieved by looking for the best way to divide the L-shape, and then to divide all three L-shapes equally in this way.

In any case the rates of this strategy are:

R_{12} = 0.59073 trits/transm.

R_{21} = 0.59046 trits/transm.

R_{tot}= 0.59059 trits/transm.

This also is a rather good result in comparison with the innerbound rate, but again not very close to the outerbound (R_{tot} = 0.63093 trits/transm.). This strategy does not result in a Markov strategy at once, but because of the fact that after three transmissions the square is divided only into rectangles and L-shapes finding such a strategy might not be too difficult.

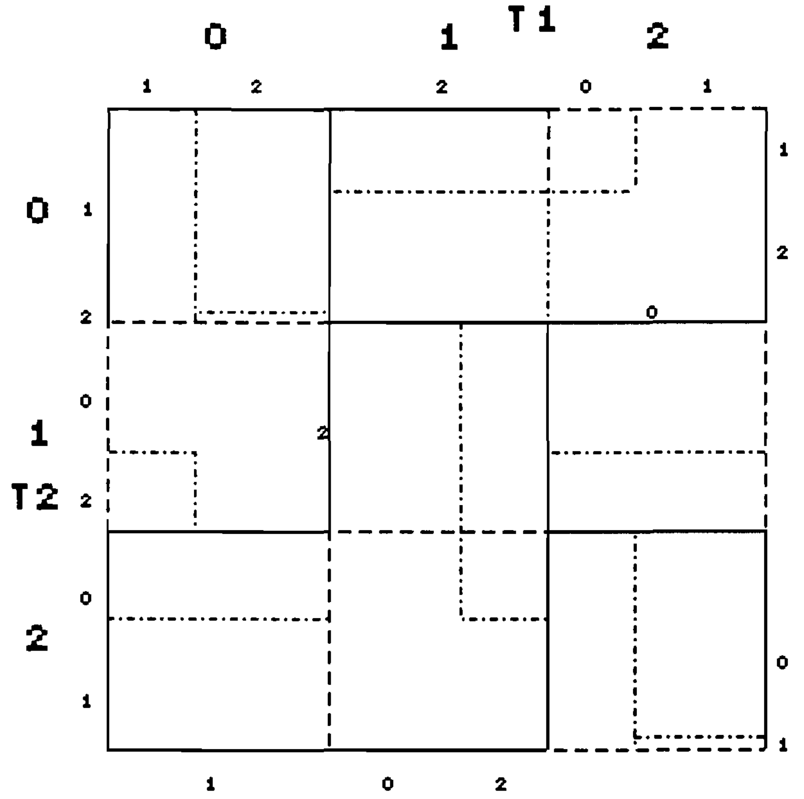


Fig. 4.7 First 2 transmissions of the K_3-strategy for $S^b_{10.10}$

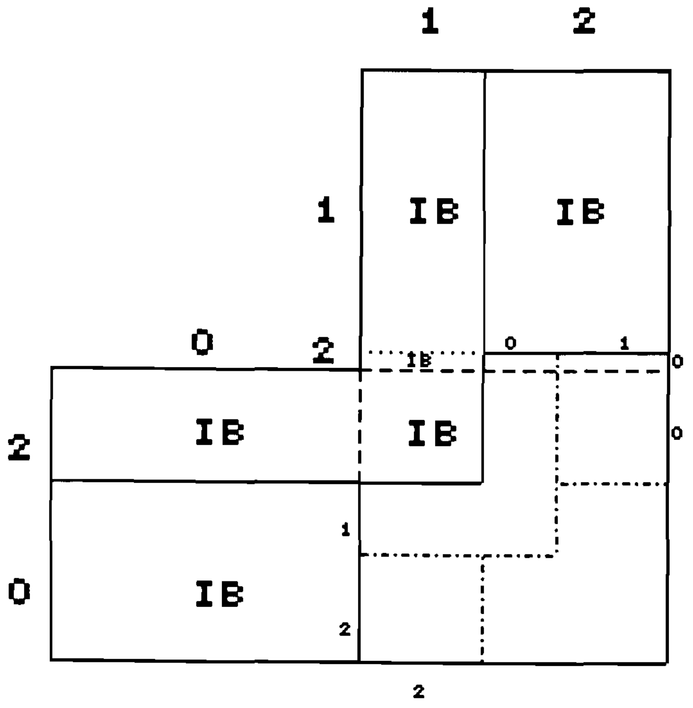


fig. 4.8 Division of L-shape nr.1

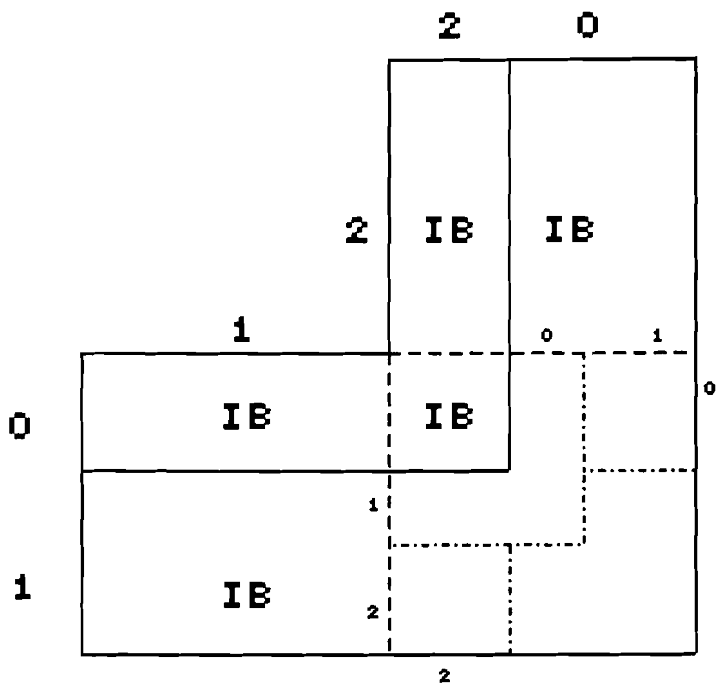


fig. 4.9 Division of L-shape nr. 2

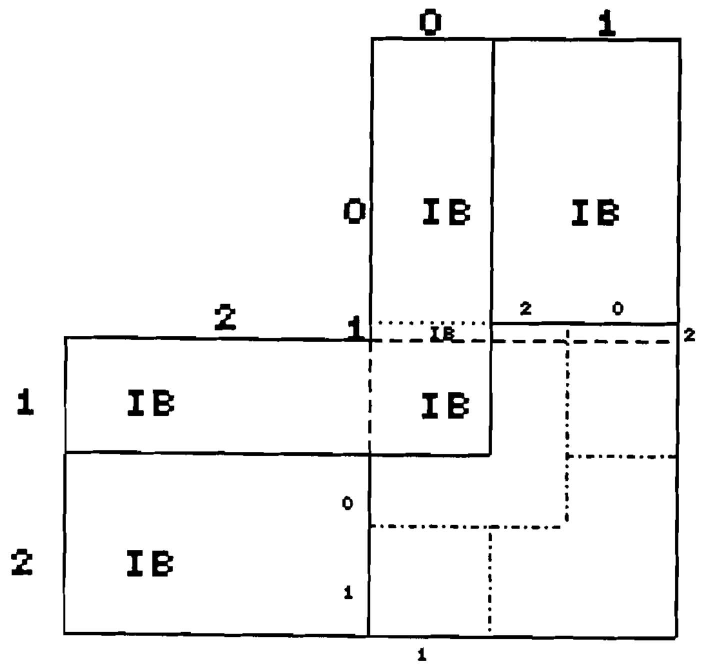


fig. 4.10 Division of L-shape nr. 3

Chapter 5. Dominating part-patterns and dominated partition patterns

5.1 Introduction

In this chapter we will study the so-called dominating part-patterns, because we will see, that with the help of these patterns the problem of finding good strategies can be reduced substantially. Part-patterns of a tdTWC are the partition patterns that remain when one or more of the input symbols are not used. A part-pattern that consists of m rows x n columns will be denoted as an [m,n] part-pattern. An [m,n] part-pattern corresponds to a dTWC with n and m input symbols respectively; when discussing Shannonbounds or capacity regions of part-patterns, the Shannonbounds or capacity regions of the corresponding channels are meant. Fig. 5.1 shows a tdTWC with all its [2,2], [3,2] and [2,3] part-patterns.

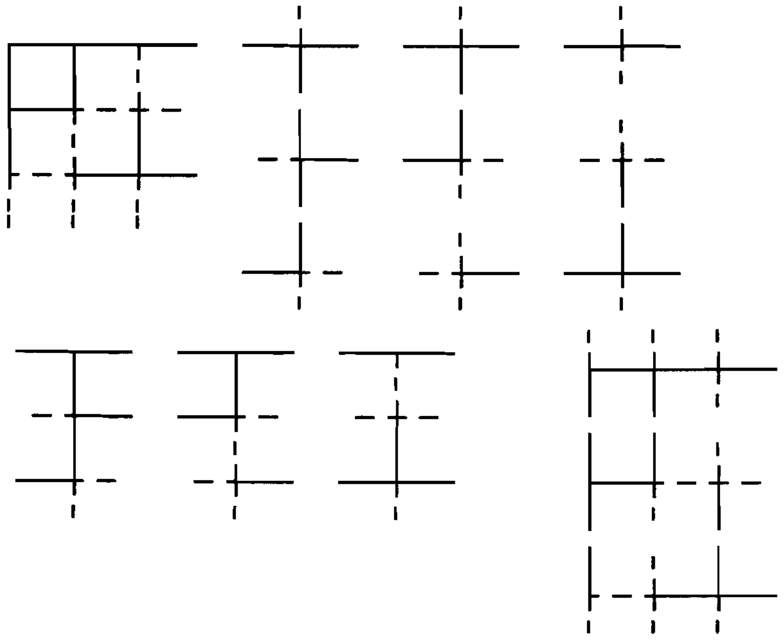


fig. 5.1 A tdTWC with all its [2,2]-, [2,3]-, [3,2]-part-patterns

In [4] Jacobs discovered that some tdTWC's have capacity regions or Shannonbounds, which are equal to the capacity regions or Shannonbounds of one of their part-patterns. She introduced the concept of dominating part-patterns. A part-pattern is said to be a dominating part-pattern, if its capacity region entirely contains the capacity regions of all the other part-patterns of the tdTWC concerned. Engels [1] made a more elaborate study of this phenomenon. He extended the definition of dominating part-patterns to a more useful one by considering Shannonbounds, when capacity regions are not known. He called a pattern dominating, if:

1. its capacity region contains the capacity regions of all other part-patterns.

or:

2. The Shannon innerbound contains the innerbounds and the Shannon outerbound contains the outerbounds of all other part-patterns if the capacity region is not known.

Using this definition he partially ordered all the part-patterns w.r.t. the part-patterns they dominate and the part-patterns they are dominated by. With the help of this list he checked all the partition patterns on having dominating part-patterns. He found that there were partition patterns that had the Shannon innerbound of a dominating part-pattern, but had a different outerbound, and that there were partition patterns with a dominating part-pattern but unique Shannon inner- and outerbounds, i.e. not the ones of the dominating part-pattern. In this chapter we will concentrate on tdTWC's having

Shannonbounds (or capacity regions) of one of their part-patterns. These tdTWC's will be called dominated tdTWC's and with 168 of the 308 partition patterns (the channels $S_{1.1}\ldots S_{1.14}$ of [4] are not regarded) they form a very important class. It will be shown that there is great regularity among those patterns.

5.2. Out rules for dominated tdTWC's

Some rules will now be given, that when applied to a tdTWC tell which inputs do not have to be used. With their help it is possible to see almost at once whether a channel is dominated, and by what part-pattern it is dominated.

rule 1a (see also [4])

Requirement: 1. One of the part-patterns is the pattern shown in fig. 5.2a

2. Every row and column of the partition pattern contains at most two solid lines.

The capacity region is:

$$G_{1a}= \{ (R_{12},R_{21}) \in \mathbb{R}^2 \mid 0 \leq R_{12} \leq {}^3\log 2 \text{ , } 0 \leq R_{21} \leq {}^3\log 2 \}$$

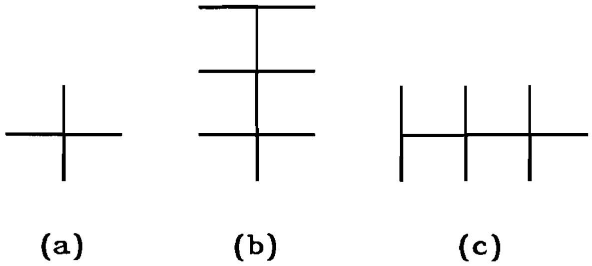


fig. 5.2 Part patterns, required for rule 1

Rule 1b (1c)

Requirement: 1. One of the part-patterns is the [3,2]-([2,3]-) pattern shown in fig. 5.2b (5.2c).

2. Every row (column) contains at most two solid lines.

The capacity regions are given by:

$$G_{1b} = \{ (R_{12},R_{21}) \in \mathbb{R}^2 \mid 0 \leq R_{12} \leq {}^3\log 2 \text{ , } 0 \leq R_{21} \leq 1 \}$$
$$G_{1c} = \{ (R_{12},R_{21}) \in \mathbb{R}^2 \mid 0 \leq R_{21} \leq {}^3\log 2 \text{ , } 0 \leq R_{12} \leq 1 \}$$

Proof 1a:

Requirement 2 ⇒ At most G_{1a} is achievable.

Requirement 1 ⇒ G_{1a} can be achieved.

Therefore G_{1a} is the capacity region. □

Rule 1b and 1c can be proved in a similar way.

Rule 2: (see also [4])

Requirement: The partition pattern contains one of the part-patterns depicted in fig. 5.3 (or their transposed versions).

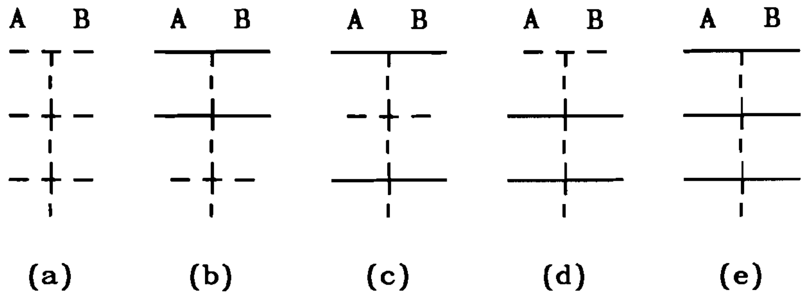


fig. 5.3 Part-patterns required in rule 2

Capacity region: The inputs A and B can be considered as one super input. So the capacity region is the region of the part-pattern that remains, when either one of the inputs A or B is not used.

Proof: Consider e.g. the part-pattern of fig. 5.3b, and connect an arbitrary column to it. Assume the $P(x_1x_2)$-distribution as shown in fig. 5.4. This distribution results in a rate pair, say (R_{12},R_{21}), and let R be defined by: $R := R_{12} + \lambda R_{21}$.

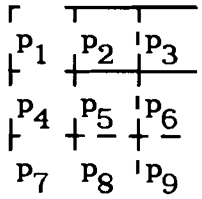


fig. 5.4 The defined input probability distribution.

We consider a second input probability distribution, constructed from

$P(x_1x_2)$ using the following relations:

$$\begin{array}{ll} p'_i:=p_i & i=1,4,7 \\ p'_i:=p_i+p_{i+1} & i=2,5,8 \\ p'_i:=0 & i=3,6,9 \end{array} \qquad (5.2.1)$$

I.e. the third column is discarded.

We define:

$$h(p) := -p\cdot\log p - (1-p)\cdot\log (1-p) \qquad (5.2.2)$$

Now R_{12} only consists of terms that are either zero or have the form $(p_i+p_{i+1}+p_{i+2})\cdot h(\frac{p_i}{p_i+p_{i+1}+p_{i+2}})$ (i=1,4,7). So the value of R_{12} is not changed by the transform (5.2.1), i.e. $R'_{12}=R_{12}$.

Using the convexity of h(p), we get for R_{21}:

$$\begin{aligned} R_{21}= & f(p_1,p_4,p_7)+(p_2+p_5+p_8)\cdot h(\frac{p_2}{p_2+p_5+p_8})+(p_3+p_6+p_9)\cdot h(\frac{p_3}{p_3+p_6+p_9}) \\ \leq & f(p'_1,p'_4,p'_7)+(p'_2+p'_5+p'_8)\cdot h(\frac{p'_2}{p'_2+p'_5+p'_8}) = R'_{21} \end{aligned}$$

with $f(p_1,p_4,p_7) = f(p'_1,p'_4,p'_7)$ is the contribution of the first column to R_{21}.

So $R = R_{12}+ \lambda R_{21} \leq R'_{12} + \lambda R'_{21} = R'$, which means that discarding a column does not yield a lower rate. This implies that all the points achieved using the third input can always be achieved, when this input is discarded.

A similar proof can be given for the other part-patterns mentioned.

□

Rule 3:

<u>Requirement</u>: One of the [3,2] part-patterns shown in fig. 5.5 (or the permuted or transposed versions of them) is contained in the partition pattern.

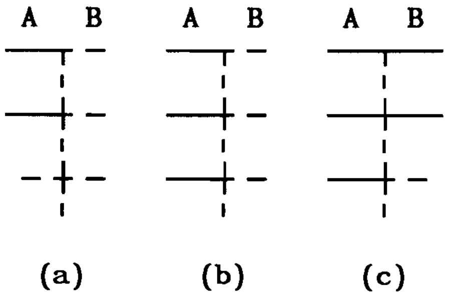


fig. 5.5 Part-patterns required in rule 3

Capacity region: The capacity region is the region of the pattern that remains when the column with the smaller number of partitions is discarded.

Proof: Consider the pattern of fig. 5.5c. Choose arbitrarily a third column and an input distribution, analogous to the proof of rule 2. We will use (5.2.1) again and define:

$$H(x,y,z) := -\frac{x}{x+y+z} \cdot \log\left(\frac{x}{x+y+z}\right) - \frac{y}{x+y+z} \cdot \log\left(\frac{y}{x+y+z}\right) - \frac{z}{x+y+z} \cdot \log\left(\frac{z}{x+y+z}\right) \qquad (5.2.3)$$

Again $R_{12}= R'_{12}$ is not affected by (5.2.1).

Remember that:

$$a \log a \ + \ b \log b \ \le \ (a+b) \log (a+b) \quad (a,b > 0) \quad (5.2.4)$$

Then we have for R_{21}:

$$R_{21}= f(p_1,p_4,p_7)+(p_2+p_5+p_8)\cdot H(p_2,p_5,p_8)+(p_3+p_6+p_9)\cdot h(\frac{p_3}{p_3+p_6+p_9})$$

(5.2.4)

$$\le f(p_1,p_4,p_7)+(p_2+p_5+p_8)\cdot H(p_2,p_5,p_8)+(p_3+p_6+p_9)\cdot H(p_3,p_6,p_9)$$

(convexity of H)

$$\le f(p'_1,p'_4,p'_7)+(p'_2+p'_5+p'_8)\cdot H(p'_2,p'_5,p'_8) = R'_{21}$$

The proof is finished by the same argument as the proof of rule 2, thus having proved that an input can be discarded without affecting the capacity region. Similar proofs can be given for the other patterns mentioned. □

Rule 4:

<u>Requirement</u>: The pattern shown in fig. 5.6 (or a permuted or transposed version of it) is one of the part-patterns.

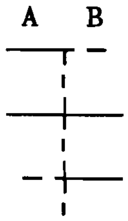


Fig. 5.6 Part-pattern required in rule 4

Shannonbounds: Compare the Shannonbounds of the part-pattern, that remains, when input A is not used (G_{iA},G_{oA}) to those corresponding to the part-pattern, that remains when input B is not used (G_{iB},G_{oB}).

Now there are three possibilities:

1. $G_{iA}=G_{iB}$ and $G_{oA}=G_{oB}$, then the partition pattern is dominated by the part pattern that remains, when either input A or B is not used.
2. $G_{iA}\subset G_{iB}$ and $G_{oA}\subset G_{oB}$. In this case the tdTWC is dominated by the pattern corresponding to $P(x_1=B) = 0$.
3. $G_{iB}\subset G_{iA}$ and $G_{iA}\subset G_{iB}$. Then the dominating part-pattern is the pattern corresponding to $P(x_1=A) = 0$.

Rule 5:

<u>Requirement</u>: Regard a column (row).

1. There are no solid lines in this column (row).
2. Every element in this column (row) has a neighbour element, outside this column (row), with no solid lines inbetween them (remember that this must be considered cyclically).

An example is given in fig.5.7.

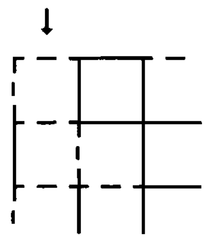


fig. 5.7 Example of rule 5

Shannonbounds: The Shannonbounds of the tdTWC are the same as the ones of the part-pattern that remains, when zero input probability is assigned to the column (row) described above.

Unfortunately a proof of rule 4 and 5 has not been found, but by checking all the tdTWC's no exceptions to these rules have been found.

These rules, however in an adapted form, can also be used to determine whether a [3,2]- or [2,3]-pattern is dominated by a [2,2]-pattern.In appendix A all dominated partition patterns are listed. It is seen there, that with the help of these 5 rules 145 of the 168 dominated partition patterns can be predicted, which shows the regularity of these channels.

5.3 In rules

Since the 5 rules of the previous section did not concern all the dominated patterns, we will now state two rules, that when applicated on a tdTWC decide that an input must be used. Notice, that for every tdTWC, with the exception of $S_{1.1}....S_{1.14}$, each user has to use at least two of its inputs, so at most one column and one row can be discarded.

Rule 1:

If a column (row) contains more partitions than the other two columns (rows), then this column (row) must be used.

Proof: considered trivial.

Rule 2:

Suppose we have a column (row) and we know for sure, that this column must be used. Now this column contains an element surrounded on both upper- and lower- (right- and left-) side by a partition. In this case the row (column) corresponding to this element must be used.

Proof:

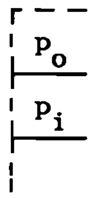


fig. 5.8 Column of a partition pattern

Regard the column of fig. 5.8. Define the input probability in this column outside the regarded element p_o, and inside this element p_i. Since the column is used, we have: $p_o > 0$, and we have to prove that

$p_i > 0$. The contribution of the element to R_{21} is:

$$-\frac{p_i}{p_i + p_o} \log \left(\frac{p_i}{p_i + p_o}\right) =: f(p_i) \qquad (5.3.1)$$

So $$f'(p_i) = -\frac{p_o}{(p_i+p_o)^2} \cdot \left(1 + \log \left(\frac{p_i}{p_i + p_o}\right)\right) \qquad (5.3.2)$$

Let $p_i \downarrow 0$, then $f'(p_i) \longrightarrow +\infty$.

Consider the function $R := R_{12} + \lambda\, R_{21}$, with $\lambda > 0$. Then $\left.\frac{\partial R}{\partial p_i}\right|_{p_i=0} = +\infty$.

Therefore a higher rate can be achieved when $p_i > 0$, thus using the row corresponding to p_i. This proves the fact that for achieving the Shannon outerbound the regarded row must be used. The same argument can be used to prove, that this holds for any input probability distribution, i.e. the row must always be used. □

If we look at the partition pattern of fig. 5.9, we have an example, of how these rules could be used.

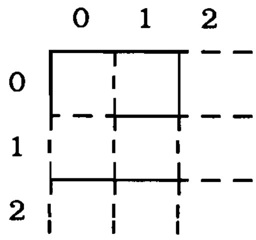


Fig. 5.9 Partition pattern of $S^b_{2.6}$

By in rule 1, we can say that column 1 and row 0 must be used. Applying in rule 2, results in the fact that also column 2 and the rows 1 and 2 must be used to achieve a high rate. Only column 0 might be discarded, and out rule 3 of section 5.2 tells us that we can indeed discard it, thus finding the dominating part pattern.

Chapter 6. Conclusions and recommendations

6.1 Conclusions

It has been shown, that a general method to compute strategies can be designed. Unfortunately only short strategies can be calculated, because of the fast increasing computing complexity, when strategy lengths become larger. However, the results of chapter 4 show that also with short strategies remarkable improvements upon the innerboundrate can be achieved. It also appears that there are no basic differences between T-channels and non T-channels w.r.t. to the problem of finding good Shannon strategies.

In general when channels are dominated by one of their part-patterns, the problem of finding a good strategy can be simplified substantially, because then we only have to deal with the part-pattern. It may happen that a good strategy for the part-pattern already exists so the problem is solved. And if not, then the part pattern generally has a simpler structure and needs a lower number of thresholds, so finding the solution costs less (computing-) time.

The class of dominated partition patterns is with 168 out of 308 non-trivial dtTWC's rather large, so in quite a lot of cases one is able to make use of this advantage. Also there is a great amount of regularity among these channels, since only by 5 rather simple rules one is able to predict the domination in 145 of the 168 cases.

With the help of the computed strategies, one is basically able to construct Markov strategies, and this gives good hope that with the help of computed Shannon strategies for other channels it also might be possible to construct Markov strategies rather easily.

6.2 Recommendations for further research

Since a very limited number of strategies has been studied, it is useful to study some more tdTWC's in order to get a better insight into coding strategies for these channels. Since domating part-patterns can be a very useful tool when constructing strategies or determining capacity regions, it is very interesting to make a more elaborate study of their behaviour, w.r.t. to the predictability. It is also useful to try to make a link between the Shannonbounds of a channel and its capacity region, so one could be able to prove the equivalence of two channels, by only considering Shannonbounds. However this may be very difficult.

Finally it is interesting to construct Markov strategies, with the help of the computed Shannon strategies.

References

[1] H.F.J.A. Engels, *Partition patterns van ternaire deterministische TWC's en hun dominating part-patterns*, information theory II report, Eindhoven University of Technology, Department of Electrical Engineering, Eindhoven, 1987.

[2] R. Fletcher, *Practical methods of optimization*, vol. 1, *Unconstrained optimization*, Wiley-Interscience, New York, 1986.

[3] E.W. Gaal, *Deterministic two-way channels, capacity regions, and strategies*, graduate report, Eindhoven University of Technology, Department of Electrical Engineering, Eindhoven, 1985.

[4] A. Jacobs, *On the capacity regions of ternary deterministic two-way channels*, Eindhoven University of Technology, Department of Electrical Engineering, Eindhoven, 1986.

[5] J.L. de Jong, *NONLINMIN, een procedure voor het minimaliseren van niet-lineaire functies onder niet-lineaire nevenvoorwaarden*, Colloquium Numerieke Programmatuur vol. 2, H.J.J. te Riele (ed.), MC-syllabus nr. 29.2 pp. 93-120, Amsterdam, 1977.

[6] 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.

[7] J.P.M. Schalkwijk, *On an extension of an achievable rate region for the binary multiplying channel*, IEEE Trans. Inform. Theory, vol. IT 29, pp. 445-448, May 1983.

[8] C.E. Shannon, *Two-way communication channels*, in Proc. 4th Berkely Symp. Math. Stat. Prob., vol. 1, pp. 611-644, 1961.
Reprinted in Key Papers in the Development of Information Theory, D. Slepian (ed.), pp. 339-372, IEEE, New York, 1974.

[9] B.J.M. Smeets, *Shannon's and other strategies for the binary multiplying channel*, Graduate report, Eindhoven University of Technology, Department of Electrical Engineering, Eindhoven, 1983.

[10] N.N.T. Thijssen, *Uitbreiding en verfijning van de Algol procedure NONLINMIN voor het minimaliseren van niet-lineaire functies onder niet-lineaire nevenvoorwaarden*, Graduate report, Eindhoven University of Technology, Department of Mathematics and Computing Science, Eindhoven, 1980.

Appendix A:

This appendix contains a list of all dominated partition patterns, the out rules, that can be applied and the number of the dominating part-patterns. The naming of the partition patterns is taken from [4] just like the numbers of the part-patterns. Part-patterns marked with * are [2,2] part-patterns; the other ones are either [3,2] or [2,3] part-patterns, but those are considered equivalent under the relation ~.

The rule numbers are always followed by a letter, 'h' denotes that the out rule should be applied on a row, 'v' is used for columns. Of course 1a,1b,1c means that resp. out rule 1a,1b or 1c should be applied.
When two rules are listed for one partition (one for columns, one for rows), those rules must be applied in the indicated order; the second rule to be applied must be regarded as a natural extension of the listed rule.

Partition Pattern	Rule nrs.	dominating part-pattern
$S^{a}_{2.2}$	2h,2v	4^{*}
$S^{b}_{2.2}$	3v,2h	4^{*}
$S^{c}_{2.2}$	3v,3h	4^{*}
$S^{a}_{2.3}$	2v	11
$S^{b}_{2.3}$	3v	11
$S^{a}_{2.4}$	3v,2h	5^{*}
$S^{b}_{2.4}$	2v,2h	4^{*}
$S^{c}_{2.4}$	3v,3h	5^{*}
$S^{d}_{2.4}$	3h,2v	4^{*}
$S^{a}_{2.5}$	3v,5h	5^{*}
$S^{b}_{2.5}$	4v,2h	4^{*}
$S^{c}_{2.5}$	3v,5h	5^{*}
$S^{d}_{2.5}$	3v	16
$S^{e}_{2.5}$	4v,5h	4^{*}
$S^{a}_{2.6}$	3v	14
$S^{b}_{2.6}$	3v	11
$S^{c}_{2.6}$	3v	14
$S^{d}_{2.6}$	3v	17
$S^{e}_{2.6}$	3v	11
$S^{f}_{2.6}$	3v	17
$S^{a}_{2.7}$	3v	18
$S^{b}_{2.7}$	2v	11
$S^{a}_{2.8}$	2v,2h	5^{*}
$S^{b}_{2.8}$	3v,2h	5^{*}
$S^{a}_{2.9}$	4v,2h	5^{*}
$S^{b}_{2.9}$	2v,5h	5^{*}
$S^{c}_{2.9}$	4v,3h	5^{*}
$S^{d}_{2.9}$	2v	16
$S^{e}_{2.9}$	4v,3h	5^{*}
$S^{f}_{2.9}$	2v,5h	5^{*}
$S^{a}_{2.10}$	4v,5h	5^{*}
$S^{b}_{2.10}$	4v,5h	5^{*}
$S^{a}_{2.11}$	2v	14
$S^{b}_{2.11}$	3v	14
$S^{c}_{2.11}$	2v	17
$S^{d}_{2.11}$	3v	17
$S^{a}_{2.12}$	4v	14
$S^{b}_{2.12}$	3v	17
$S^{c}_{2.12}$	3v	14
$S^{d}_{2.12}$	4v	17
$S^{e}_{2.12}$	3v	17
$S^{a}_{2.13}$	3v	18
$S^{b}_{2.13}$	2v	14
$S^{c}_{2.13}$	3v	18
$S^{d}_{2.13}$	2v	17
$S_{2.14}$	2v	18
$S^{a}_{3.4}$	2h	22
$S^{b}_{3.4}$	3h	22
$S^{a}_{3.5}$	5h	22
$S^{a}_{3.8}$	2h	32
$S^{b}_{3.8}$	3h	32
$S^{a}_{3.9}$	5h	32
$S^{c}_{3.9}$	5h	32
$S^{a}_{4.4}$	1a	6^{*}

Partition Pattern	Rule nrs.	dominating part-pattern
$S^b_{4.4}$	2v,3h	5^*
$S^c_{4.4}$	2h,2v	4^*
$S^a_{4.5}$	1a	6^*
$S^b_{4.5}$	4v,3h	5^*
$S^c_{4.5}$	3v	25
$S^d_{4.5}$	4v,2h	5^*
$S^e_{4.5}$	3v	25
$S^a_{4.6}$	3v	26
$S^b_{4.6}$	3v	22
$S^c_{4.6}$	3v	26
$S^d_{4.6}$	3v	28
$S^e_{4.6}$	3v	22
$S^f_{4.6}$	3v	28
$S^a_{4.7}$	3v	29
$S^b_{4.7}$	2v	22
$S^a_{4.8}$	1a	6^*
$S^b_{4.8}$	2v,2h	5^*
$S^a_{4.9}$	1a	6^*
$S^b_{4.9}$	1a	6^*
$S^c_{4.9}$	1a	6^*
$S^d_{4.9}$	2v	25
$S^e_{4.9}$	4v	25
$S^f_{4.9}$	2v	25
$S^a_{4.10}$	1a	6^*
$S^b_{4.10}$	4v	25
$S^a_{4.11}$	2v	26
$S^b_{4.11}$	3v	26
$S^c_{4.11}$	2v	28
$S^d_{4.11}$	3v	28
$S^a_{4.12}$	4v	26
$S^b_{4.12}$	3v	26
$S^c_{4.12}$	4v	26
$S^d_{4.12}$	3v	28
$S^e_{4.12}$	3v	26
$S^a_{4.13}$	3v	29
$S^b_{4.13}$	2v	26
$S^c_{4.13}$	3v	29
$S^d_{4.13}$	2v	28
$S_{4.14}$	2v	29
$S^a_{5.5}$	1a	6^*
$S^b_{5.5}$	5h	25
$S^e_{5.5}$	1a	6^*
$S^f_{5.5}$	5h	25
$S^a_{5.6}$	5v	26
$S^d_{5.6}$	5v	26
$S^g_{5.6}$	5v	28
$S^a_{5.7}$	5v	27
$S^a_{5.8}$	1a	6^*
$S^b_{5.8}$	4h,2v	5^*
$S^a_{5.9}$	1a	6^*
$S^b_{5.9}$	1a	6^*
$S^c_{5.9}$	1a	6^*
$S^d_{5.9}$	1a	6^*
$S^h_{5.9}$	?	25

Partition Pattern	Rule nrs.	dominating part-pattern
$S^{i}_{5.9}$	?	25
$S^{a}_{5.10}$	1a	6^*
$S^{c}_{5.10}$	1a	6^*
$S^{a}_{5.11}$	?	26
$S^{b}_{5.11}$	?	26
$S^{d}_{5.11}$	?	28
$S^{a}_{5.12}$	?	26
$S^{c}_{5.12}$	?	26
$S^{d}_{5.12}$	?	28
$S^{f}_{5.12}$	?	26
$S^{h}_{5.12}$	?	28
$S^{i}_{5.12}$	?	26
$S^{a}_{5.13}$	?	29
$S^{d}_{5.13}$	?	29
$S_{5.14}$	?	29
$S^{a}_{6.8}$	3h	35
$S^{b}_{6.8}$	3h	35
$S^{c}_{6.8}$	3h	32
$S^{a}_{6.9}$	5h	35
$S^{b}_{6.9}$	5h	35
$S^{g}_{6.9}$	5h	35
$S^{h}_{6.9}$	5h	35
$S^{a}_{7.8}$	1c	36
$S^{b}_{7.8}$	2h	32
$S^{a}_{7.9}$	1c	36
$S_{8.8}$	1a	6^*
$S^{a}_{8.9}$	1a	6^*
$S^{b}_{8.9}$	1a	6^*
$S_{8.10}$	1a	6^*
$S^{a}_{8.11}$	2v	35
$S^{b}_{8.11}$	3v	35
$S^{a}_{8.12}$	4v	35
$S^{b}_{8.12}$	3v	35
$S^{a}_{8.13}$	1b	36
$S^{b}_{8.13}$	2v	35
$S_{8.14}$	1b	36
$S^{a}_{9.9}$	1a	6^*
$S^{b}_{9.9}$	1a	6^*
$S^{c}_{9.9}$	1a	6^*
$S^{d}_{9.9}$	1a	6^*
$S^{e}_{9.9}$	1a	6^*
$S^{f}_{9.9}$	1a	6^*
$S^{a}_{9.10}$	1a	6^*
$S^{b}_{9.10}$	1a	6^*
$S^{c}_{9.10}$	1a	6^*
$S^{a}_{9.11}$	?	35
$S^{b}_{9.11}$	?	35
$S^{d}_{9.11}$	?	35
$S^{e}_{9.11}$	?	35
$S^{a}_{9.12}$	?	35
$S^{b}_{9.12}$	?	35
$S^{d}_{9.12}$	?	35
$S^{e}_{9.12}$	?	35
$S^{g}_{9.12}$	4v	35

Partition Pattern	Rule nrs.	dominating part-pattern
$S^{h}_{9.12}$	?	35
$S^{a}_{9.13}$	1b	36
$S^{d}_{9.13}$	1b	36
$S_{9.14}$	1b	36
$S^{a}_{10.10}$	1a	6^{*}
$S^{c}_{10.10}$	1a	6^{*}

Appendix B:

In this apppendix the values of the thresholds are listed. On the next page the location of the thresholds in relation to other thresholds can be seen. In the figure the situation is plotted for one direction; the indices of the horizontal thresholds can be found by adding found by add 182 to the indices of the vertical thresholds and then transposing the square.

On the next pages the results are given for $S_{3.3}$ and $S^{b}_{10.10}$. The tables have the threshold indices as entries and for each threshold either a real (this number is ≤ 1 and ≥ 0) or an integer number is given. If the number is a real, then we are dealing with an independent threshold and the number depicts the actual value of this threshold in the strategy (the numbers are 'absolute', i.e. the coordinate transform of chapter 3 does not have to be done anymore). If it is an integer, then we are dealing with a coupled threshold. The integer stands for the index of the independent threshold, this threshold is coupled to.

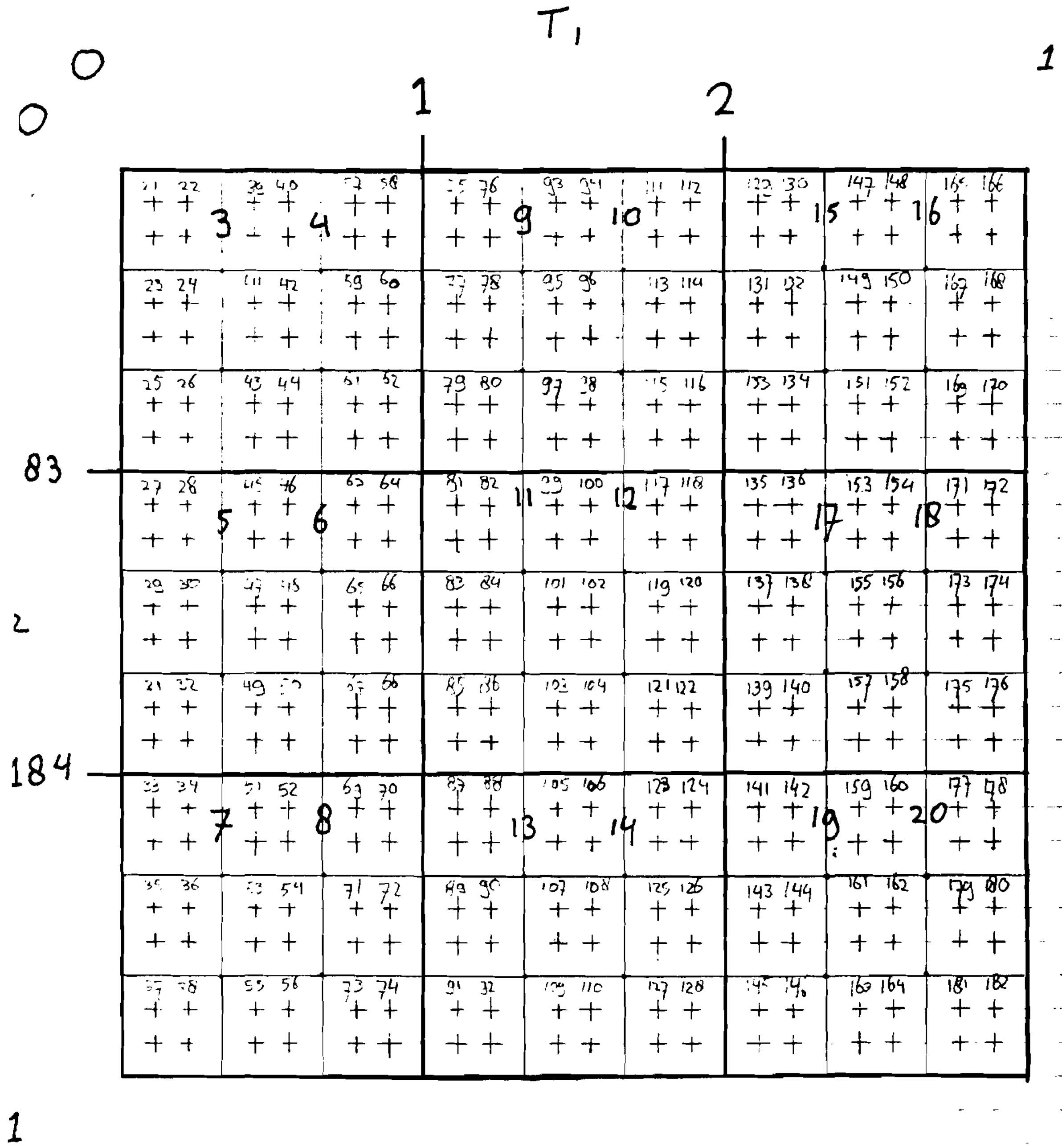


fig. B.1 Location of the thresholds

$S_{3.3}$

```
  1 6.435934832E-01
  2 8.217967309E-01
  3 3.874535647E-01
  4 5.155235285E-01
  5 0.000000000E+00
  6 3.217967327E-01
  7 1.918895588E-01
  8 4.177415178E-01
  9 8.217967309E-01
 10 8.217967309E-01
 11          9
 12         10
 13          9
 14         10
 15 1.000000000E+00
 16 1.000000000E+00
 17         15
 18         16
 19         15
 20         16
 21 2.544419563E-01
 22 3.209477605E-01
 23 0.000000000E+00
 24 1.937267823E-01
 25 0.000000000E+00
 26 1.937267823E-01
 27 0.000000000E+00
 28 0.000000000E+00
 29 0.000000000E+00
 30 0.000000000E+00
 31 0.000000000E+00
 32 0.000000000E+00
 33 0.000000000E+00
 34 9.594477938E-02
 35 0.000000000E+00
 36 9.594476255E-02
 37 0.000000000E+00
 38 9.594476250E-02
 39 5.155235285E-01
 40 5.155235285E-01
 41         39
 42         40
 43         39
 44         40
 45 2.113249114E-01
 46 2.665608221E-01
 47         45
 48         46
 49         45
 50         46
 51 4.177415178E-01
 52 4.177415178E-01
 53         51
 54         52
 55         51
 56         52
 57 6.435934832E-01
 58 6.435934832E-01
 59         57
 60         58
 61         57
 62         58
 63 5.331216536E-01
 64 5.883575693E-01
 65         63
 66         64
 67         63
 68         64
 69 6.435934832E-01
 70 6.435934832E-01
 71         69
 72         70
 73         69
 74         70
 75 8.217967309E-01
 76 8.217967309E-01
 77 7.597563995E-01
 78 7.907765671E-01
 79 7.597563622E-01
 80 7.907765459E-01
 81         75
 82         76
 83         77
 84         78
 85         79
 86         80
 87         75
 88         76
 89         77
 90         78
 91         79
 92         80
 93 8.217967309E-01
 94 8.217967309E-01
 95         93
 96         94
 97         93
 98         94
 99         93
100         94
101         93
102         94
103         93
104         94
105         93
106         94
107         93
108         94
109         93
110         94
111 8.217967309E-01
112 8.217967309E-01
113        111
114        112
115        111
116        112
117        111
118        112
119        111
120        112
121        111
122        112
123        111
124        112
125        111
126        112
127        111
128        112
129 1.000000000E+00
130 1.000000000E+00
131 9.379595963E-01
132 9.689798005E-01
133 9.379596590E-01
134 9.689798284E-01
135        129
136        130
137        131
138        132
139        133
140        134
141        129
142        130
143        131
144        132
145        133
146        134
147 1.000000000E+00
148 1.000000000E+00
149        147
150        148
151        147
152        148
153        147
154        148
155        147
156        148
157        147
158        148
159        147
160        148
161        147
162        148
163        147
164        148
165 1.000000000E+00
166 1.000000000E+00
167        165
168        166
169        165
170        166
171        165
172        166
173        165
174        166
175        165
176        166
177        165
178        166
179        165
180        166
181        165
182        166
183 5.917447672E-01
184 8.038499827E-01
185 3.562398066E-01
186 4.739922894E-01
```

```
187 0.000000000E+00
188 [illegible]
189 0.000000000E+00
190 2.955723977E-01
191 8.038499827E-01
192 8.038499827E-01
193       191
194       192
195       191
196       192
197 9.911565976E-01
198 9.955782982E-01
199       197
200       198
201       197
202       198
203 2.339437847E-01
204 2.950917957E-01
205 0.000000000E+00
206 1.781199033E-01
207 0.000000000E+00
208 1.781199033E-01
209 0.000000000E+00
210 0.000000000E+00
211 0.000000000E+00
212 0.000000000E+00
213 0.000000000E+00
214 0.000000000E+00
215 0.000000000E+00
216 0.000000000E+00
217 0.000000000E+00
218 0.000000000E+00
219 0.000000000E+00
220 0.000000000E+00
221 4.739922894E-01
222 4.739922894E-01
223       221
224       222
225       221
226       222
227 1.941235319E-01
228 2.449979528E-01
229       227
230       228
231       227
232       228
233 1.941236747E-01
234 2.449980453E-01
235       233
236       234
237       233
238       234
239 5.917447672E-01
240 5.917447672E-01
241       239
242       240
243       239
244       240
245 4.899959907E-01
246 5.408703812E-01
247       245
248       246
```

```
249       245
250       246
251 4.899959528E-01
252 5.408703627E-01
253       251
254       252
255       251
256       252
257 5.917447672E-01
258 6.977973749E-01
259 7.310349193E-01
260 7.674424510E-01
261 7.310349193E-01
262 7.674424510E-01
263       257
264       258
265       259
266       260
267       261
268       262
269       257
270       258
271       259
272       260
273       261
274       262
275 8.038499827E-01
276 8.038499827E-01
277       275
278       276
279       275
280       276
281       275
282       276
283       275
284       276
285       275
286       276
287       275
288       276
289       275
290       276
291       275
292       276
293 8.038499827E-01
294 8.038499827E-01
295       293
296       294
297       293
298       294
299       293
300       294
301       293
302       294
303       293
304       294
305       293
306       294
307       293
308       294
309       293
310       294
```

```
311 9.911565976E-01
312 9.911565976E-01
313 8.603899186E-01
314 9.257732598E-01
315 8.603899075E-01
316 9.257732527E-01
317       311
318       312
319       313
320       314
321       315
322       316
323       311
324       312
325       313
326       314
327       315
328       316
329 9.955782982E-01
330 9.955782982E-01
331       329
332       330
333       329
334       330
335       329
336       330
337       329
338       330
339       329
340       330
341       329
342       330
343       329
344       330
345       329
346       330
347 1.000000000E+00
348 1.000000000E+00
349       347
350       348
351       347
352       348
353       347
354       348
355       347
356       348
357       347
358       348
359       347
360       348
361       347
362       348
363       347
364       348

Rtot  =  5.5221E-01
R12  =   5.2036E-01
R21  =   5.8406E-01
```

$S^b_{10.10}$

```
  1 3.343783764E-01
  2 6.665717309E-01
  3 0.000000000E+00
  4 1.324393865E-01
  5         3
  6         4
  7 0.000000000E+00
  8 3.343783764E-01
  9 3.343783764E-01
 10 3.343783764E-01
 11 5.362048835E-01
 12 5.362048835E-01
 13        11
 14        12
 15 7.986351641E-01
 16 1.000000000E+00
 17 1.000000000E+00
 18 1.000000000E+00
 19        15
 20        16
 21 0.000000000E+00
 22 0.000000000E+00
 23        21
 24        22
 25 0.000000000E+00
 26 0.000000000E+00
 27        21
 28        22
 29        21
 30        22
 31        25
 32        26
 33 0.000000000E+00
 34 0.000000000E+00
 35        33
 36        34
 37 0.000000000E+00
 38 0.000000000E+00
 39 0.000000000E+00
 40 0.000000000E+00
 41 4.414646720E-02
 42 8.829292823E-02
 43        41
 44        42
 45        39
 46        40
 47        41
 48        42
 49        41
 50        42
 51 1.114594588E-01
 52 2.229189176E-01
 53 1.114594588E-01
 54 2.229189176E-01
 55        53
 56        54
 57 2.138848205E-01
 58 3.343783764E-01
 59 1.997523845E-01
 60 2.670653804E-01
 61        57
 62        58
 63        57
 64        58
 65        59
 66        60
 67        57
 68        58
 69 3.343783764E-01
 70 3.343783764E-01
 71 3.343783764E-01
 72 3.343783764E-01
 73        69
 74        70
 75 3.343783764E-01
 76 3.343783764E-01
 77        75
 78        76
 79 3.343783764E-01
 80 3.343783764E-01
 81 4.158176108E-01
 82 5.362048835E-01
 83        81
 84        82
 85 4.016538788E-01
 86 4.689293811E-01
 87        81
 88        82
 89        81
 90        82
 91        85
 92        86
 93 3.343783764E-01
 94 3.343783764E-01
 95 3.343783764E-01
 96 3.343783764E-01
 97        95
 98        96
 99 5.362048835E-01
100 5.362048835E-01
101 5.362048835E-01
102 5.362048835E-01
103       101
104       102
105        99
106       100
107       101
108       102
109       101
110       102
111 4.451094946E-01
112 5.558406127E-01
113 4.451094782E-01
114 5.558405694E-01
115       111
116       112
117 5.796604993E-01
118 6.231161151E-01
119 5.362048835E-01
120 5.362048835E-01
121       117
122       118
123       117
124       118
125       119
126       120
127       117
128       118
129 7.105928810E-01
130 7.546140387E-01
131       129
132       130
133 6.665717309E-01
134 7.986351641E-01
135 7.777144872E-01
136 8.888572436E-01
137       135
138       136
139 7.777144646E-01
140 8.888572231E-01
141       129
142       130
143       129
144       130
145       133
146       134
147 8.657567761E-01
148 9.328783880E-01
149 9.187879149E-01
150 9.187879149E-01
151       149
152       150
153 1.000000000E+00
154 1.000000000E+00
155 1.000000000E+00
156 1.000000000E+00
157       155
158       156
159       147
160       148
161       149
162       150
163       149
164       150
165 1.000000000E+00
166 1.000000000E+00
167 1.000000000E+00
168 1.000000000E+00
169       165
170       166
171 1.000000000E+00
172 1.000000000E+00
173 1.000000000E+00
174 1.000000000E+00
175       171
176       172
177       165
178       166
179       167
180       168
181       165
182       166
183 3.334352074E-01
184 6.618990948E-01
185 0.000000000E+00
186 3.147026337E-01
```

```
187 0.000000000E+00
188 1.279590216E-01
189       187
190       188
191 5.358791360E-01
192 5.358791360E-01
193 3.334352074E-01
194 3.334352074E-01
195       191
196       192
197 7.955037084E-01
198 1.000000000E+00
199       197
200       198
201 9.811371090E-01
202 1.000000000E+00
203 0.000000000E+00
204 0.000000000E+00
205 0.000000000E+00
206 0.000000000E+00
207       205
208       206
209 0.000000000E+00
210 0.000000000E+00
211 0.000000000E+00
212 0.000000000E+00
213       211
214       212
215       209
216       210
217       211
218       212
219       211
220       212
221 1.049008779E-01
222 2.098017558E-01
223 1.065910787E-01
224 2.130584625E-01
225       221
226       222
227 4.265301333E-02
228 8.530602059E-02
229 0.000000000E+00
230 0.000000000E+00
231       227
232       228
233       227
234       228
235       229
236       230
237       227
238       228
239 3.192566156E-01
240 3.239342828E-01
241       239
242       240
243 3.334352074E-01
244 3.334352074E-01
245 2.094269473E-01
246 3.334352074E-01
247       245
248       246
249 1.964510836E-01
250 2.649431455E-01
251       245
252       246
253       245
254       246
255       249
256       250
257 4.009165170E-01
258 4.683978265E-01
259 3.334352074E-01
260 4.136909837E-01
261       259
262       260
263 3.334352074E-01
264 3.334352074E-01
265 3.334352074E-01
266 3.334352074E-01
267       265
268       266
269       257
270       258
271       259
272       260
273       259
274       260
275 5.358791360E-01
276 5.358791360E-01
277 5.358791360E-01
278 5.358791360E-01
279       275
280       276
281 3.334352074E-01
282 3.334352074E-01
283 3.334352074E-01
284 3.334352074E-01
285       281
286       282
287       275
288       276
289       277
290       278
291       275
292       276
293 5.778857917E-01
294 6.198924437E-01
295       293
296       294
297 6.618990948E-01
298 6.618990948E-01
299 4.429231699E-01
300 5.524111324E-01
301       299
302       300
303 4.429231699E-01
304 5.524111324E-01
305       293
306       294
307       293
308       294
309       297
310       298
311 7.955037084E-01
312 7.955037084E-01
313 7.064339660E-01
314 7.509688372E-01
315       313
316       314
317       311
318       312
319       313
320       314
321       313
322       314
323 7.674554109E-01
324 8.710546141E-01
325 7.683117662E-01
326 8.747244376E-01
327       325
328       326
329 7.955037084E-01
330 8.778513659E-01
331 8.636691390E-01
332 9.318345695E-01
333       329
334       330
335       329
336       330
337       331
338       332
339       329
340       330
341 9.882811145E-01
342 9.973821905E-01
343 9.811371090E-01
344 9.811371090E-01
345       341
346       342
347 1.000000000E+00
348 1.000000000E+00
349       347
350       348
351 1.000000000E+00
352 1.000000000E+00
353       347
354       348
355       347
356       348
357       351
358       352
359 1.000000000E+00
360 1.000000000E+00
361       359
362       360
363 1.000000000E+00
364 1.000000000E+00

Rtot = 5.905928112E-01
R12  = 5.907267471E-01
R21  = 5.904588752E-01
```