DISCRETE NOISELESS CODING

by

RICHARD S. MARCUS

A.B., University of Pennsylvania
(1954)

B.S. in E.E., University of Pennslyvania
(1955)

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February, 1957

Signature of Author _____
Department of Electrical Engineering, January 14, 1957

Certified by _____
Thesis Supervisor

Accepted by _____
Chairman, Departmental Committee on Graduate Students

DISCRETE NOISELESS CODING

by

RICHARD S. MARCUS

Submitted to the Department of Electrical Engineering on January 14, 1957 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

This paper considers the problem of efficient coding (in the information theory sense) for finite, discrete, memoryless message sources and finite, discrete, memoryless, noiseless channels. It describes important known results and methods and includes some new results. Various classes of the coding problem are clearly distinguished. Emphasis is placed on the classes in which the number of message blocks is restricted either to the number of original messages or to the number of channel symbols, whichever is larger. However, procedures for larger numbers of message blocks, which lead to perfect efficiency, are also discussed. Various bounds on the efficiency are described for different procedures.

The case of cost-weighted channel symbols is discussed in parallel with the equal-cost case which has received the most attention in the literature so far. Cost-weighted symbols include those which have, for instance, unequal time durations. An extension of the Shannon procedure and bounds to this cost-weighted case is described. An interesting question as to the admissibility of proper signal sets in the cost-weighted case is raised but not solved.

Thesis Supervisor: Peter Elias
Title: Associate Professor of Electrical Engineering

ACKNOWLEDGMENT

The author is deeply indebted to Professor Peter Elias whose instructive suggestions and patient guidance made this thesis possible.

In addition, a grateful acknowledgment is due to Professors Robert M. Fano, David A. Huffman, Claude E. Shannon, and Dr. Marcel P. Schutzenberger, all currently at the Massachusetts Institute of Technology, whose work forms a large part of this thesis and who each took the trouble to read the thesis in its manuscript form.

Finally, the General Communications Company and the Research Laboratory of Electronics are to be thanked for their support during the preparation of the thesis.

TABLE OF CONTENTS

TABLE OF CONTENTS (2)

DISCRETE NOISELESS CODING

1. THE CODING PROBLEM

1.1 THE COMMUNICATION SYSTEM

We are interested in coding as it may be applied to the transmission of information in any communication system. For this discussion we shall adopt the point of view accepted in information theory as proposed by Shannon.[1*] From this point of view we conceive the general communication system as containing (among other things) 1) an information source or message source, 2) a transmitter or coder, 3) a channel, and 4) a receiver. The aim is to transmit the information to the receiver in some efficient way. To do this the transmitter codes the messages from the information source into signals which it sends over the channel. Here, then, coding involves the representation of messages by signals. The coding device may be thought of as a transducer which accepts information in one form and sends it out in a more useful form. We would like to make this transducer error free, efficient, and simple.

In this discussion we consider only discrete systems in which both the message and the signal are selected from sets of a finite number of elements. The message set contains m elements (written m_i ; $i = 1,2,\ldots,m$). The signal set contains D elements which may be called channel symbols (written d_j ; $j = 1,2,\ldots,D$). The signal set may be called an alphabet. Coding may then be described as the process whereby the message, m_i, or a sequence of messages called a message block (written M_k ; $k = 1,2,\ldots,M$) is replaced by a sequence of channel symbols called a code word written W_k ; $k = 1,2,\ldots,M$).

In general, one may associate with each symbol d_j a cost c_j. This cost is most frequently thought of as a time duration of the symbol ($c_j = t_j$ seconds) but it may be expressed in terms of power, bandwidth, or any other economically motivated consideration. If we do not wish

---

*Superscripts refer to references listed in the Bibliography.

to specify a particular unit, we may give the general unit "unc" to the c_j's.*

Each message has a certain a priori probability of occurence which is written p_i. For this discussion we consider these probabilitites to stay fixed and the source is then called "memoryless". (A source which has memory may be represented, to a first approximation, by a memoryless one by considering p_i as the probability of m_i averaged over all possible situations). The information content or entropy, H, of a memoryless message source is given by[1] $H = -\Sigma\, p_i \log p_i$ bits/message** where the logarithm is to the base two. For a channel with memory H will actually be less than this, the difference increasing with the dependence on past history.

1.2 CHANNEL CAPACITY

The channel capacity, C, may be defined as the maximum rate at which information may be transmitted over the channel. Shannon[1] defines the capacity as follows

$$C = \lim_{T \to \infty} \frac{\log N(T)}{T} \qquad \text{bits/second} \tag{1}$$

where N(T) = the number of allowed signals of duration T.
This definition will be shown to be equivalent if we replace seconds by uncs.

To find the rate of transmission, R, over the channel we need to know the probability with which each symbol, d_j, is used. Let us write these probabilities p_j. Then R is given by

$$R = \frac{-\Sigma\, p_j \log p_j}{\Sigma\, p_j c_j} \qquad \text{bits/unc} \tag{2}$$

We let the average cost per symbol be written $\bar{c}$ where

$$\bar{c} = \Sigma\, p_j c_j \qquad \text{uncs/symbol} \tag{3}$$

---

*Unc, which rhymes with bunk, is an abbreviation for "unit cost".

**In this paper the summation is taken over the entire range of the index unless otherwise noted.

Thus the rate is the ratio of the average information content per symbol to the average cost per symbol. Then to find C we want to maximize R, i.e., find the set of p_j's for which R is a maximum. Let us assume that we are free to use the symbols in any order we choose, (Otherwise there are constraints in the channel. This case will be treated in an example in Section 1.3). Then we may maximize R subject only to the constraint:

$$\Sigma\, p_j = 1 \tag{4}$$

To do this we use the method of Lagrange multipliers. Let

$$y = R + \lambda \Sigma\, p_j \tag{5}$$

$$\frac{\partial y}{\partial p_j} = \frac{\bar{c}(-\ln 2 \; - \log p_j) + c_j \Sigma\, p_j \log p_j}{\bar{c}^2} + \lambda \tag{6}$$

where $\ln 2 = \log_e 2$

Now let

$$\frac{\partial y}{\partial p_j} = 0 \qquad \text{for } p_j = P_{mj} \quad \text{(the maximizing probabilities)} \tag{7}$$

Then

$$\frac{\bar{c}(\ln 2 \; + \log P_{mj}) - c_j \; \Sigma\, P_{mj} \log P_{mj}}{\bar{c}^2} - \lambda = 0 \tag{8}$$

But

$$\Sigma\, P_{mj} \log P_{mj} = -C\bar{c} \tag{9}$$

Hence

$$\bar{c}(\ln 2 + \log P_{mj} + Cc_j) = \lambda \bar{c}^2 \tag{10}$$

To evaluate λ let us multiply by P_{mj} and sum for all j:

$$\bar{c}(C\, \Sigma\, P_{mj} c_j \; + \Sigma\, P_{mj} \log P_{mj} + \ln 2\, \Sigma\, P_{mj})$$

$$= \lambda \bar{c}^2\, \Sigma\, P_{mj} = \lambda \bar{c}^2 = \bar{c}(C\bar{c} - C\bar{c} + \ln 2) \tag{11}$$

Hence

$$\lambda = \frac{\ln 2}{\bar{c}} \tag{12}$$

Then

$$\bar{c} \log P_{mj} = \bar{c}(\ln 2 - \ln 2 + Cc_j) \tag{13}$$

or

$$P_{mj} = 2^{-Cc_j} \tag{14}$$

Since $\Sigma P_{mj} = \Sigma 2^{-Cc_j} = 1$, we have C as the real-valued solution to the equation

$$\Sigma 2^{-Cc_j} = 1 \tag{15}$$

It can be shown* that $R < C$ if any of the p_j's depend on previous symbols. C, in general, is measured in bits per unc. For $c_j = t_j$ seconds, C will be measured in bits per second. For the case in which all c_j are equal (say to c_1) we have

$$\Sigma 2^{-Cc_j} = D2^{-Cc_1} = 1 \qquad \text{or} \qquad D = 2^{Cc_1} \tag{16}$$

Then

$$C = \frac{\log D}{c_1} \tag{17}$$

1.3 EXAMPLES OF CHANNELS - TELEGRAPH AND TELETYPE

The teletype channel has two symbols which are used in all possible permutations of length five. As an idealization we may say that these two symbols are equal in cost with $c_1 = c_2 = 30$ milliseconds. Then from eq. 17 we have

$$C = (\log 2)/30\text{x}10^{-2} \doteq 33 \text{ bits/second} \tag{18}$$

If we measure cost by some factor other than time we might find, however, that, for instance, $c_1 = 1$ unc and $c_2 = 2$ uncs. The capacity is then the solution of the equation

$$2^{-C} + 2^{-2C} = 1 \tag{19}$$

which is $C = 0.695$ bits/unc.

The International Morse Code uses a channel which has several symbols which may be weighted according to their time duration. One way to describe the symbols follows on the next page[3]:

---

*See Fano[2], p. III 36, or Shannon[1], p. 22. We must, of course, extend the definition of R suitably. The value of C in (15) is the same as Shannon obtains with his definition for C.

SYMBOL		DESCRIPTION	TOTAL COST
d_1	dot	one unit of time "on", one "off"	2 taps
d_2	dash	three " " " " " "	4 "
d_3	letter space	two units of time "off"	2 "
d_4	word space	five " " " "	5 "

The unit cost is now the unit of time which we call the tap. In this case, however, we are not free to use the symbols in whatever order we choose. For instance, we cannot distinguish at the receiver five letter spaces in a row from two successive word spaces. Practically, any number of successive spaces become very difficult to decode at the receiver. We may then place theconstraint on the channel that two spaces may not be sent successively. To calculate the capacity we may consider a new set of d_j's which can be used with no constraints and which generates all allowable sequences of the old d_j's. For example:

SYMBOL	TOTAL COST
$d_1' = d_1$	2 taps
$d_2' = d_2$	4 "
$d_3' = d_3 d_1$	4 "
$d_4' = d_3 d_2$	6 "
$d_5' = d_4 d_1$	7 "
$d_6' = d_4 d_2$	9 "

Thus C is the solution of

$$2^{-2C} + 2^{-4C} + 2^{-4C} + 2^{-6C} + 2^{-7C} + 2^{-9C} = 1 \qquad (20)$$

which gives $C = 0.590$ bits/tap.* This value for C is again the same as would be obtained with the Shannon definition. In this manner a channel with constraints can be transformed into one without constraints having the same capacity.

---

*Shannon[1] gives a slightly higher cost to the spaces and gets $C = 0.539$ bits/tap.

1.4 THE FUNDAMENTAL THEOREM

The Fundamental Theorem for a noiseless channel states[1] that it is possible to encode any message source into any channel with capacity C in such a way that

$$R = C - \epsilon \qquad \text{for any } \epsilon > 0 \tag{21}$$

and it is not possible to perform an encoding such that $R > C$.

This theorem suggests that C is really a good measure of channel capacity since we can transmit information over the channel at a rate as close to its capacity as we like with suitable coding procedures. This, in turn, provides substantial evidence to support the soundness of the definition of the measure of information. (See Fano[2] p. III 10 for an elaboration of this point).

The latter part of the theorem is true since we have defined C as the maximum R. To prove the first part[1] we consider all possible sequences of messages of length L. We then consider all possible code words of a given cost. We may then show that the latter number is big enough to encode a large enough group of the former such that the information rate approaches C as L approaches infinity. The sequences of messages which are not coded by code words of the given cost must be coded by more costly code words, However, this group has such small probability as L approaches infinity that the information rate still approaches C.

This is of the nature of an existence proof. A constructive procedure which attains a rate C in the limit is also given by Shannon[1] for the special case of equal-cost channel symbols. This method is extended to the cost-weighted case in this paper.

1.5 CRITERIA FOR GOOD CODES

One criterion for a good code will be its efficiency, e, which is defined by $e = R/C$. Another criterion for a good code is the ease with which it can be implemented by apparatus in any given situation. There is no single mathematical measure for this latter criterion but there are some measures which may give us an idea of the

complexity of the code and, hence, possibly the difficulty of its implementation. These are the number of message blocks, M, (which is also the number of code words), the maximum word cost, $c_{k\ max}$, and the number of messages in the longest message block, n_{max}.

M may indicate the size of the "code book" needed or may otherwise indicate the complexity in the coding apparatus. $c_{k\ max}$ is similarly important. If cost is in time then $c_{k\ max}$ determines the longest delay in being able to determine what the message was. It therefore measures the storage capacity necessary at the decoding apparatus. If the cost is in power or bandwidth, etc., there may be some upper limit beyond which it would be impossible for the costliest signal to go due to physical limitations of the apparatus. Similarly, n_{max} may measure the storage capacity needed at the encoder.

1.6 CLASSES OF THE CODING PROBLEM

The coding problem may be divided into three classes depending on the relative numbers of original messages, m, message blocks (and hence code words), M, and channel symbols, D. In Class I the number of messages is greater than or equal to the number of channel symbols. The original messages are represented directly by code words. Class II occurs when there are fewer messages than channel symbols. Here the messages are coded first into message blocks which are represented directly by channel symbols. Class III considers the most complicated type of coding in which messages are coded into blocks and are represented by code words. Summarizing, we find the following relations:

Class I	$m = M \geq D$	
Class II	$m \leqslant M = D$	(22)
Class III	$m < M > D$	

Each of the above classes may be subdivided into two groups:

Group A	Equal-cost, all c_j are equal
Group B	Cost-weighted, some c_j may be unequal

Classes IA, IIIA, and to a lesser extent IIIB, have received the most attention in the literature so far. Class III has been analyzed mostly for the limiting case of perfect efficiency. This paper considers all the classes.

2. SOME PRELIMINARY CONSIDERATIONS ON CODING

2.1 THE TREE GRAPH FOR CODING

It has been found convenient to think of all the possible code words as branches on a tree-like structure. (See Figs. 1, 3, and 4). From the root on the left extend D branches to the right, each one representing one symbol of the alphabet. The projection of the length of each branch on a horizontal axis is made proportional to the cost of the symbol. Since $\log P_{mj} = -Cc_j$ by eq. 13, we may make the projected length of each branch equal to $(-\log P_{mj})$. From the right node of each branch extends another "fan" of D branches to the right. The D^2 branches in these D fans then represent all possible code words with just two symbols. By further extensions code words of greater numbers of symbols are represented. The code word which any branch represents can be found as the sequence of channel symbols associated with the branches on the path from the root to the branch in question (including that branch). The horizontal distance from the root to the far node of any branch is the normalized cost, written q_k, of the word, i.e., the cost multiplied by C. It is measured in bits. (Note that one multiplies by C to get the normalized cost rather than divides since C varies inversely as the costs).

2.2 DECODABILITY

A set of branches from the tree will then represent the code words of a signal set. One necessary restriction on the signal set (for error-free operation) is that it be decodable. That is, when a sequence of symbols arrives at the receiver we must be able to decompose this sequence in a unique manner into code words. This problem is considered in detail later (see e.g., Section 10.1) but for the present we may say that a sufficient condition for this unique decomposability

is that no code word be a prefix of any other code word. This condition is sufficient since knowing when a code word begins we can determine when it ends.

In terms of this tree picture the prefix condition states that no used branch is allowed to lie along the path of another used branch. (A used branch is a branch which represents a code word that is in the signal set and the path of a branch is the sequence of branches from the root to that branch). We term a signal set which obeys the prefix rule as proper.[4] On the other hand we may consider sets of code words such that any infinite sequence of channel symbols has at least one code word as a prefix. Such a set will be called complete. If there is one and only one code word as a prefix to any infinite sequence, then the set is both complete and proper and the used branches form a cut set of the tree.

2.3 THE KRAFT INEQUALITY

Let us demonstrate a certain inequality. First we shall define the structure function, S_p as follows:

$$S_p = \Sigma_{k=1}^{p} 2^{-q_k} \tag{24}$$

This function tells us, in some sense, how much of the tree is "used up" by the branches representing W_1 through W_p. The value for S_p when p is the total number of branches considered is written S.

Now consider the cut set of just D branches. For these branches

$$S = \Sigma_{k=1}^{D} 2^{-q_k} = \Sigma_{j=1}^{D} 2^{\log P_{mj}} = \Sigma P_{mj} = 1 \tag{25}$$

Now consider the cut set formed by replacing one of these branches by the D branches in the fan at its right. Say this was the branch for the symbol a. Then for the sum $S = \Sigma_{k=1}^{2D-1} 2^{-q_k}$ we have the same terms as before except instead of $2^{\log P_{mj}} = P_{ma}$ we have $\Sigma_{j=1}^{D} 2^{\log P_{ma} + P_{mj}} = P_{ma} \Sigma_{j=1}^{D} P_{mj} = P_{ma}$. Therefore, $S = 1$ as before. It is easily seen that if we replace any branch by the D branches extending in a fan from it , the S value will remain unchanged. Since any cut set can be generated from the cut set of D branches by a series of such replacements

(in which D - 1 branches are added at each step), then the value of S for any cut set must be one. But a proper code must consist of the branches of a cut set or some of the branches of a cut set but not more than a cut set. Therefore, for a proper code $S \leq 1$ in which the equality holds if and only if the code is complete. This is known as the Kraft[5] inequality.*

3. DISCUSSION OF CLASS IA

3.1 BEST EFFICIENCY - IDEAL AND ACTUAL

Let us first consider coding for the case designated as Class IA. For equal-cost symbols the tree is symmetrical and the cost of word W_k is just $q_k = n_k \log D$, where n_k is the number of symbols in W_k. We note that R can be expressed in terms of the word costs and probabilities rather than channel symbol figures as follows:

$$R = \frac{-\Sigma\, p_k \log p_k}{\Sigma\, p_k q_k} = \frac{H}{\Sigma\, p_k n_k \log D} = \frac{H}{\bar{n} \log D} \qquad (23)$$

where p_k is the probability of the k^{th} message block (which is here a single message)**

H is the entropy of the message source

$\bar{n} = \Sigma\, p_k n_k$ = the average number of symbols per message

We see then, that in Class IA, for a given channel of D symbols and source with entropy H, the problem is to minimize the average number of symbols per message, $\bar{n}$.

Now let us determine under what conditions the value for $\bar{n}$ will be a minimum. As we have seen $q_k = n_k \log D$ for the equal-cost case. For a proper code we have:

$$S = \Sigma_{k=1}^{M} 2^{-q_k} = \Sigma\, 2^{-n_k \log D} = \Sigma\, D^{-n_k} \leq 1 \qquad (26)$$

---

*It has also been called the Szilard inequality, since Mandelbrot[6] claims Szilard[7] discovered it in a structurally indentical problem concerning Maxwell Demons in Thermodynamics.

**Note eq. 23 assumed that the p_k's are constant, i.e., memoryless coding. The notation p_k for the probabilities of the messages blocks should not be confused with p_j and p_i since the context should make the meaning clear.

Now we minimize $\bar{n} = \Sigma\, p_k n_k$ subject to the constraint $\Sigma\, D^{-n_k} \leq 1$ in order to find the ideal value for $\bar{n}$. For $\bar{n}$ to be minimum the equality sign must hold in the Kraft inequality. Otherwise we could reduce some n_k while still keeping $S \leq 1$ and hence reduce $\bar{n}$. We may now minimize $\bar{n}$ subject to the constraint $\Sigma\, D^{-n_k} = 1$. We again employ the Lagrange multiplier technique: Let

$$y = \Sigma\, p_k n_k + \lambda(\Sigma\, D^{-n_k} - 1) \tag{27}$$

To find the value of n_k which minimizes $\bar{n}$ we set $\frac{\partial y}{\partial n_k} = 0$

$$\frac{\partial y}{\partial n_k} = p_k - \lambda D^{-n_k} \ln D = 0 \tag{28}$$

To evaluate λ we sum over k

$$1 = \Sigma\, p_k = \lambda \ln D\, \Sigma\, D^{-n_k} = \lambda \ln D \tag{29}$$

Hence,

$$\lambda = 1/\ln D \tag{30}$$

$$D^{-n_k} = p_k \tag{31}$$

or,

$$n_k = -\log_D p_k = \frac{-\log p_k}{\log D}$$

The minimum value of $\bar{n}$ is then $\bar{n}_{min}$ where

$$\bar{n}_{min} = \Sigma\, p_k n_k = \frac{-\Sigma\, p_k \log p_k}{\log D} = \frac{H}{\log D} \tag{32}$$

This gives the ideal efficiency, e_{max}

$$e_{max} = \frac{H}{\bar{n}_{min}} \times \frac{1}{C} = \frac{H}{H/\log D} \times \frac{1}{\log D} = 1 \tag{33}$$

which is also the best we could do according to the Fundamental Theorem

We see from eq. 31 that to code with maximum efficiency we want $n_k = -\log_D p_k$. Because $(-\log_D p_k)$ is in general not an integer this cannot be done exactly. This is an example of a typical problem which occurs in discrete coding: One has to use integral values of a variable to satisfy certain conditions and in general one can only approximate

the desired result. The actual best efficiency then, in general, falls short of the ideal.

In finding a soltion for a certain class of cases one may ask two kinds of questions: 1) How well does this solution do (say with respect to efficiency) compared to the best possible solution (either ideal ($e = 1$) or the actual optimum solution)? 2) What bound on the efficiency, good for the whole class, can be achieved by this solution? A solution that gives a good answer to one of these questions does not necessarily provide a good answer to the other.

3.2 SUFFICIENCY OF COST SET SATISFYING KRAFT INEQUALITY

Let us show that a proper code exists with a given set of n_k if the set satisfies the Kraft inequality. We can show that this is so by a constructive procedure. The structure function may be written

$$S = \Sigma_{a=1}^{L} N(a)\, D^{-a} \qquad a = 1,2,\ldots,L \tag{34}$$

where L is the length of the longest word

$N(a)$ = the number of words such that $n_k = a$.

Now let us pick $N(1)$ code words of length 1. In general, there will be some code words of length 2 which do not have any of the $N(1)$ code words of length one as prefixes. Pick $N(2)$ of these. Now there should be some code words of length three which are not prefixed by any of the used words of length one or two. This process can be continued until there are code words for all of the n_k. The only reason that this wouldn't work is that the tree got "used up". But this would mean $S_p > 1$ for some $p \leq M$. But this would violate the Kraft inequality. Thus we have proven our contention.

3.3 THE SHANNON PROCEDURE

We have seen that in general we cannot pick $n_k = -\log_D p_k$ because $(-\log_D p_k)$ may not be an integer. But we can pick n_k such that $n_k = \lceil -\log_D p_k \rceil$, where $[x]$ means the smallest integer that is bigger than or equal to x. There are several ways in which this procedure may be shown to work. Let us order the messages according to decreasing probability. Then let us pick a code word for message m_1 of length $n_1 = \lceil -\log_D p_1 \rceil$. Similarly, we pick code word W_2 such that $n_2 = \lceil -\log_D p_2 \rceil$ and W_2 obeys the prefix rule. This procedure is continued, always picking n_k greater than but as close as possible to the ideal value. If the procedure is to fail it would be because there was no suitable branch to pick, i.e., for some p, $S_p > 1$. But we have picked $n_k \geq -\log_D p_k$. Therefore,

$$S = S_M = \Sigma\, D^{-n_k} \leq \Sigma\, D^{-\log_D p_k} = \Sigma\, p_k = 1 \tag{35}$$

Hence,

$$S_p \leq S_M \leq 1 \qquad \text{for } p \leq M \tag{36}$$

Therefore, the procedure will always work.

In the Shannon procedure the number of symbols for any word is bounded by

$$-\log_D p_k \leq n_k = \lceil -\log_D p_k \rceil < -\log_D p_k + 1 \tag{37}$$

This, in turn, gives the following bound on the average number of symbols per message:

$$\bar{n} = \Sigma\, p_k n_k < \Sigma\, p_k(-\log_D p_k + 1) = \Sigma\, p_k - \frac{\Sigma\, p_k \log p_k}{\log D} = 1 + \frac{H}{\log D} \tag{38}$$

Hence,

$$\bar{n} < \frac{H}{\log D} + 1 \tag{39}$$

Hence,

$$e = \frac{H}{\bar{n}C} > \frac{H(1/\log D)}{\frac{H}{\log D} + 1} = \frac{1}{\frac{\log D}{H} + 1} \tag{40}$$

Comments on the goodness of this method will come after other methods

have been described.

Before we mention other methods we may show how this method could be proven in other ways. One way is to show that the signal set satisfies the Kraft inequality:

$$\Sigma\, D^{-n_k} = \Sigma\, D^{-\left[-\log_D p_k\right]} \leq \Sigma\, D^{\log_D p_k} = \Sigma\, p_k = 1 \tag{41}$$

Therefore, by the sufficiency of the Kraft inequality, there exists a proper signal set with these n_k.

Shannon[1] describes a simple procedure for performing the above type of coding for the binary alphabet and showing that this gives a code obeying the prefix rule. W_p is formed by expanding E_p as a binary number for n_p places where

$$E_p = \Sigma_{k=1}^{p-1} p_k, \qquad n_p = \left[-\log p_p\right] \tag{42}$$

W_p must differ from W_k for $k > p$ since E_k must be greater than E_p by at least 2^{-n_p} and hence must have a different binary expansion in the first n_p places.

Of course the Shannon procedure is not optimum in general since in most cases we could reduce some of the n_k and still have a prefix code. We could describe modified procedures following the same general pattern which would be better but they would still not be optimum and we could not easily obtain better bounds from them. We could, for instance, describe a somewhat improved method for picking the words originally. This would be to pick the cheapest word available for W_p such that $S_p \leq E_{p+1}$. This would work since $S_p \leq S_M = S \leq E_{m+1} = 1$.

4. THE EXTENDED SHANNON PROCEDURE (IB)

Let us now consider how the Shannon procedure may be extended to the case of cost-weighted symbols. First we must see what costs should be associated ideally with the words designating the various messages. We wish to minimize the average cost of a message, $\bar{w}$

$$\bar{w} = \frac{\bar{q}}{C} = \frac{\Sigma\, p_k q_k}{C} \tag{43}$$

subject to the constraint $\Sigma\, 2^{-q_k} \leq 1$. This is very similar to the

variational problem we solved to minimize $\bar{n}$ and we see the solution to be:

$$q_k = -\log p_k \tag{44}$$

The efficiency is then:

$$e = \frac{H}{\bar{q}} = \frac{-\Sigma\, p_k \log p_k}{-\Sigma\, p_k \log p_k} = 1 \tag{45}$$

Since maximum efficiency is achieved, the channel is being used in the optimum way and the symbols are then being used with probabilities given by $P_{mj} = 2^{-Cc_j}$. It is true here, as in the equal-cost case, that $(-\log p_k)$ may not correspond to the cost of any possible word. However, we may describe the following extension to the Shannon procedure:

Order the messages according to decreasing probability. Draw a vertical line through the tree at a distance $(-\log p_1)$ from the root. This line will cut some branches of the tree forming a cut set. (Some of the branches may be cut at their far nodes). For W_1 pick the cheapest word corresponding to any branch cut. Draw another line at distance $(-\log p_2)$. For W_2 pick the cheapest word corresponding to any branch cut which is not prefixed by W_1. Continue this procedure, always picking the cheapest branch cut which is not prefixed by previously chosen words, until all M words are chosen.

For this manner of picking words, the lowest the cost will be is for perfect match, i.e., $q_k = -\log p_k$. On the other hand, the cost can be no more than this value plus the normalized cost of the costliest symbol (i.e., the length of the longest branch) which is $(-\log P_{mD})$ which we write L_D. Thus the cost of each word is bounded as follows:

$$-\log p_k \leq q_k < -\log p_k + L_D \tag{46}$$

We note that there is no possibility of equality on the right hand side since if a line passes through a node we always pick the branch on the left of the node. We may then bound the average cost as follows:

$$\bar{q} = \Sigma\, p_k q_k < \Sigma\, p_k\, (-\log p_k) + \Sigma\, p_k L_D = H + L_D$$

i.e.,,

$$H \leq \bar{q} < H + L_D \tag{47}$$

Hence,

$$e = \frac{H}{\bar{q}} > \frac{H}{H + L_D} = \frac{1}{1 + \frac{L_D}{H}} \tag{48}$$

The proof that this procedure works follows the same kind of reasoning we have already used in the equal-cost case. First, we note that if the procedure is to fail it is because we have "run out" of tree to use. However, an inspection of the method will show that if there are no more words to be chosen we have cut through the tree, i.e., the previous words chosen form a cut set. Now consider the structure function for the first p words which was defind by eq. 24:

$$S_p = \Sigma_{k=1}^{p} 2^{-q_k} \tag{49}$$

Also consider the sum of the probabilities of all messages which we may write E_{p+1} according to eq. 42:

$$E_{p+1} = \Sigma_{k=1}^{p} p_k \tag{50}$$

Since $q_k \geq -\log p_k$, we have

$$S_p = \Sigma_{k=1}^{p} 2^{-q_k} \leq \Sigma_{k=1}^{p} 2^{\log p_k} = \Sigma_{k=1}^{p} p_k = E_{p+1} \tag{51}$$

But for a cut set of p words $S_p = 1$. Hence, $E_{p+1} = 1$. This indicates that if ever the tree got used up (i.e., a cut set were formed) then all the messages would have been used up also. Therefore, we can never have a message for which there is no code word and the proof is completed. The remarks given in the last paragraph Section 3.3 as to the optimality of the procedure also apply here.

Two other methods which are possible involve the same kind of procedure except we start with the least probable message and pick either the top-most or bottom-most branch cut and continue to the most probable message. Although the author is convinced that these procedures will always work also (by trying examples), he has not been able to prove it. The difficulty is that some of the branches which might be used may be "left out", i.e., a cut set may not be formed. Hence, it is no longer true that $S_p \leq E_{p+1}$. In order to prove the procedure one must show, in some way, that if branches are left out there must be sufficient mismatching between message probabilities and word costs to allow branches for all messages.

It may be wondered whether any "reasonable" method of picking W_k such that $-\log p_k \leq q_k < -\log p_k + L_D$ might work. This is not the case. One "reasonable" method would be to choose the largest q_k possible

subject to the above constraint. But here the inefficiency in using the tree that this rule necessitates may cause it to fail. If the code words are to be chosen in order of decreasing probability, this method will fail for some cases as witnessed by the following example: Let $c_2 = 2c_1$, then $P_{m1} = 0.618$ and $P_{m2} = 0.382$. (See Fig. 3). Let $p_1 = .11$; $p_2 = p_3 = \ldots = p_6 = .09^+$; $p_7 = p_8 = \ldots = p_{15} = .056^+$. Here we see that when we get to M_1 the tree is already "used up". We may note that p_7 through p_{15} were chosen so as to use up as much of the tree as possible with messages of relatively little probability. The method would have worked, in this example, if the messages had been considered in order of increasing probability.

It seems that schemes such as these are inherently better if words are picked from M_1 to M_M rather than vice versa. All examples tried show the first way to be better except when the messages are equiprobable in which case there is no difference between the methods.

5. FANO AND BLACHMAN PROCEDURES (IA AND IB)

We see from eq. 14 that for the binary equal-cost channel the symbols should be used with equal probability. In order to achieve channel capacity it is necessary that these probabilities be constant from symbol to symbol and not merely average out to the desired values, for if the probability distribution ever differs from that given by eq. 14, the average information rate must be lower than the maximum possible rate. This is another way of saying that the probability of each symbol should be independent of what symbols preceded it. Then in the binary equal-cost case we want each symbol to have the probability one-half at any position in the code. One way of approximating this condition was described by Fano[8]. It follows:

Order the messages by decreasing probability. Separate the messages into an upper and a lower group such that the difference between the sum of the probabilities of the messages of each group is as small as possible. To all messages of the first group make d_1 the first symbol of each word and to all messages of the second group assign the first symbol d_2. Now subdivide the first group into two groups which are again of nearly equal probability. Assign d_1 as the second symbol to all words in one

of these subgroups and d_2 for the other subgroup. Do the same for the second group. Continue making subdivisions and assigning symbols in this way until each subgroup has only one member.

This procedure seems to be always more efficient than the "straight" Shannon procedure (though no proof of this has been found) but not necessarily more efficient than the Shannon procedure with modifications. No simple bound has been proved as has been in the Shannon procedure.

Now let us consider the extension to the general D symbol cost-weighted case as described by Blachman.[9] Here we divide the messages up into groups whose probabilities are close to the probabilities we want for the symbols. These groups are similarly divided into subgroups as before. More precisely let

$$E_{p1} = \Sigma_{k=1}^{p-1} p_k \tag{52}$$

and

$$V_s = \Sigma_{j=1}^{s-1} P_{mj} \tag{53}$$

We shall represent the p^{th} message by the symbol sequence $d_{a1}, d_{a2}, \ldots, d_{ar}, \ldots, d_{an}$ of n symbols, where

$$V_{ar} \leq E_{pr} < V_{a(r+1)} \tag{54}$$

and

$$E_{p(r+1)} = (E_{pr} - V_{ar})/P_{mar} \tag{55}$$

Blachman states that the following conditions hold:

$$\prod_{r=1}^{n} P_{mar} \leq p_p < \prod_{r=1}^{n-1} P_{mar} \tag{56}$$

However, the inequality on the left is not always true. Consider the following counterexample: $p_1 = .7$, $p_2 = .3$, $P_{m1} = .6$, $P_{m2} = .4$. The Blachman code for this situation is just $W_1 = d_1$ and $W_2 = d_2$. This gives $p_2 = .3 < P_{m2} = .4$ in contradiction to relation 56. No proof or counterexample has been found for the right hand inequality. If it were true we could derive the same bound as in the Shannon procedure as follows:

$$q_p = -\Sigma_{r=1}^{n} \log P_{mar} = -\Sigma_{r=1}^{n-1} \log P_{mar} - \log P_{man} \leq$$

$$L_D - \log \prod_{r=1}^{n-1} P_{mar} < L_D - \log p_p \tag{57}$$

We note that in most cases the Blachman procedure does not give a

cut set and could usually be improved upon in those cases. The Blachman procedure again seems better than the straight Shannon procedure but not necessarily better than the Shannon procedure with modifications.

6. OPTIMUM CODING (IA) - THE HUFFMAN PROCEDURE

The above methods work and seem to give us good results most of the time in terms of efficiency. We may ask, however, for a given probability distribution of the message source and a given set of costs for the channel symbols, how may we obtain the optimum code, i.e., the signal set which gives the maximum efficiency. It is to be noted that for Class I maximum efficiency is not necessarily close to one.

In obtaining the optimum code for Class I we are faced with the first significant difference between the equal-cost and the cost-weighted cases. In the equal-cost case we have a relatively simple procedure (the Huffman procedure) for obtaining the optimum code. In the cost-weighted case, on the other hand, we have discovered no simple systematic procedure, outside of trying all the possibilities, which will insure obtaining the optimum solution in the general case.

The Huffman procedure[10] is essentially a method of building the optimum signal set on the code tree by starting at the far branches and working back to the root. We now describe the method first for the binary channel.

Order the messages by decreasing probability. The last two messages have code words which are the same length and differ only in the last symbol. Assign the last symbol to each of these words. Replace these two messages by a single equivalent message whose probability is the sum of the probabilities of the messages. Reorder the messages, if necessary, to keep them in order of decreasing probability. The last two messages again have code words whose length is the same and which differ only in the last symbol. Repeat assignment of symbols, combination of the messages, and regrouping of the messages as before. Continue this procedure until only one message remains. At each step we build up the tree by adding on two more branches. These branches may represent code words or prefixes of code words.

The D symbol case is similar but we combine the D least probable

messages at a time. The exception to this is the initial step in which only m_o messages are combined, where m_o is an integer between 2 and $D - 1$, inclusive, such that $(M - m_o)/D - 1$ is an integer. For examples see Huffman.[10]

Let us show that the Huffman procedure is optimum for the binary channel case. First we assert that an optimum signal set must be a cut set, i.e., every word equal to or shorter in length than the longest code word must be either a code word, a prefix of one, or prefixed by one. This must be true of words shorter than the longest code word (for $n_k < n_M$) since if it were not true we could substitute the shorter word for a longer code word and decrease the average cost. It must also be true for words such that $n_k = n_M$, since if it were not we could drop the last letter of any code word which had the same first $n_M - 1$ letters as the word in question. This new code word would still obey the prefix rule and the average cost would again be decreased.

Secondly, we have the order rule which asserts that the lengths of code words must increase as the probabilities of the corresponding messages decrease; i.e., $n_1 \leq n_2 \leq \ldots \leq n_M$ (where the messages are ordered by decreasing probability). If , for instance, $n_1 \geq n_2$, we could interchange W_1 and W_2 and reduce the average cost. By observing the tree structure, we see that these two assertions make it necessary that $n_M = n_{M-1}$.

If the optimum code is such that there are more than two messages whose code word lengths equal n_M, the maximum code word length, there must be an even number of them because of the cut-set nature of the signal set. We may then arbitrarily select a code for which the W_M and W_{M-1} have the same prefix, since if there exists any optimum code for which this isn't true, we can rearrange that code, with no increase in cost, to make it true.

We then assert that an optimum code for the new message ensemble, E', which is obtained by replacing M_M and M_{M-1} by a new message, M'_{M-1}, whose probability is $p_M + p_{M-1}$, will give an optimum code for the original ensemble, E, by adding the two symbols, in turn, to W'_{M-1} to get W_M and W_{M-1}. To prove this assume there is an optimum code for E' with cost $\bar{n}'_{opt}$ which when expanded in the above manner does not give an optimum

code for E. This means

$$\bar{n} = \bar{n}'_{opt} + p_M + p_{M-1} > \bar{n}_{opt} \tag{58}$$

But then we could take an optimum code for E and reduce it to get a cost

$$\bar{n}'_2 = \bar{n}_{opt} - p_M - p_{M-1} < \bar{n}'_{opt} \tag{59}$$

This contradiction proves our assertion. The same reasoning holds for each additional regrouping and thus these successive regroupings must determine an optimum code.

We say an optimum code and not the optimum code because of the many permutations of letters in the words which are possible. First of all, at each of these regroupings there is an arbitrary choice of assignment of channel symbols. So if there are R regroupings there are 2^R variations possible. In the second place, if there are more than two words of a given length they can be interchanged so that successive messages do not necessarily have the same prefix. Thirdly, for equiprobable messages and equivalent messages, code words can be interchanged. This would add to the number of variations so far considered only if there were a different number of letters in some of these words. We may say that the Huffman procedure determines an admissible class of codes and all other optimum codes may be obtained by a suitable rearrangement of letters.*

In the D symbol case the optimum signal set must be a cut set or almost a cut set. We observe that a cut set for the D symbol tree contains M members, where

$$M = D + a(D-1) = b(D-1) + 1 \tag{60}$$

where a and b are integers.

We can see that this is so by observing that the set of all D symbols is a cut set and any other cut set may be obtained by successively replacing one branch by D others. If the number of messages is one of those given by eq. 60, we must have a cut set for the optimum signal set. The same reasoning as in the binary case applies to words of smaller

---

*Otter [20] gives a discussion which bears on the number of signal sets possible with the same cost.

length than $n_{M(opt)}$. For words equal in length to n_M we must also have for each fan either all D of the branches used as code words ar none of the branches used as code words. This is true since eq. 60 is satisfied and we see by observing the tree that if it were not true we could regroup the code words so as to make it true and reduce some of the code word lengths in doing so.

If eq. 60 is not satisfied, say

$$M = b(D - 1) + m_o \qquad \text{where} \quad 1 < m_o < D \tag{61}$$

then the optimum set must be some set which is a cut set except for $D - m_o$ "missing" branches of length n_M. The same reasoning as above applies where $n_k < n_M$. From eq. 60 we know that the number of missing branches must be $I(D - 1) - m_o + 1$ where I is some integer. But I must equal one for if it were greater we could "consolidate" the code tree as suggested above and reduce some of the code word lengths.

Among all the optimum codes, which include those which vary in the placing of the missing branches, the Huffman procedure restricts itself to that class which contains all the missing branches in one fan. The same arguments apply as in the binary case except that now m_o of D messages are combined in the first step and the number of messages is reduced by D - 1 at each succeeding step. For $D > 2$ there are more possibilities for variations on the optimum code if eq. 60 is not satisfied.

7. MESSAGE ENSEMBLES WHICH MATCH POORLY (IA)*

Now that we have found a procedure for determining the optimum code we may ask how good the optimum code is for various classes of message ensembles. We may first inquire whether there are any ensembles for which the Shannon bound on cost is approached by the optimum code. A simple example of just such an ensemble is one for which $M = 2$ and $p_1 \to 1$. Then each message must be represented by a single symbol of a binary channel. We then have $\bar{q} = p_1 + p_2$ which approaches one. $H = -(p_1 \log p_1 + p_2 \log p_2)$ approaches zero. Therefore, $\bar{q} - H = \delta \to 1$

Thus the difference between average cost and entropy, which we write δ, approaches the maximum value given by the Shannon bound for this example. We may extend the result to the case of the D symbol

---

*The material of this Section is due mainly to P. Elias.

channel and for which H is arbitrarily large. The nasty distribution which will prove the point has one large probability approaching one and a lot of small probabilities. Specifically, let

$$M = 1 + (D - 1)D^{N^2}$$

$$p_1 = 1 - 1/N \tag{62}$$

$$p_k = \frac{1}{N(M - 1)} \qquad \text{for } k = 2,3,\ldots,M$$

We may determine that the optimum code has

$$n_1 = 1$$

$$n_k = N^2 + 1 \tag{63}$$

Then

$$\bar{q} = \bar{n} \log D = \log D\left(1 - \frac{1}{N} + \frac{N^2 + 1}{N}\right) \tag{64}$$

Therefore, $\bar{q} \to (\log D)(1 + N)$ as $N \to \infty$. But

$$H = -p_1 \log p_1 + \frac{1}{N}\log (M-1)N$$

$$= -p_1 \log p_1 - \frac{1}{N} \log N + \frac{1}{N} \log(D-1) + \frac{N^2}{N} \log D \tag{65}$$

Hence,

$$H \to N \log D \quad \text{as} \quad N \to \infty. \tag{66}$$

Hence,

$$\delta = \bar{q} - H \to (N + 1 - N)\log D = \log D \tag{67}$$

as $N \to \infty$ and $H \to \infty$. Of course since $H \to \infty$, $e \to 1$.

The large δ in the above examples seems to result in some way from the one large probability. One may then wonder whether δ would be arbitrarily small for the optimum code if the maximum probablity, p_{max}, were sufficiently small. That this is not true we can show by counter-example. Consider M equiprobable messages. Then $p_{max} = 1/M$. Suppose $M/D^a = r$ where $0 < r < 1$ and a is an integer. Since M is not a power of D we know there must be some mismatching and some $\delta = \delta_m > 0$. If we now consider $M' = MD^b$ (b is an integer), we see that the same code as above will be used to divide the M' messages into M groups. Onto these M words all the D^{b-a} combinations of b-a letters will be perfectly matched with the information provided by each digit being log D but the

same mismatch occurs for the original code so that δ remains δ_m. This is true for any b and hence for b approaching infinity and p_{max} approaching zero. As an example, consider D = 2 and $M' = 3x2^N$. Then Huffman coding will lead to three equiprobable groups of messages which must be assigned code words d_1, d_2d_1, and d_2d_2. Thus for the preliminary choice among the three groups $\bar{n} = 5/3 \doteq 1.667$, H = 1.586 and $\delta = 0.081 > 0$ while p_{max} approaches zero.

The worst case for the binary channel and equiprobable messages occurs when $(M - 2^S)/2^S$ approximately equals 0.385, where S is the largest integer such that $2^S \leq M$. δ is the largest for this situation and equals about 0.086. What the largest value of δ is in this situation for D > 2 is an unanswered question.

The question may then be asked for a given p_{max} what is δ_{max}. Elias has determined an upper bound for $\delta_{max}(p_{max})$ for the binary channel. It is

$$\delta_{max} \leq \delta_o - \frac{\delta_o - 2p_{max}}{2 - \delta_o} \tag{68}$$

where $\delta_o = 2 - (2 - 2p_{max})^c$, $\quad c = 1/2$

This bound approaches 0.172 as p_{max} approaches zero. We note from the previous example that for the binary channel we can state that the following relation holds: $0.086 \leq \delta_{max}(0) \leq 0.172$. What the actual upper bound is for δ_{max} as p_{max} approaches zero is another unanswered question.

We have described above some examples in which the best coding gives as poor as possible a match between the probablitity distribution of the message ensemble and the channel. It may be remarked that for the average distribution, which is to say a distribution picked at random in some way, the match will most likely be much better. More of this later.

8. OPTIMUM CODING (IB)

8.1 GENERAL COMMENTS

Let us now discuss the optimum code for Class IB. First we shall show that the optimum signal set must be a cut set or almost so. Suppose W_M for the optimum code has cost q_M. Then any word with cost less than q_M must be used as was shown for the equal-cost case. But because of the unsymmetrical nature of the tree this does not mean that all fans are filled (i.e., that all branches in a fan are used). There may be some fans filled if their roots are at a distance from the most costly word given by $q_M - q_r \leq L_D$, where q_r is the cost at the root of the fan. Of course, in any case, at least two branches, if any, must be used on a fan. Again, as in the equal-cost case, the binary channel is the only case in which the optimum signal set has to be a cut set. Obviously the order rule still holds, i.e.,

$$q_k \geq q_j \quad \text{for } k > j \tag{69}$$

What is lost is the simplicity of the relationship between n_k and q_k. Two words of the same length may have quite different costs. Therefore, we can no longer say that the last two words, W_M and W_{M-1}, have the same length and there is no simple extension of the Huffman procedure.

What we can do in general is: 1) apply the extended Shannon procedure and then reduce the costs of some of the words by making a cut set if we do not already have one; 2) use the Blachman procedure with similar modifications; 3) use the Huffman "Shakedown" procedure; 4) try by trial and error to match costs to probabilities.

8.2 THE HUFFMAN "SHAKEDOWN" PROCEDURE

The Huffman "Shakedown" procedure* is based on an extension to the order rule. Suppose at a given node, A, we sum the probabilities of all code words whose branch paths pass through A. Call this sum p_A. Take a second node, B, which is on a different path. Similarly, find p_B. Then for minimum cost it must follow that $p_A > p_B$ if B is to the right of A on the tree and vice versa. If this were not true we could interchange the code structure branching out from node A with that from node B.

---

*D.A. Huffman, unpublished results.

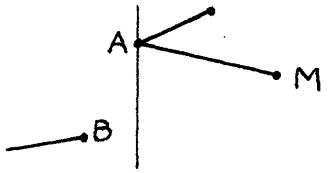
The saving in cost would be $|p_A - p_B| x |q_A - q_B|$, where q_A and q_B are costs at A and B, respectively.

The procedure is a repeated use of this rule. For D = 2 one may start with a cut set containing M branches. Initially, assign the words by the simple order rule. Then one looks for violations of the extended order rule and rearranges the code structure to satisfy the rule. No single detailed method of doing this which will gaurantee obtaining the optimum solution when all node pairs are satisfied has been found.

8.3 EQUIPROBABLE MESSAGES

The special case of equiprobable messages ($p_k = 1/M$) is solved. First define a vertical signal set as a set of branches cut by a vertical line drawn at some cost-distance, y, from the root of the tree. This line may cut a branch at either end. We then assert that an optimum signal set for the binary channel is a vertical cut set with M branches.

Consider the costliest word. This is labeled M in the drawing below:



Consider any other code word, say B, which is not cut by a vertical line through branch M. Then $q_B < q_A$. But $p_A = \frac{2}{M} > p_B = \frac{1}{M}$. This violates the extended order rule. Therefore, all code words must lie on a vertical set. But since the optimum code must be a cut set, the solution is a vertical cut set with M branches. We may find this cut set by moving an imaginary vertical line to the right from the root until M branches are cut. If there is a set of nodes with the same cost, q_c, such that the number of branches cut for $y < q_c$ is less than M but the number of branches cut for $y > q_c$ is more than M, there is a certain ambiguity as to which cut set of M to choose. However, these are obviously equivalent from the cost point of view.

For the D symbol channel the argument that the optimum code is on a vertical set is true, a fortiori, since $p_A \geq \frac{2}{M}$. But now the solution

is not necessarily a cut set. However, for minimum cost, it is clear that we want the M least costly words from some vertical set. Let us denote such a set by M(y), where y is the cost corresponding to the leftmost line which cuts the set. Let us order the M(y) by increasing y. If the y line cuts more than one node at a time we consider first the signal set with first one of the fans from these nodes as being in the set, written $M(y_1)$, then two fans from the node, written $M(y_2)$, and so on.

The reason that we do not always choose the first set is that we might reduce the cost by replacing some of the most costly words and the least costly word by some of the words having the least costly word as a prefix. It can be seen that the cost decreases continuously up to a certain y and then continuously increases. It is possible, as is always true for the binary channel, that the least costly M(y) is the first one. In any case, we move the imaginary vertical y line from left to right until the cost for M(y) starts increasing. The optimum signal set is then the one before the cost starts to increase. It should be clear that the optimum value for y is less than $y_o + L_D$, where y_o is the first y.

For this optimum M(y) the following relation must hold:

$$q_M - q_1 \leq L_1 + L_2 \tag{70}$$

where L_1 is an abbreviation for $(-\log P_{m1})$, etc.

Relation 70 must hold, for if it did not, we could replace the most costly word and the least costly one by two words made up of the least costly word and symbols d_1 and d_2, respectively. The saving in cost would be $q_M + q_1 - (L_1 + L_2) > 0$.

For D = 3 we may say that the optimum M(y) is the first one for which relation 70 holds. In any case we may bound the total cost as follows:

$$q_k \leq \log M + L_1 + L_2 \qquad k = 1,2,\ldots,M \tag{71}$$

Hence,

$$\bar{q} \leq L_1 + L_2 + \log M \tag{72}$$

This is an improved bound over the Shannon bound if $L_1 + L_2 < L_D$.

8.4 D = 2, M = 3

Let us consider the problem of finding the optimum code for the binary channel when there are only three messages. Let $r = c_2/c_1$. We must consider two cases; I) $1 \leq r \leq 2$ ($r \geq 1$, since $c_2 \geq c_1$ by convention) and II) $r \geq 2$. There are two cut sets possible for M = 3. These give code 1 (0, 10, 11) -- written in the binary number alphabet for simplicity -- and code 2 (1, 00, 01). If the matching of messages to signal words follows the order rule, then there is only one way to assign signal words for code 1. This is the order (0, 10, 11) with costs given by c_1 (1, 1+r, 2r). However, for code 2 there are two ways: 1) (1, 00, 01) with costs c_1(1, 2, 1+r) for $r \leq 2$, and 2) (00, 1, 01) with costs c_1(2, r, 1+r) for $r \geq 2$.

For case I we see by the generalized order rule that we should use code 1 for $p_2 + p_3 \leq p_1$ and code 2 for $p_2 + p_3 \geq p_1$. For $p_2 + p_3 = p_1 = 0.5$, there is no difference in cost. For case II we see that the cost for code one is $\bar{w}_1 = c_1(p_1 + (1+r)p_2 + 2rp_3)$ and for code 2 we see that $\bar{w}_2 = c_1(2p_1 + rp_2 + (1+r)p_3)$. Therefore, we should use code 1 or code 2 according as $p_2 + (r-1)p_3$ is less than or greater than p_1, respectively. For case II we note that the answer depends on r.

8.5 DIFFICULTY OF SIMPLE SOLUTION IN GENERAL

For M = 4 there are five codes to consider. These fall in three cases depending on the range of r. We have found no simple procedure for determining which code is optimum except to compare all possibilities. As M increases the complexity increases greatly. There are many more codes for a given M than in the equal-cost case. To get a simple procedure to determine the optimum code we have to reduce the problem for a large M into a simpler problem or in some other way describe an algorithm which greatly reduces the procedure of examining all possible codes. The analysis of a program for a digital computer which would examine all codes could prove instructive. Linear programming procedures might also prove helpful in this problem.

8.6 LOWER BOUND ON $S = \Sigma 2^{-q_k}$

The Kraft inequality tells us that the code words cannot be too cheap in a certain sense. We also know that the code words do not have to be too expensive and can use this fact to lower bound the function $S = \Sigma_{k=1}^{M} 2^{-q_k}$. From eq. 46 we know for the Shannon code: $q_k < -\log p_k + L_D$. Hence,

$$S = \Sigma 2^{-q_k} > \Sigma p_k 2^{-L_D} = P_{mD} \tag{73}$$

Together with the original Kraft inequality we then have:

$$P_{mD} < \Sigma 2^{-q_k} \leq 1 \tag{74}$$

For the equal-cost case this reduces to:

$$1/D < \Sigma D^{-n_k} = \Sigma 2^{-q_k} \leq 1 \tag{75}$$

We may inquire as to whether these lower bounds apply to the optimum code. For the equal-cost case the answer is "yes", with the following simple argument to prove it. Suppose relation 75 did not hold. Then for some b

$$q_b \geq \log D - \log p_b \tag{76}$$

We can then shorten word W_b by one symbol and form word W_b' and still have

$$q_b' \geq -\log p_b \tag{77}$$

Then

$$S' = \Sigma 2^{-q_k'} = S + 2^{-q_b'} - 2^{-q_b}$$

$$= S + 2^{-q_b}(D - 1) \leq 1/D + (1/D)(D - 1) = 1 \tag{78}$$

Therefore, this new code is proper, since it obeys the Kraft inequality, and we have a contradiction. This proves that $S > 1/D$ for the optimum code.

We can show that this result holds for the cost-weighted case as well. Consider the optimum signal set. We shall show that if $S \leq 1/D$, there are unused branches whose cost is less than q_M and therefore the signal set is not optimum in contradiction to the hypothesis. We know that the optimum signal set is a cut set except for some "left-over" branches on the fringes of the tree. We could form a cut set by adding on to all the branches which are used, all the branches cut by the line $y = q_M$ which are not used and do not have prefixes which are used. Let us suppose there are L of these "left-over" branches. We define S_{Lo} by

$$S_{Lo} = \Sigma_{i=1}^{L} 2^{-q_i} \tag{79}$$

where q_i are the costs of the left-over branches.

Then

$$S_{Lo} + S = 1 \tag{80}$$

Now every left-over branch must be in a group with no more than D - 2 left-over branches. We assert that for every such group there must be at least one used code word. (A used code word is a word in the signal set). This is true since there is at least one branch in the group which is either a used code word or prefixes a used code word. If not a used code word, there is another group for which at least two branches are used code words or prefixes. Since for each branch which is a prefix there exists at least two used code words, we see that there must be at least one used code word for each group which has some left-over branches. But since

$$2^{-q_i} \leq 2^{-q_M} \tag{81}$$

we have

$$S_{Lop} \leq (D - 2)2^{-q_M} \tag{82}$$

where $S_{Lop} = \Sigma\, 2^{-q_i}$, summed over all i in any particular group.

Now suppose the left-over branches occur in G groups. Then

$$S_{Lo} \leq G(D - 2)2^{-q_M} \tag{83}$$

But there must be at least G used branches, hence

$$S \geq G2^{-q_M} \tag{84}$$

Hence,

$$S_{Lo} \leq (D - 2)S \tag{85}$$

But if $S < 1/D$, then

$$S_{Lo} < (D - 2)\frac{1}{D} \tag{86}$$

Hence,

$$1 = S_{Lo} + S < (D - 2 + 1)\frac{1}{D} = \frac{D - 1}{D} < 1 \tag{87}$$

which is a contradiction. Therefore, the assumption that $S < 1/D$ for an optimum signal set is impossible and the opposite assertion that $S \geq 1/D$ is proved.

9. MESSAGE ENSEMBLES WHICH MATCH POORLY (IB)

We may now inquire about message ensembles whose probability distributions will cause trouble in trying to code efficiently. First we note that coding may be inefficient for the binary channel in which one symbol is much more costly than the other. Consider equiprobable messages. For M = 2 we have

$$\bar{q} = (1/2)(L_1 + L_2), \qquad H = \log 2 = 1 \tag{88}$$

If we let $L_2 \to \infty$ and $L_1 \to 0$, then

$$\delta = \bar{q} - H \to \frac{L_2}{2} - 1 \doteq \frac{L_2}{2} \tag{89}$$

In the general case we may pick L_2 so large that, for a given M, the following code is optimum:

$W_1 = 1$	$q_1 = L_2$
$W_2 = 10$	$q_2 = L_2 + L_1$
$W_3 = 100$	$q_3 = L_2 + 2L_1$
...	...
$W_{M-1} = 1000...00$ (M-2 0's)	$q_{M-1} = L_2 + (M-2)L_1$
$W_M = 0000...00$ (M-1 0's)	$q_M = (M-1)L_1$

For the above code

$$\bar{q} = (1/M)(L_2 + L_1 + L_2 + 2L_1 + \ldots + L_2 + (M-1)L_1)$$

$$= \frac{1}{M}L_2(M-1) + \frac{1}{M \times M}(M-1)L_1 \tag{90}$$

For large enough M

$$H = \log M > H_o \quad \text{for any preassigned } H_o$$

and

$$\delta = \bar{q} - H = \frac{M-1}{M}L_2 + \frac{M-1}{2}L_1 - \log M \doteq L_2 + \frac{M}{2}L_1 - \log M$$

and if $L_2 \to \infty$

$$\delta \doteq L_2 - \log M \doteq L_2 \tag{91}$$

Thus for the binary channel we can find distributions for which $\delta \doteq L_2$. It is obvious that we cannot expect to find distributions so

bad that $\delta \doteq L_D$ for the general D symbol channel. This is so because if the L_D is made large enough, we could do better merely by disregarding it and using only D - 1 symbols.

If we throw out the D^{th} symbol we can still apply the extended Shannon procedure to the D - 1 symbol channel and obtain

$$\bar{w} < \frac{H + L_{D-1}}{C'} \tag{92}$$

where C' is the capacity of the D - 1 symbol channel.

But now

$$e = \frac{H}{\bar{w}C} = \frac{H}{\bar{w}C'} \times \frac{C'}{C} \tag{93}$$

Then

$$e > \frac{H}{H + L_{D-1}} \times A \qquad \text{where } A = C'/C \tag{94}$$

If L_D is very large compared to L_{D-1}, A will be close to one. We note that

$$\Sigma_{j=1}^{D} 2^{-Cc_j} = 1 \qquad \text{and} \qquad \Sigma_{j=1}^{D-1} 2^{-C'c_j} = 1 \tag{95}$$

Let

$$C_A = C - C' \tag{96}$$

Then

$$\Sigma_{j=1}^{D-1} 2^{-C'c_j} = \Sigma_{j=1}^{D-1}\left(2^{C_A c_j}\right)\left(2^{-Cc_j}\right) \geq 2^{C_A c_1} \Sigma_{j=1}^{D-1} 2^{-Cc_j} =$$

$$2^{C_A c_1}(1 - P_{mD}) \tag{97}$$

Hence,

$$C_A < \frac{1}{c_1} \log \frac{1}{1 - P_{mD}} \tag{98}$$

Then

$$A = C'/C = \frac{C - C_A}{C} = 1 - \frac{C_A}{C} > 1 - \frac{\frac{1}{c_1} \log \frac{1}{1-P_{mD}}}{C} \tag{99}$$

Therefore, if $L_D \to \infty$, $P_{mD} \to 0$ and $A \to 1$.

It is an open question as to the efficiency of coding when there is one large probability close to one and many small probabilities. If the small probabilities are all equal, it seems intuitively clear that the coding will be, in some sense, more efficient than for the equal-cost case since one should be able to match these probabilities better. Just how much better is not known. It seems reasonable that we could not have the same δ approaching L_D for the large M situation for the cost-weighted case that we had in the equal-cost case. This is espedially true if the costs

are incommensurate. For the incommensurate case the nodes on the tree seem to spread out more and more uniformly along the horizontal cost axis as the tree branches out.*

It seems similarly true that the lower bound on δ for the equiprobable message case is reduced. This should be even more true as p_{max} approaches zero but there is still probably some greatest lower bound which is greater than zero.

10. CODING OTHER THAN PROPER CODING

10.1 SARDINAS-PATTERSON RULE FOR UNIQUE DECOMPOSABILITY

We have seen that a sufficient condition on the signal set for the code to be uniquely decomposable is that it obey the prefix rule. That this is not a necessary condition may be seen by considering the signal set $W_1 = 0$, $W_2 = 01$. Sardinas and Patterson[11] give a procedure for determining whether a given signal set is uniquely decomposable. It follows:

The signal set itself is called segment class zero, written Seg_0. If one word prefixes another, the remainder of the second word is in segment class one, written Seg_1. If one word in Seg_0 prefixes a word in Seg_1, or vice versa, the remainder of the longer word is placed in Seg_2. In a like fashion Seg_{i+1} is generated from Seg_i and Seg_0. The rule then states that the code is uniquely decomposable if no word in Seg_0 appears in Seg_i, $i \geq 1$.

Since the maximum word length is bounded (n_{max}) the algorithm will give a definite answer in a finite number of steps. There are two possibilities for success: 1) Seg_i is empty for some i and hence, Seg_j is empty for all $j \geq i$; 2) $Seg_j = Seg_i$ (Seg_i is not empty) for some $j \geq i$. For the latter case the segment classes repeat in a periodic fashion, the period being $j - i$. In case 1) the code has local decodability (a term invented by Schutzenberger), in case 2) it does not. A code has local decodability if there exists $L < \infty$ such that for any given m we can uniquely decode the first $m - n_{max}$ symbols into messages once the first L symbols are known.

---

*See Gnedenko and Kolmogorov[21] for mathematical details.

10.2 AN INEQUALITY FOR PROPER SIGNAL SETS

We shall now find necessary and sufficient conditions on the set of words in a signal set such that there exists a proper signal set with the given set of costs. We do this first for D = 2. The cost of a word is characterized by the number of d_1 and d_2 symbols in it. Let the number of code words with x d_1's and y d_2's be given by N(x,y). The total number of words with this composition is $C_{x,y}$

$$C_{x,y} = \frac{(x+y)!}{x!y!} \tag{100}$$

Hence

$$N(x,y) \leq C_{x,y} \tag{101}$$

But if any code word with composition (x',y') is used such that $x' \leq x$ and $y' \leq y$, then we cannot use all $C_{x,y}$ words of composition (x,y) as code words. Specifically, there are $C_{x-x',y-y'}$ words of composition (x,y) which are prefixed by the word of composition (x',y') and hence cannot be used simultaneously with it. Similarly, if we have $N(x',y')$ words of composition (x',y') we may not use $N(x',y')C_{x-x',y-y'}$ words of composition (x,y). In the same way, if there are any words of composition (x'',y'') such that $x'' \leq x$ and $y'' \leq y$, we cannot use $N(x'',y'')C_{x-x'',y-y''}$ additional words of composition (x,y). We now consider all words which are prefixes of words of composition (x,y). A necessary condition on the N(x,y) for a prefix code is then given by

$$\Sigma_{i=0}^{x} \Sigma_{j=0}^{y} N(x-i,y-j)C_{i,j} \leq C_{x,y} \tag{102}$$

This must hold simultaneously for all (x,y) compositions. It should be clear that if it holdsfor some (x,y) it also holds for any (x',y') such that $x' \leq x$ and $y' \leq y$. Also, if it holds for some (x,y) it also holds for any (x+i,y+j) as long as $N(x+i,y+j) = 0$ for all $i+j > 0$. Then if for a given signal set the maximum $n = x + y$ for a code word is n_{max}, the conditions 102 reduce to the $n_{max} + 1$ inequalities such that $x + y = n_{max}$.

We note that if the equality holds for some (x,y) then the tree is full for that (x,y). If the equality holds for all (x,y) such that $x + y = n$ then the tree is full and the signal set is a cut set.

We may see that conditions 102 are sufficient in the following way. Make code words for all (x,y) such that n = x + y = 1. Then make code words obeying the prefix condition for all (x,y) such that n = x + y = 2. Continue this procedure. If it fails, at least one of the conditions 102 must not be satisfied.

We may easily extend these results to the D symbol case and obtain necessary and sufficient conditions corresponding to conditions 102 as follows:

$$\Sigma_{i_1=0}^{x_1} \Sigma_{i_2=0}^{x_2} \cdots \Sigma_{i_D=0}^{x_D} N(x_1-i_1, x_2-i_2, \ldots, x_D-i_D) C_{i_1, i_2, \ldots, i_D} \leq C_{x_1, x_2, \ldots, x_D} \tag{103}$$

where $N(x_1-i_1, x_2-i_2, \ldots, x_D-i_D)$ is the number of code words with x_1-i_1 d_1's, etc. and $C_{x_1, x_2, \ldots, x_D}$ is given by[18]

$$C_{x_1, x_2, \ldots, x_D} = \frac{(x_1 + x_2 + \ldots + x_D)!}{x_1! x_2! \ldots x_D!}$$

10.3 ADMISSIBILITY OF PROPER SIGNAL SETS

One may ask whether any of these non-prefix codes could be more efficient than the best prefix code. This is answered[6] in the negative in the equal-cost case by a simple argument. First we show that the Kraft inequality is a necessary condition for a uniquely decomposable code. We employ a proof by contradiction. Suppose it is not so. Then there exists a good code such that

$$\Sigma 2^{-q_k} = 1 + \delta \qquad \delta > 0 \tag{104}$$

Then consider a message probability distribution such that $p_k = 2^{-q_k} - \epsilon_k$, where $\epsilon_k > 0$ and $\Sigma \epsilon_k = \delta$. Then

$$\bar{q} = \Sigma p_k q_k < \Sigma q_k 2^{-q_k} \tag{105}$$

and

$$H = -\Sigma p_k \log p_k = -\Sigma(2^{-q_k} - \epsilon_k) \log(2^{-q_k} - \epsilon_k) \tag{106}$$

but

$$-\log(2^{-q_k} - \epsilon_k) > -\log 2^{-q_k} = q_k \tag{107}$$

Hence, $$H > \Sigma(2^{-q_k} - \epsilon_k)q_k = \bar{q} \qquad (108)$$

But this contradicts the Fundamental Theorem. Q.E.D.

We know, however, that the Kraft inequality is a sufficient condition that there exists a prefix code in the equal-cost case with a given $q_k = n_k \log D$. Therefore, since the optimum code must satisfy the Kraft inequality, there exists a prefix code as good as the optimum code. I.e., the class of prefix codes is admissible when considering efficiency for the equal-cost case. It is believed that the same result should hold for the cost-weightd case but this has not been proven yet. One stumbling block is that the Kraft inequality is no longer sufficient to insure a prefix code with a given set of q_k. Consider, for example, costs $q_1 = L_2$ and $q_2 = 2L_2$. Then $2^{-L_2} + 2^{-2L_2} = P_{m2} + P_{m2}^2 < 1$ and the Kraft inequality is satisfied. But the only signal set will be $W_1 = d_2$ and $W_2 = d_2 d_2$, which is clearly not a prefix code.

We could prove the hypothesis for the cost-weighted case if we could prove a somewhat stronger one which is probably true. Namely, any signal set which passes the Sardinas-Patterson test can be changed into a signal set which is proper, if it is not already so, merely by rearranging some of the symbols in the words. This proper signal set would then have the same cost structure. We could also prove the hypothesis by proving the following statement, which is also probably true: if one of the conditions 102 is not satisfied, the signal set fails the Sardinas-Patterson test.

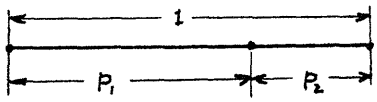
If we could prove the admissibility hypothesis in the general case, one might be tempted to say that we may as well forget about any codes but proper ones, since proper codes seem simpler to use. However, some codes already in use* are non-proper and, besides, efficiency is not the only criterion for a good code, as we have seen. Other considerations (see, e.g., Schutzenberger[12] and Laemmel[19]), including resistance to noise and facility of synchronization, might outweigh the efficiency consideration.

---

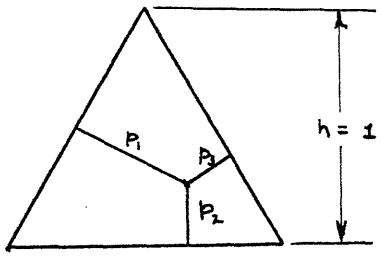
*An example is spoken English. George Miller of Harvard pointed out that the following pair of sentences could be confused due to the fact that spoken words may have the same sound prefix: 1) The good can decay many ways; 2) The good candy came anyways.

11. GEOMETRIC PICTURE OF THE CODING PROBLEM

The discussion of Section 9 leads us to consider another way of looking at the coding problem as described by Mandelbrot.[13] It is a geometrical picture for visualizing the problem of picking a signal set to match a given message set. First we describe a geometric interpretation for a set of probabilities. For M = 2 the probability distribution can be represented by a point on a line segment of unit length. See the following figure:

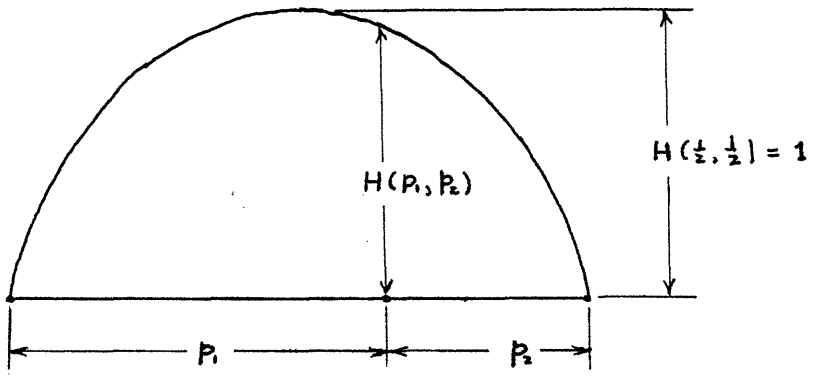


The point divides the line segment into segments of length p_1 and p_2. Obviously, there is a one to one correspondence between any point on the line and any probability distribution. Similarly, for M = 3, we can represent any probability distribution by a unique point inside an equilateral triangle of unit altitude, and vice versa. See the following figure:



This follows from the geometric fact that the sum of the distances of a point inside an equilateral triangle to the sides is equal to the altitude of the triangle. These distances then represent the three probabilities. Similarly, in three dimensions, we can think of a probability set with four members as being represented by the distances from a point inside a regular tetrahedron to its faces. In general, a set of M probabilities can be represented by the distances from a point in an "equilateral hypertriangle" of dimension M-1. The co-ordinates used for the representation are known as barycentric co-ordinates.

The entropy, $H = -\Sigma\, p \log p$, of a probability distribution is a function only of the p's. We may then think of a surface in M dimensional space whose distance is H from the point in the hypertriangle given by the probabilities. The distance is measured along an axis perpendicular to the other M - 1 axes. For example, for M = 2, we see that the H surface is a convex curve. See the following figure:



For M = 3, we see (Fig. 2) that the H surface is a dome-shaped affair like the roof of Kresge auditorium at M.I.T. For M >3 the visualizing becomes impossible except in an abstract way.

Finally, we want to include the cost of coding in the geometrical picture. Let us consider a given code with its set of costs, q_k. The cost of using this code is $\bar{q} = \Sigma\, p_k q_k$ which is a linear function of the probabilities. Therefore, the $\bar{q}$ surface, where $\bar{q}$ is measured along the same co-ordinate as H was, will be a hyperplane. Here we note that the probabilities and costs are no longer ordered by decreasing probability or increasing cost, in general.

The distance between the cost hyperplane and the H surface is $\delta = \bar{q} - H$ and is a measure of the inefficiency of the code. Wherever the plane is tangent to the H surface the code is a perfect match for that set of probabilities, i.e., the efficiency is one.

There is a cost hyperplane for each code. The optimum code for a given message probability set will be that one whose cost hyperplane is closest to the H surface at the point corresponding to the probability set.

If we consider the locus of the optimum cost plane for each point of the hypertriangle, we get a cost surface consisting of sections of planes. For example, consider M = 3 and observe Fig. 2 in which a section of the cost surface is shown. For M = 3 the cost surface is just made up of sections of ordinary planes.

It may be shown[13] that both the cost surface and the H surface are convex. Therefore, a cost hyperplane can be tangent to the H surface at only one point. For the binary channel any optimum signal set must be a cut set. But for a cut set there is a set of probabilities for which $H = \bar{q}$, namely $p_k = 2^{-q_k}$. Therefore, for the binary channel, all sections of the cost surface are tangent at one point. This is not true for D > 2 where some optimum codes need not be cut sets and some sections of the cost surface will not touch the H surface.

The most inefficient codes are seen to be those for which two or more sections of the cost surface intersect. This is true since by the convex nature of the H surface, any movement away from an intersection must decrease $\delta = \bar{q} - H$.

We now see that it is, in general, better to have as small a D as possible as far as the possibility of efficient coding is concerned. This is true, because for a given M, the most possible combinations of optimum codes will occur for the smallest D, namely D = 2. Therefore, with more segments the cost surface matches the H surface more closely. For the same reason the cost-weighted channel is, in general, more efficient than the equal-cost channel. There are a greater number of codes in the cost-weighted case because permutations of channel symbols give codes with different costs, unlike the equal-cost case.

12. EXAMPLE: CODING ENGLISH LETTERS - THE MORSE CODE

As an example of several of the coding properties we have been talking about, let us consider the problem of encoding the letters of the English alphabet. We take for our message ensemble the 26 letters plus the word space. A set of probabilities for these 27 messages is given in Table 1. For these probabilities H = 4.03 bits/letter. The Morse Code channel, as described in Section 1.3, has a capacity of C = 0.590 bits/tap. The average time for code words in the Morse Code is 8.661 taps/letter. Therefore,

$$e = \frac{H}{\bar{w}C} = \frac{4.03}{8.661 \times .591} = 0.790 \qquad (109)$$

Another code, the American Morse Code, has another space symbol of duration 2 taps which is used between dots and dashes of some code words for the letters. This channel, which has been discarded because of the confusion that arose in trying to distinguish among the space symbols, naturally has a higher capacity. The characteristic equation, using the constraint that no two space symbols may follow one another is

$$2^{-2C} + 2^{-3C} + 2^{-4C} + 2^{-4C} + 2^{-5C} + 2^{-6C} + 2^{-7C} + 2^{-9C} = 1 \qquad (110)$$

The solution of eq. 110 is C = 0.736 bits/tap. This assumes, of course, that the tap is the same unit of time in each calculation. The average time per letter for the American Morse Code has been calculated to be 7.765 taps/letter. This is about 10 per cent faster than the International Morse Code (Casper[3] says about 5 per cent) and gives an efficiency of 0.705.

Could the International Morse Code be made more efficient? If one restricts oneself to using the spaces as letter and word ends and to keeping the maximum number of symbols per letter at four, then the only consideration required to obtain optimum efficiency under these restrictions is to use all of the 27 least costly sequences of dots and dashes of length four or less so that the order rule is obeyed. The Morse Code obeys this rather well except that the letter "O" is given too long a sequence and the sequences 0011 and 0101 are not used (0 = dot, 1 = dash). The optimum code, given as code 1 in Table 1, would have a cost of 8.260 bits/letter, the largest part of the improvement coming from correcting the code word for "O". This would give an efficiency of 0.827.

We may now ask what efficiency could we obtain if we consciously strove to match cost and probability according to the optimizing condition of eq. 14. To apply this condition to the Morse Code channel as given would require a complicated analysis which might include spaces in letters. Let us rather consider a simplified Morse Channel with just the dot and dash. The capacity for this channel is the solution of

$$2^{-2C} + 2^{-4C} = 1 \qquad (111)$$

which is C = (1/2)(0.695) = 0.347 bits/tap.

Text continued on page 47.

RANK	MESSAGE	PROBABILITY*	CODE 1	CODE 2	CODE 3
1	word space	0.200	**	000	00
2	E	.105	0	0010	010
3	T	.072	1	0010	1001
4	O	.0654	00	01000	1100
5	A	.063	01	0101	1101
6	N	.059	10	0110	1111
7	I	.055	000	10000	0110
8	R	.054	010	100	0111
9	S	.052	001	1010	10110
10	H	.047	0000	1100	10100
11	D	.035	100	01001	10000
12	L	.029	11	01110	11101
13	C	.023	1000	100010	101111
14	F	.0225	0010	10110	101010
15	U	.0225	110	11010	101011
16	M	.021	101	11100	100011
17	P	.0175	0100	1111	111000
18	W	.012	011	01111	1011100
19	Y	.012	0001	100011	1011101
20	G	.011	1011	10111	1000100
21	B	.0105	111	11011	1000101
22	V	.008	0110	111010	1110010
23	K	.003	1100	1110110	111001110
24	X	.002	1001	1110111100	111001100
25	J	.001	0011	1110111101	111001101
26	Q	.001	0101	1110111110	1110011110
27	Z	.001	0111	1110111111	1110011111

Table 1

---

*The probability values were taken from Brillouin.[14]

**The word space is represented by the symbol "word space".

Using the tree of Fig. 3, we start with the most common message, the word space, and assign branches with close to the proper cost. This gives code 2 of Table 1. The average cost for this code is $\bar{w}$ = 11.726 taps/letter. Therefore, e = (4.03)(1/0.347)(1/11.726) = 0.99.

If we code into a channel with two equal-cost symbols, we may use the Huffman procedure to get the optimum result. This is code 3 of Table 1 and Fig. 4. In this code the efficiency is e = 0.97. We note that the redundancy, given by 1 - e, is three times as great for code 3. This is an example which substantiates our conjecture that the cost-weighted channel is in general more efficient.

Code 2 was obtained quickly with about as much effort as for code 3. If it is not the optimum code, it is certainly very close in the value for efficiency.

13. CODING FOR CLASS IIA

13.1 GENERAL COMMENTS

Let us consider the problem of coding a message set, m_i, into a channel with symbols, d_j, such that $m < D$. In order to make use of the capacity of the channel one will want to code certain blocks of the m_i to be transmitted as a single symbol. (For the present discussion let us assume that one wishes to transmit the channel symbols singly and not coded into signal words). Let us begin by taking the case in which the channel symbols are equal-cost. This, then is problem IIA. Problem IIA, in a sense, seems to be the reverse of the problem of coding many equiprobable messages into a cost-weighted channel with fewer symbols than messages. We may, therefore, inquire whether a good code for the one problem might not also be a good code for the other. Such, indeed, turns out to be the case.

The set of blocks of messages is called the message set. It is found convenient to describe the possible message blocks by a message tree, just as code words were represented by branches on a signal tree. The tree has m branches in each fan. Each branch corresponds to one of the original messages, m_i. The length of a branch, or more precisely its horizontal component, is $(-\log p_i)$, i.e., the self-information of the message m_i.

We see that the message set must be complete. This must be true in order to be able to encode any possible infinite sequence of messages. (Consider, for example, any sequence beginning with an "uncovered sequence" and continuing with that sequence repeated over and over). If the message set is complete, there is one and only one way to form any sequence of messages into message blocks. (For a finite sequence there may be some messages left over). Since successive message blocks are independent of one another, the probability of using any given message block is then just the product of the probabilities of the messages in that block. The cost of coding will always be

$$\bar{q} = \Sigma\, p_k q_k = \Sigma\, p_k \log D = \log D \tag{112}$$

The problem then is to pick the message blocks so as to maximize

$$H = -\Sigma\, p_k \log p_k = -\Sigma_{i,k}\, p_k \alpha_{ik} \log p_i \tag{113}$$

where α_{ik} is the number of m_i in M_k.

This, of course, also maximizes $e = H/\bar{q} = H/\log D$. To do this we would like to make the p_k's as "equal as possible" for if they were all equal we would have $H = \log D$ and $e = 1$. This is similar to the reverse problem where we seek to make the signal words as equal in cost as possible. There, however, we seek to minimize $\Sigma_{k,j}\, \alpha_{kj} c_j$, where α_{kj} is the number of d_j's in W_k.

13.2 THE VERTICAL CUT SET SOLUTION

In order to approximate the ideal result we take the message set to be the vertical cut set on the message tree with D elements. This can be done as long as

$$D = b(m - 1) + 1 \qquad b \text{ is an integer} \tag{114}$$

Eq. 114 is obtained by the same reasoning that eq. 60 was. Assuming eq. 114 holds, we may bound $\log p_k$ as follows:

$$-\log p_M - (-\log p_1) \leq s \tag{115}$$

where $s = -\log p_m$, the self-information of the least probable message, and the M_k are ordered by decreasing p_k.

Hence,

$$-\log p_k \geq -\log p_1 \geq -\log p_M - s \geq \log D - s \tag{116}$$

Hence,

$$H = -\Sigma\, p_k \log p_k > \Sigma\, p_k(\log D - s) = \log D - s \tag{117}$$

We can use the inequality in (117) since not all of the equalities in (116) can hold at once. Thus we can bound the efficiency by

$$e = \frac{H}{\bar{q}} > \frac{\log D - s}{\log D} = 1 - \frac{s}{\log D} \tag{118}$$

For a given message ensemble we see that $e \to 1$ as $D \to \infty$. If $s \geq \log D$, the bound is useless. However, for p_1 approaching one and $m = D$ we have e approaching zero for the best signal set anyway. Therefore, we cannot expect to find a better universal bound with the same parameters.

Now suppose eq. 114 is not satisfied but rather

$$D = b(m - 1) + 1 + m_o, \qquad 1 < m_o < D \tag{119}$$

We cannot then get a cut set such that $M = D$. But we can find one such that $M = D - m_o$. We could then use just $D - m_o$ of the channel symbols. The efficiency is this case can be bounded as follows:

$$H > \log(D - m_o) - s \tag{120}$$

Hence,

$$e > \frac{\log(D - m_o) - s}{\log D} \tag{121}$$

Of course when eq. 119 applies we can improve the efficiency by using m_o message blocks in addition to the $D - m_o$ message blocks in the vertical cut set. However, the message set would no longer be proper. This means (as will be shown in an example in this Section) that some of the message blocks may have to be stored until later messages determine them as the correct message block to be transmitted. The complication in encoding, in effect, makes the effective number of message blocks greater than D. Thus non-proper message coding in Class II is similar in effect to using a Class III type procedure. Since the probabilities of the message blocks are no longer independent of past message blocks, we cannot hope to achieve an efficiency of one. We note that although we know that an increased efficiency is possible, it is difficult to get an improved bound over relation 121 since it is very difficult to determine the probabilities of the message blocks due to lack of independence.

We conjecture that the vertical cut set with M message blocks has the greatest entropy (H) of any proper set with M elements. This would mean that the vertical cut set method is optimum for Class IIA. The proof of this conjecture should follow along lines similar to the proof in Section 8.3 that the optimum signal set for equiprobable messages was also a vertical cut set. There we wanted to minimize $\bar{q} = \Sigma\, p_k q_k = 1/M\, \Sigma\, q_k$. The proof in that case followed simply from the fact that $\bar{q}$ was a linear function of the costs. Here we want to maximize $H = -\Sigma\, p_k \log p_k$. This is more involved since H is not a linear function of the corresponding variable $(-\log p_k)$. However, we should be able to prove the result by considering the first, second, and third derivatives of H. This proof has not been carried out in detail.

13.3 POSSIBILITY OF NON-PROPER MESSAGE SET

We may now show that sometimes an increased efficiency can be obtained by using a non-proper message set for the case where m = 2 even though we can always find a cut set with just D elements in this case. If there is to be an improvement in efficiency it will be because we can make the probabilities of using the message blocks more equal than for any proper message set. So let us start with an original message source for which any proper message set will be quite "tilted".

For example, we pick m = 2, $p_1 = 0.9$, $p_2 = 0.1$, and D = 3. We write $m_1 = a$ and $m_2 = b$. Then the optimum proper message set is (aa, ab, b). This gives

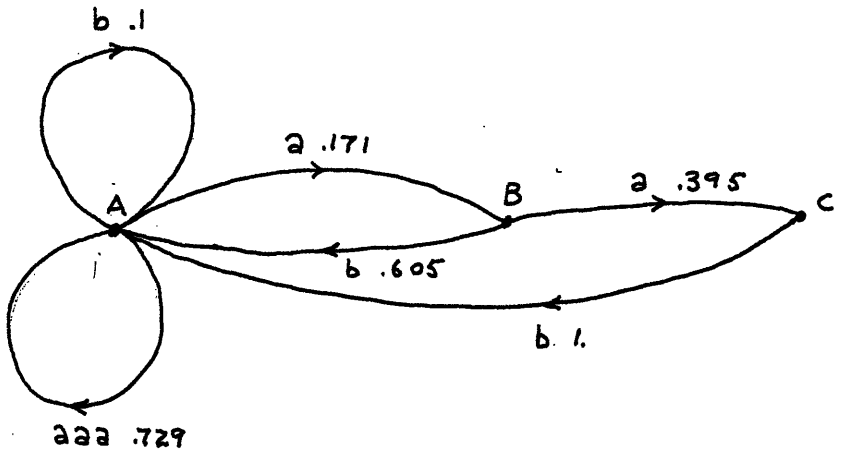
$$H = -(.81 \log .81 + .09 \log .09 + .1 \log .1) = 0.891$$

and

$$e = H/\log D = 0.891/1.58 = 0.563 \qquad (122)$$

Let us consider instead the non-proper message set ($M_1 = a$, $M_2 = b$, $M_3 = aaa$). We see immediately that the encoding operation is not yet determined. We must say when to use M_1 and M_3. Let us say use M_3 as often as possible. It may be necessary, then, to store up some of the m_i and delay coding the message blocks until sufficient information is available. For example, for the sequence aab, one would have to wait for the b to decide that the first a should be encoded as M_1 rather than as part of M_3. But now the message blocks are no longer independent.

The probabilities, p_k, are no longer merely products of the p_i and as a matter of fact they may change depending on the previous history of the source. For example, after message M_1 is sent the probability of M_3 becomes zero. To determine the information rate for the channel we must determine all the probabilities for the various states the channel may be in and average the rate for all of these states according to the probabilities of the states. A state diagram of the coding operation follows;



The three states with possible transitions from each are shown.. Some simple combinatorial analysis gives the probabilities associated with each transition. Then the state probabilities, P_A, P_B, and P_C, are determined from the following set of equations:

$$\begin{aligned} P_A + P_B + P_C &= 1 \\ P_B &= 0.171\, P_A \\ P_C &= 0.395\, P_B = 0.0675\, P_A \end{aligned} \tag{123}$$

The solutions are $P_A = 0.808$, $P_B = 0.138$, $P_C = 0.054$. The information rate for each state is just the entropy of the probability distribution of the transitions from that state. Thus we have

$$H_A = -(.729 \log .729 + .171 \log .171 + .1 \log .1) = 1.101 \text{ bits}$$

$$H_B = -(.605 \log .605 + .395 \log .395) = 0.947 \text{ bits}$$

$$H_C = -1 \log 1 = 0$$

The average entropy is then

$$H = P_A H_A + P_B H_B + P_C H_C = .890 + .130 = 1.02 \text{ bits} \tag{124}$$

This gives a higher efficiency than for the optimum proper code.

$$e = \frac{1.02}{1.585} = 0.644 > 0.563 \tag{125}$$

We see that if we considered message blocks of length three we would get the following proper code of type Class IIIA; (We write the channel symbols in ternary numbers).

MESSAGE BLOCK	CODE WORD
M_1 = aaa	0
M_2 = aab	112
M_3 = aba	121
M_4 = abb	122
M_5 = baa	211
M_6 = bab	212
M_7 = bba	221
M_8 = bbb	222

14. CODING FOR CLASS IIB

The problem for the cost-weighted channel (Class IIB) is more difficult than for the equal-cost channel (Class IIA). Here the cost is no longer just log D but is given by

$$\bar{q} = \Sigma\, p_k q_k \tag{126}$$

where $q_k = -\log P_{mj}$ for some j.

For minimum cost we want to obey the order rule so we set k = j, where the M_k's are arranged according to decreasing probability and the d_j's according to increasing cost. So now we want to maximize

$$e = \frac{H}{\bar{q}} = \frac{-\Sigma\, p_k \log p_k}{\Sigma\, p_k q_k} \tag{127}$$

by proper choice of the M_k. We obtain an efficiency of one, which we know is an absolute maximum by the Fundamental Theorem, for q_k given by

$$q_k = -\log p_k \tag{128}$$

Let us restrict ourselves to proper codes for the present. Therefore, we have to pick a cut set of D branches in the message tree which most closely satisfies eq. 128. We note that the extended Shannon code in reverse (i.e., pick an M_k such that $q_k + s \geq -\log p_k \geq q_k$)will not work since this does not, in general, give a cut set. We could consider picking the message blocks corresponding to the node to the left of the first branch from the top which is cut by the vertical line corresponding to q_k, and which had not previously been prefixed by a used message block. This would give the bound $q_k \geq -\log p_k \geq q_k - s$. We cannot do this for all q_k, however, since we would use up the tree,ingeneral, before we got to q_D. However, we could follow this procedure at first and then follow the extended Shannon procedure, changing over from one to the other at a point so as to obtain a cut set. We are assured that this is possible as long as eq. 114 is satisfied. The combined bound on a single message block is then

$$\begin{aligned} &q_k + s > -\log p_k > q_k - s \\ &p_m 2^{-q_k} < p_k < (2^{-q_k})/p_m \end{aligned} \tag{129}$$

Thus

$$e = \frac{-\Sigma\, p_k \log p_k}{\Sigma\, p_k q_k} > 1 - \frac{s}{\sum_{k=1}^{M} p_k q_k} > 1 - \frac{s}{H - s}$$

or

$$e > 1 - \frac{s/p_m}{\sum_{k=1}^{M} q_k 2^{-q_k}} = 1 - \frac{(-\log p_m)/p_m}{-\sum_{j=1}^{M} P_{mj} \log P_{mj}} \tag{130}$$

where H is the entropy of the message set.

If eq. 119 rather than eq. 114 is satisfied, we cannot obtain a cut set with D message blocks. Here, as in Class IIA, we may use just the first $D - m_o$ channel symbols. The bound 130 still applies where M now equals $D - m_o$. Again, as in Class IIA, we may increase the efficiency at the cost of added complexity by adding m_o message blocks and making the message set non-proper.

We note that no optimum solution has been obtained for Class IIB and the problem of the optimum solution appears about as difficult as for ClassIIB. In practice, to achieve an efficient solution one could set up the message tree and make good approximations to the ideal. One could also apply the Blachman procedure in which we treat the channel symbols as messages with probabilities q_k to be encoded into channel symbols equal in number to m which have costs such that the probabilities of their optimum use are p_i, the probabilities of the messages. Of course the message set obtained must be reduced to a cut set.

15. CODING FOR CLASS III

We now consider the most general coding problem (within the scope of this paper). This is problem III in which the message source is first coded into message blocks and these blocks are then coded into sequences of channel symbols called code words. We may abbreviate this description by referring to it as "block coding". For Class III relationship 22c applies: $m < M > D$. For problem III it is very important to consider complexity when comparing two coding procedures. This is true because we can make the efficiency arbitrarily close to one by a sufficiently complex code as will be shown (and, indeed, as follows from the Fundamental Theorem).

15.1 EQUAL LENGTH BLOCKS

The simplest kind of block coding occurs when the message set consists of all possible permutations of length L of the messages. This is a convenient arrangement as far as the message set is concerned. This message set may then be coded into signal words by any procedure. (The Huffman procedure will give maximum efficiency in the equal-cost case). The message set has $M = m^L$ members. The costliest word is bounded by

$$Ls \leq q_M < Ls + L_D \tag{131}$$

The entropy of the message set, which we write H, is easily shown to be:

$H = LH_o$, where H_o is the entropy of the original message source. (See proof in footnote*). The extened Shannon procedure gives us the bound

$$\bar{q} < H + L_D = LH_o + L_D \tag{132}$$

Hence,

$$e > \frac{H}{H + L_D} = \frac{LH_o}{LH_o + L_D} = \frac{1}{1 + \frac{L_D}{LH_o}} \tag{133}$$

Thus the efficiency approaches one as L approaches infinity. This is a direct constructive proof of the Fundamental Theorem. It was given by Shannon[1] for the binary equal-cost channel.

It may be shown[8,9] that for the Fano and Blachman procedures the efficiency also approaches one as the block length approaches infinity. for the equal length block coded message set. (One difficulty in the proof for the Blachman procedure has already been noted in Section 5. The conclusion should still be valid, however).

15.2 BALANCED BLOCKS

Another block coding scheme may be called balanced coding. In this procedure one attempts to code a message set of equal or nearly equal probabilities for the message blocks into a signal set of equi-

---

* $$H_o = -\Sigma\, p_i \log p_i \tag{134}$$

$$H = -\Sigma_{j_1, j_2, \ldots, j_L}\; p_{j_1} p_{j_2} \cdots p_{j_L} \log p_{j_1} p_{j_2} \cdots p_{j_L} \tag{135}$$

where p_{j_1} is the probability of the symbol used in the first position, etc.

$$H = -\Sigma_{j_1, j_2, \ldots, j_L}\; p_{j_1} p_{j_2} \cdots p_{j_L} (\log p_{j_1} + \ldots + \log p_{j_L}) =$$

$$-\Sigma_{j_1 \ldots j_L}\; p_{j_1} \cdots p_{j_L} \log p_{j_1} - \ldots - \Sigma_{j_1 \ldots j_L}\; p_{j_1} \cdots p_{j_L} \log p_{j_L} \tag{136}$$

But if we take the first sum over all $j_2, j_3, \ldots, j_L$ we get

$$-\Sigma_{j_1}\; p_{j_1} \log p_{j_1} = H_o \qquad \text{since} \quad \Sigma_{j_2 \ldots j_L}\; p_{j_2} \cdots p_{j_L} = 1 \tag{137}$$

Hence,

$$H = H_o + H_o + \ldots + H_o = LH_o \tag{138}$$

costly (or nearly so) signal words, i.e., the coding groups are to be balanced in cost, q_k, and probability, p_k. Specifically, pick a given M. Then take the first vertical cut set in both the message and signal trees to have M members. The message blocks in order of decreasing probability are then matched to the signal words in order of increasing cost. The order rule says that this is the best match for the two given sets.

Of course it is not possible to find just M members in each cut set unless we satisfy eqs. 114 and 60. However, if m = D we may always find a match for any M = b(m - 1) + 1 or if m ≠ D we may find a match for some M (namely those such that M = a(m - 1) + 1 = b(D - 1) + 1 for some integral values of a and b). In general we may always pick M = a(m - 1) + 1 and then pick the first signal cut set with M or more branches and discard the most costly signal words until just M are left. If this is done we have

$$q_M - q_1 \leq L_D \tag{139}$$

and

$$\log p_1 - \log p_M \leq s \tag{140}$$

where p_1 and p_M are the probabilities of the least and most probable message blocks, respectively.

A crude* bound may be obtained on the efficiency as follows

$$p_k \leq 1/M, \quad \text{hence,} \quad -\log p_M \geq \log M \tag{141}$$

Hence,

$$-\log p_k \geq -\log p_1 \geq -\log p_M - s \geq \log M - s \tag{142}$$

Hence,

$$H = -\Sigma\, p_k \log p_k > \Sigma\, p_k(\log M - s) = \log M - s \tag{143}$$

We have a strict inequality in (143) since all the equalities cannot hold at once in(142) for all k.

Similarly, we may bound $\bar{q}$

$$\bar{q} = \Sigma\, p_k q_k \leq \Sigma\, p_k(\log M + L_D) = \log M + L_D \tag{144}$$

Hence,

$$e > \frac{\log M - s}{\log M + L_D} = \frac{1 - s/\log M}{1 + L_D/\log M} \tag{145}$$

We note that the efficiency approaches one as M approaches infinity. This is another constructive proof of the Fundamental Theorem. We may

---

*Bound 147 is better.

also write

$$e > \frac{H}{\log M + L_D} \tag{146}$$

Assuming that the balanced method gives at least as good a result as does the extended Shannon procedure with the same message set (a statement which seems clearly true but has not been proven), we may use the bound given in relation 48. Thus

$$e > \frac{1}{1 + \frac{L_D}{H}} > \frac{1}{1 + \frac{L_D}{\log M - s}} \tag{147}$$

15.3 COMPARISON OF 15.2 AND 15.1

We may compare balanced coding with equal block length coding in three areas. In the first place we note that balanced coding is advantageous where it is desired to keep the costliest signal word from being too expensive. We see from eq. 46 that

$$q_k < \log M + L_D = L \log m + L_D \tag{148}$$

For equal length block coding, on the other hand, the least probable message gets far from the mean and we may approach the bound 131

$$Ls < q_M < Ls + L_D \tag{149}$$

We know that $s \geq \log m$, with the possibility of the inequality being quite pronounced if the original message distribution is strongly tilted. Therefore, for large L we may have $q_M >> q'_M$, where the primed and unprimed q's refer to balanced and equal length block coding, respectively.

Secondly, the bound for 15.2 is better in some sense than that for 15.1. That is, if M is big enough, for the same M, $e' > e$ or $e' = e$ for $M' < M$.* We see that this is so as follows:

$$e = \frac{1}{1 + \frac{L_D}{LH_o}} \tag{150}$$

where H_o is the entropy of the original m messages.

---

*Note that e and e' are here bounds on the efficiency in the two cases and not the actual efficiencies.

$$e' \doteq \frac{1}{1 + \frac{L_D}{H'}} > \frac{1}{1 + \frac{L_D}{\log M - s}} \tag{151}$$

where H' is the entropy of the M messages in the balanced message set. But

$$M = m^L \tag{152}$$

Hence,

$$e' > \frac{1}{1 + \frac{L_D}{L(\log m - s/L)}} \tag{153}$$

For large L, s/L becomes insignificant and the expression in parenthesis becomes essentially log m, as compared to H_o in the expression for e. But $\log m \geq H_o$ with the inequality again more pronounced for more tilted original message distributions. Thus our hypothesis is proved for large L which means large M. This does not say which message set would actually be better as far as efficiency is concerned and, in general, it could be either.

As a third consideration, however, a particular class of distributions exists for which it is definitely better to use the balanced message set to get the best efficiency for a given M when M is small. This occurs when p_1 is close to one and $L_1 >> -\log p_1$. Here we cannot expect to get a good match until $-\log p_{k1} \doteq L_1$ (where p_{k1} is the probability of the first message block, with the subscript k added here to distinguish it from the probability of the most probable original message, p_1.) The balanced message set achieves this result for the smallest value of M. (See Bennett,[16] Shannon[1] (p. 33), and Elias[17]).

15.4 THE TREE PICTURE AGAIN

We note that eq. 127 applies to Class III as well as Class II. Thus the efficiency has a perfect value of one if, and only if, $q_k = -\log p_k$ for all k. Here q_k may be the cost of any code word. We may then think of the problem as one of matching the message tree and the signal tree such that a message set is matched to some signal set. This is a useful interpretation when one is actually trying to match a specific message ensemble to a given channel.

15.5 POSSIBILITY OF A NON-PROPER MESSAGE SET

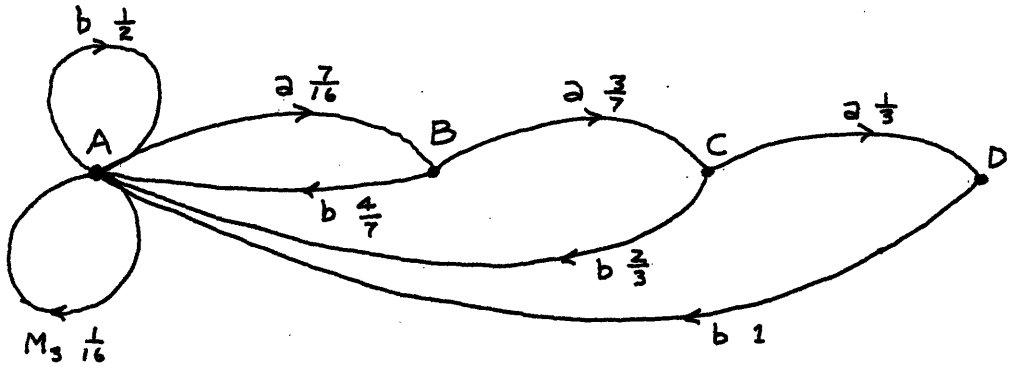
We may show that for a given M we may achieve a greater efficiency with a non-proper message set than with any proper message set. Let $m = 2$, $p_1 = p_2 = 1/2$, $D = 3$, $c_1 = 1$, $c_2 = 1$, $c_3 = B$, $M = 3$. The proper message set is either (a, b) or (ab, aa, b).

$$e = H/\bar{w}C = R/C \tag{154}$$

Since C is fixed we may maximize the efficiency by maximizing the rate, R. For set one, $R = 1/1 = 1$ bit/unc. For set two R is given by

$$R = \frac{.5 + .5 + .5}{.5 + .25 + .25B} = \frac{1.5}{.75 + .25B} \quad \text{bits/unc.}$$

Let us now consider the non-proper message set: $M_1 = a$, $M_2 = b$, $M_3 = aaaa$. We further stipulate that M_3 should be used as often as possible. We then have the following state diagram describing the process. The probabilities of transition are given.



The probabilities of the various states are determined by the following set of equations:

$$\begin{aligned} P_A + P_B + P_C + P_D &= 1 \\ P_B &= 7/16\ P_A \\ P_C &= 2/7\ P_B = 3/16\ P_A \\ P_D &= 1/3\ P_C = 1/16\ P_A \end{aligned} \tag{155}$$

These give

$$P_A = 16/27,\ P_B = 7/27,\ P_C = 3/27,\ P_D = 1/27 \tag{156}$$

For B = 3 we obtain the following information rates for the probability distributions of the states:

$$\begin{aligned} R_A &= H_A/\bar{w}_A = 1.27/1.125 = 1.128 \\ R_B &= H_B/\bar{w}_B = .97/1 = 0.97 \\ R_C &= H_C/\bar{w}_C = .91/1 = 0.91 \\ R_D &= H_D/\bar{w}_D = 0/1 = 0. \end{aligned} \tag{157}$$

This gives a total average information rate R

$$R = P_A R_A + P_B R_B + P_C R_C + P_D R_D = 1.02 \tag{158}$$

This is greater than the rate for either proper message set (which is one for B = 3). Indeed, for a range of B values up to b = 3.6 this non-proper message set is more efficient than any proper one for M = 3. However, as we noted for the example of an efficient non-proper message set for Class IIA in Section 13, this code is more complicated and should perhaps be compared with a message set with $2^4 = 16$ blocks.

16. MULTIPLE-COST CODING

Suppose each symbol has two costs associated with it. For example, one cost in time and one in power. What, then, can we mean by maximizing the information rate? There are two information rates now (one for each cost). One solution would be to decide the relative importance of the two cost factors and combine them to make a single cost factor. For example, if c_j' represents one set of costs and c_j'' another, we may decide that the first set is r times more important than the second one and get a combined cost factor set, c_j, where

$$c_j = rc_j' + c_j'' \tag{159}$$

Of course this is directly extended to many sets of cost factors.

Another way to approach the problem is to consider the information rates separately. Let us call them R_1 and R_2.

$$R_1 = \frac{-\Sigma\, p_j \log p_j}{\Sigma\, p_j c_j'} = \frac{H}{\bar{c}'}, \quad R_2 = \frac{H}{\bar{c}''} \tag{160}$$

We may then ask for the p_j set which will maximize R_1 for a given R_2, or vice versa. Blachman[9] shows that the solution should be of the form

$$p_j = 2^{-\mu c_j' - \nu c_j''} = P_{mj} \tag{161}$$

μ and ν are positive constants such that R_2 equals the given value and $\Sigma p_j = 1$. If there is more than one solution meeting these conditions, the solution for which R_1 is the maximum is the one to choose. Several other ways of maximizing the information rate have solutions of the form of (161). For example, maximize R_1 for a given $\bar{c}''$, etc. After the P_{mj} are determined the coding may be done as before.

17. OTHER FORMULATIONS OF THE CODING PROBLEM

One may describe the coding operation in terms other than the message block and code word picture given above. Shannon, Laemmel, and Schutzenberger all give such descriptions. For example, Shannon[1] describes the coder as a transducer which accepts input symbols (what we call messages) and transmits output symbols (channel symbols). The transducer is described by two functions:

$$\begin{aligned} d_n &= f(m_n, \alpha_n) \\ \alpha_{n+1} &= g(m_n, \alpha_n) \end{aligned} \tag{162}$$

where m_n is the n^{th} input message

α_n is the state of the transducer at the n^{th} message

d_n is the channel symbol (or sequence of channel symbols) produced when m_n is introduced and if the state is α_n

α_{n+1} is the state after d_n is transmitted.

To enable this description to include all codes previously considered we must allow one of the symbols to be a blank, i.e., for some inputs we do not immediately prescribe any channel symbol. Note that in general we want to use storage devices in the coding operation to allow a steady continuous flow of channel symbols.

Laemmel[4,19] uses the Shannon formulation and classifies extensively various types of codes with the aid of state diagrams of the coding operation. He gives much discussion to the complexity of the coding apparatus and other practical problems which are only briefly mentioned in this paper.

Schutzenberger[12] uses the notions and properties of semi-groups to develop a mathematical formulation of the coding process and to investigate some properties of this process including the decodability criterion of Sardinas and Patterson and the effect of an original error on the decoding.

18. SOME MORE GENERAL CONSIDERATIONS

Certain assumptions have been made or implied which restrict the generality of this discussion. In the first place we considered only memoryless message sources. If a source has correlation among successive messages, as is true for instance for the letters of the alphabet when considered in written text, the rate of information generated by the source is lower than the rate indicated by the average probabilities of the letters. For one kind of source the probabilities depend only on a finite number, say p, of past messages, We could code efficiently for this kind of source if we made a separate code for each of the m^p possible probability distributions. However, if the dependence is chiefly on just a few past messages, we could achieve greater efficiency for the same complexity (in terms of entries in a code book) by coding for blocks of messages of length p_1 depending on the p_2 previous messages, where $p_1 + p_2 = p + 1$. This would give m^{p_1} entries in m^{p_2} codes for a total of $m^{p_1 + p_2} = m^{p+1}$ entries, the same as before. Of course we can achieve greater efficiency by increasing p_1 while holding p_2 constant. Efficiency approaching unity is obtained by making $p_2 = p$ and letting p_1 approach infinity.

A somewhat more general description is a finite state Markov process. Here the probabilities depend on which of a finite number of states the source is in. A state is determined by the previous state and message. We may code for each state either for single messages or blocks of them.

Even more general sources could be described by an infinite state Markov process or by a generalized ergodic process which could depend on time as well as past history. This paper considered, specifically, only finite numbers of messages. We could consider an infinite number or even a continuum.

The channels considered were finite, discrete, noiseless, and, for the most part, time invariant. We did consider to some extent the possibility of a channel with certain kinds of constraints. There remains the possibility of more general channels whose constraints and costs are variable in a fashion similar to variations of probabilities for sources.

It should also be pointed out that much consideration should be given to the difficulty of applying the information theory model of the communication problem to real situations. It is possible that weights other than or in addition to (-log p) might be applicable for the information content of messages in certain cases. Practical problems of complexity of operation and susceptibility to error may outweigh factors like efficiency in the information theory sense. Still, it is important to have the information theory model as a guide to how well the coding can be done. Finally, we note that only error-free coding was considered. For $R > C$ we must have errors.

19. SUGGESTIONS FOR RESEARCH TOPICS

For those interested in a thesis topic (about the Master's level) the author offers the following observations. There are several unanswered problems of varying importance and interest stated in this paper. An evaluation of recent work in the field which is either discussed in this paper or in the references is possible. A discussion of coding for more complicated sources and channels, as mentioned in Section 18, is indicated.

There are two general regions of the coding problem. One is the mathematical formulation. The Schutzenberger work is largely in this region. The second is application to real and practical problems. Much of Laemmel's work is pointed in this direction. A prospective researcher who is not equipped for work in the first region could find many problems in the second. The application of the theory to actual situations is what determines a part of the usefulness of the mathematical formulation. Of course part of the usefulness of the mathematical formulation is also in leading to other formulations and to a better understanding of the nature of the problem.

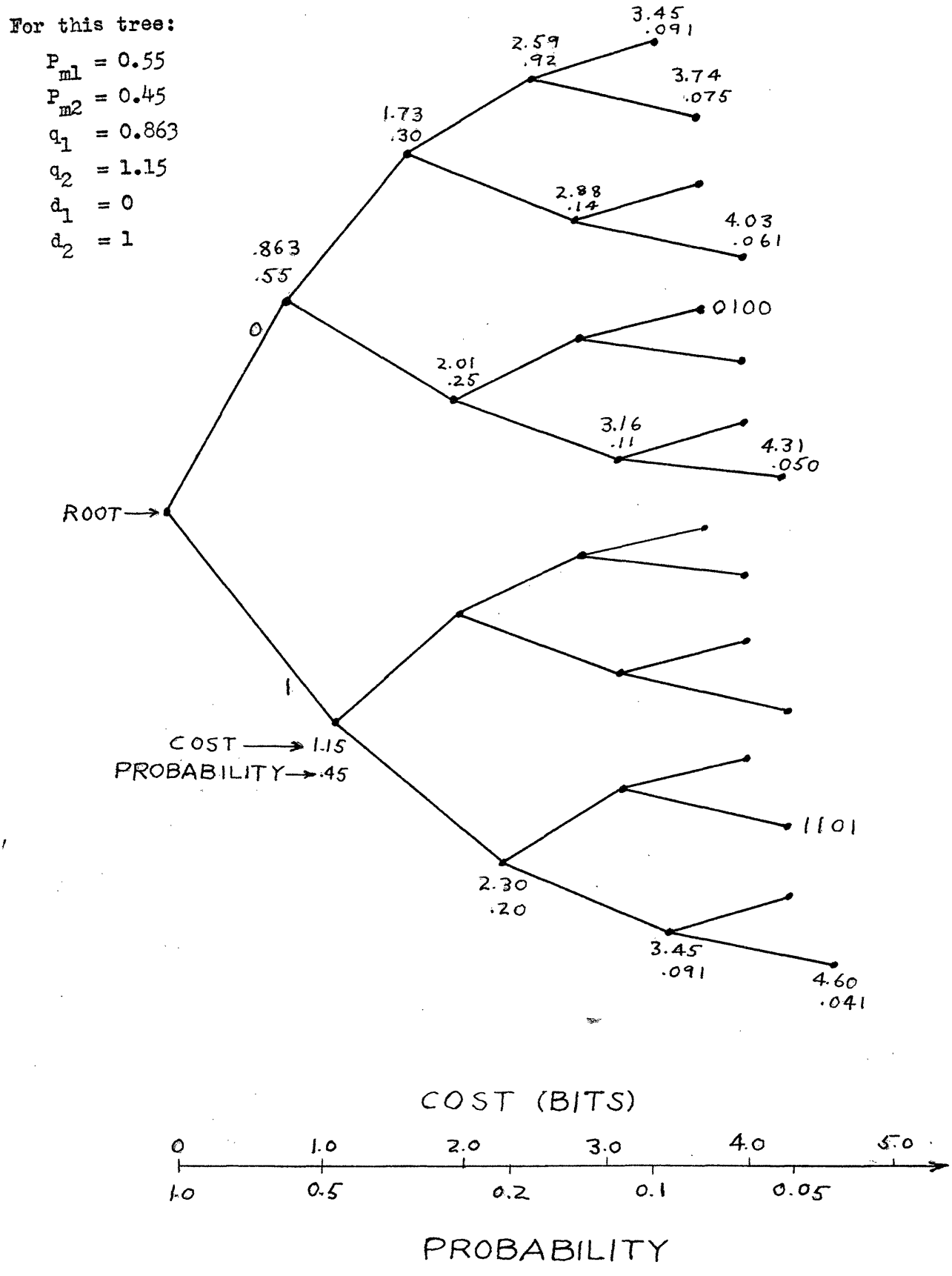


Fig. 1. A tree graph for coding

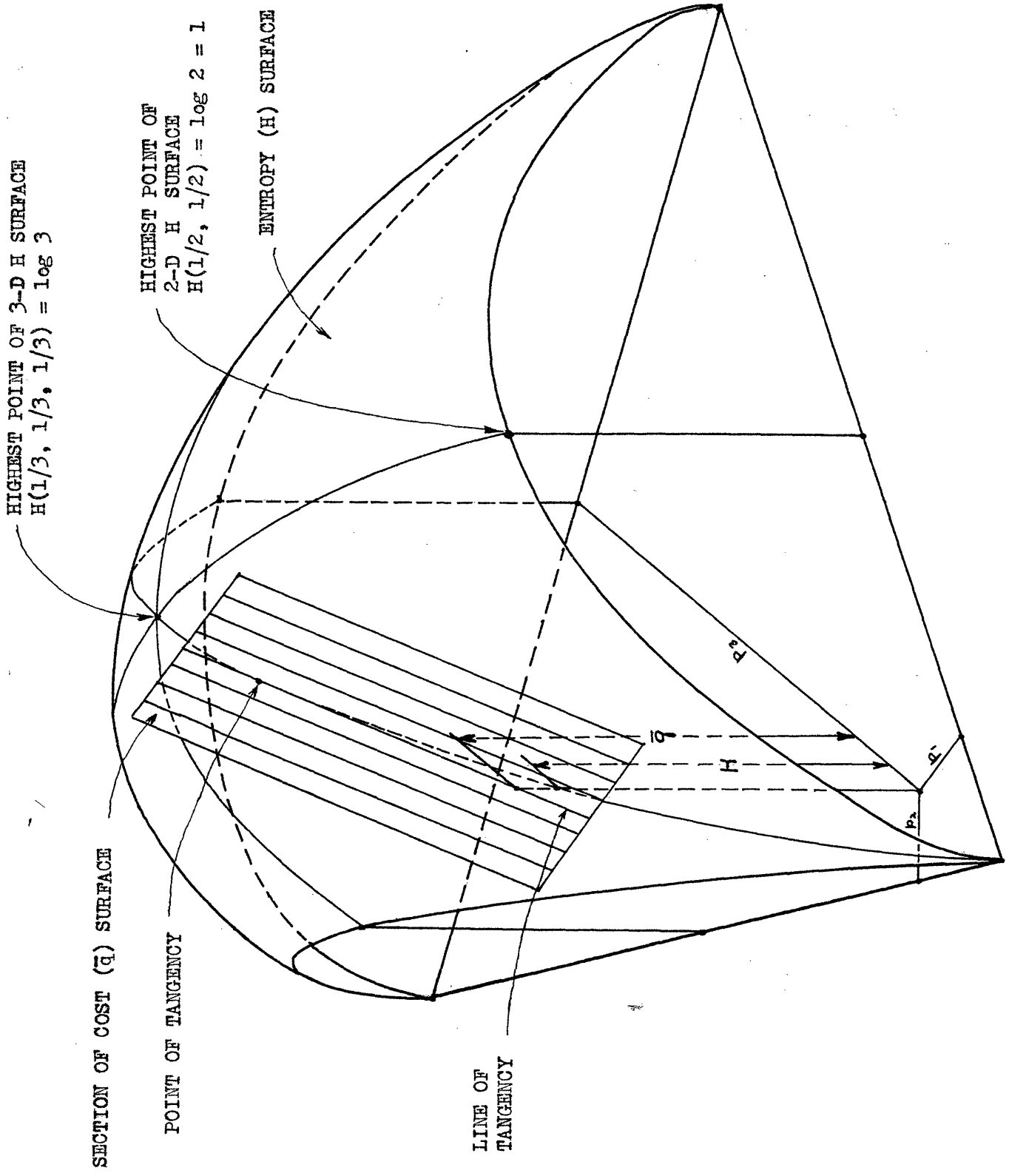


Fig. 2. Entropy (H) and cost ($\bar{q}$) surfaces in three dimensions

For this tree:

$c_2 = 2c_1$

$P_{m1} = 0.618$

$q_1 = 0.695$ bits

$P_{m2} = 0.382$

$q_2 = 1.39$ bits

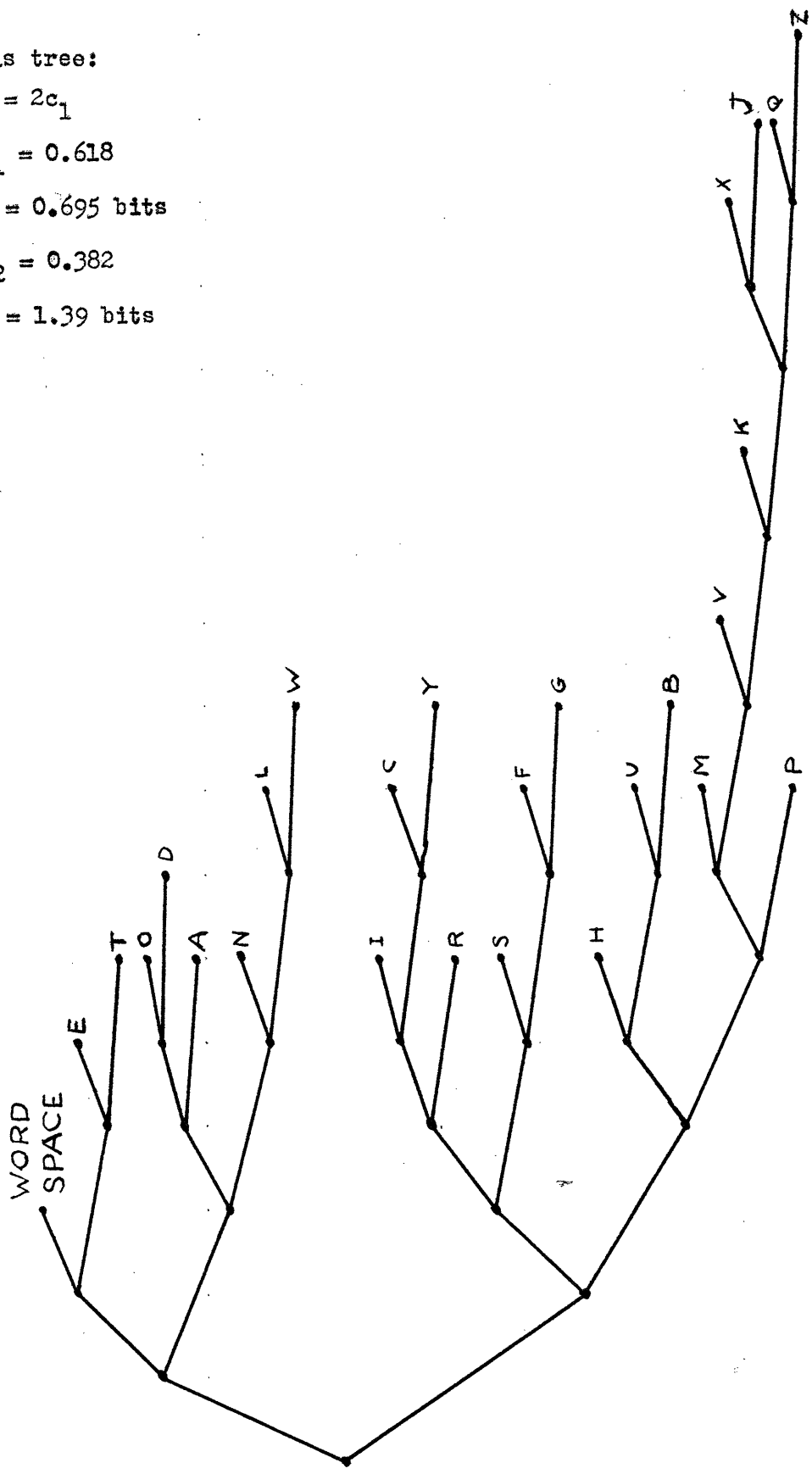


Fig. 3 - Signal tree showing code 2

For this tree:

$$c_1 = c_2$$

Hence,

$$P_{m1} = P_{m2} = 1/2$$

$$q_1 = q_2 = 1.0$$

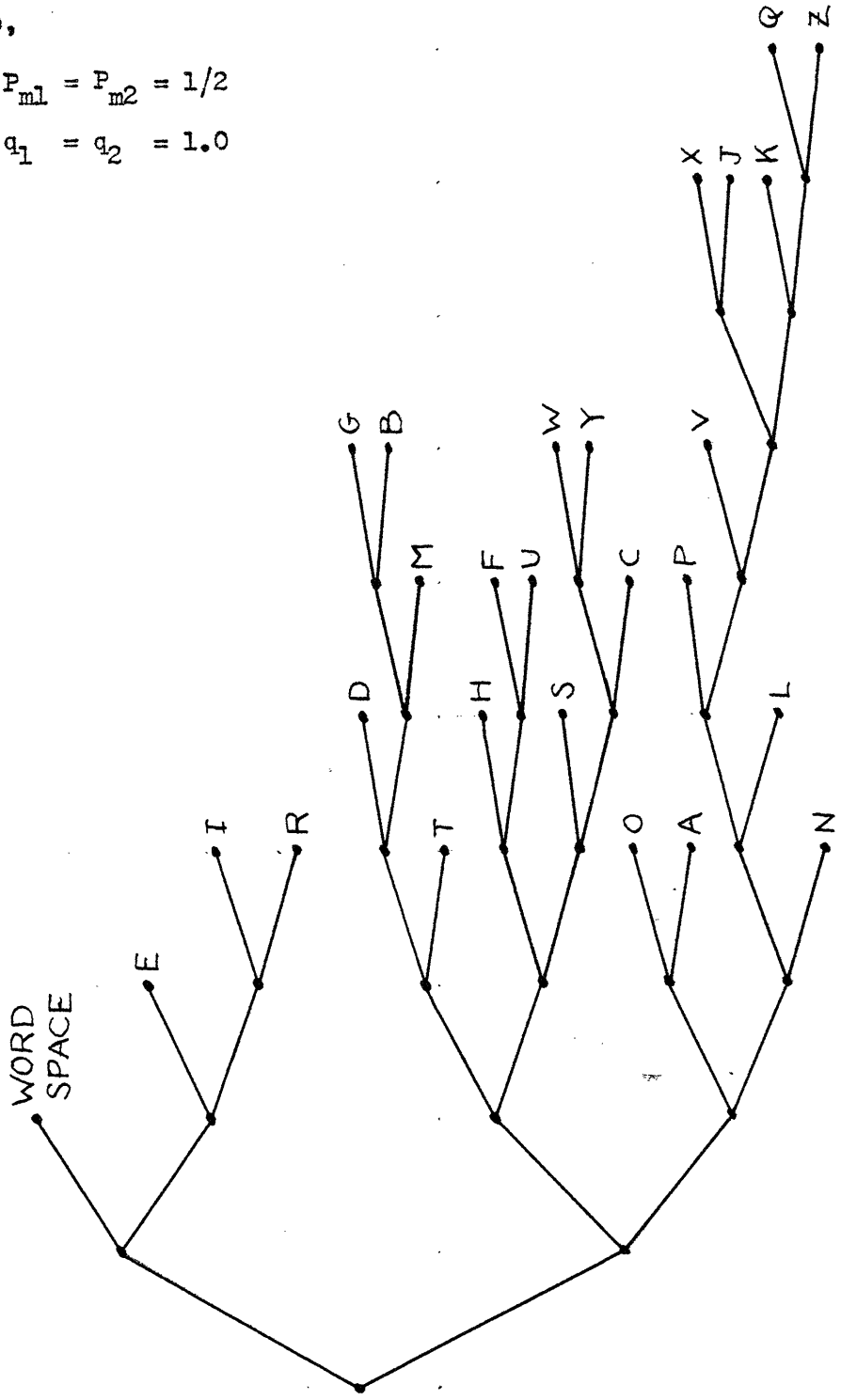


Fig. 4. Signal tree showing code 3

BIBLIOGRAPHY

1. Shannon, C.E.; "The Mathematical Theory of Communication"; Bell System Technical Journal; July, 1948.
also: Shannon, C.E. and Weaver W.; The Mathematical Theory of Communication; Univ. of Illinois Press; Urbana, Ill.; 1949.

2. Fano, R.M.; Statistical Theory of Information; Notes for subject 6.574 at the Massachusetts Institute of Technology; 1953.

3. Casper, L; Telegraph Operating; International Textbook Company; Scranton, Pa.; 1928.

4. Laemmel, A.E.;and Brogan, J.M.; "Coded Transmission of Information", Research Report R-325-53, PIB-261; Microwave Research Institute, Polytechnic Institute of Brooklyn; Jan., 1954.

5. Kraft, L.G.; "A Device for Quantizing, Grouping, and Coding Amplitude Modulated Pulses"; M.S. Thesis, Electrical Engineering Department, Massachusetts Institute of Technology; 1949.

6. Mandelbrot, B.; "On Recurrent Noise Limiting Coding"; Proceedings of the Symposium on Information Networks; New York City; 1954.

7. Szilard, L.; Zeitschrift fur Physic; 53; p. 840; 1929.

8. Fano, R.M.; Technical Report No. 65; Research Laboratory for Electronics, Massachusetts Institute of Technology; March, 1949.

9. Blachman, N.M.; "Minimum-Cost Encoding of Information"; Transactions of the Institute of Radio Engineers, Professional Group on Information Theory, PGIT-3; Symposium on Statistical Methods in Communication Engineering; p. 139; March, 1954.

10. Huffman, D.A.; "A Method for the Construction of Minimum Redundancy Codes"; Communication Networks; edited by Willis Jackson; London; 1953. Also in Proceedings of the Institute of Radio Engineers, 40; p. 1095; Sept, 1952.

11. Sardinas, A.A. and Patterson, G.W.; "A Necessary and Sufficient Condition for Unique Decomposition of Coded Messages"; Convention Record of the I.R.E.; Part 8, 1953.

12. Schutzenberger, M.P.; "On an Application of Semi-Group Methods to Some Problems in Coding"; Transactions of the I.R.E. on Information Theory, IT-2 NO. 3; p. 47; Sept., 1956.

13. Mandelbrot, B.; "Theorie des informations en l'absence de bruit"; Institut de Statistique, Univ. de Paris; 1955.

BIBLIOGRAPHY (2)

14. Brillouin, L.; Science and Information Theory; N.Y.; 1956.

15. Kunisawa, K., Honda, N. and Ikeno, N.; "Equal Length Coding in Discrete Channels"; Paper given at XIth URSI General Assembly; The Hague; 1954.

16. Bennett, F.K.; "Some Aspects of Digital Coding"; M.S. Thesis, Massachusetts Institute of Technology, Electrical Engineering Department; 1951.

17. Elias, P.E.; "Predictive Coding"; I.R.E. Transactions on Information Theory, PGIT IT-1; p. 30; March, 1955.

18. Feller, E.; An Introduction to Probability Theory and Its Applications; John Wiley and Sons; N.Y.; 1950

19. Laemmel, A.E.; "A General Class of Discrete Codes and Certain of their Properties"; Research Report R-459-55, PIB-389; Microwave Research Institute, Polytechnic Institute of Brooklyn; Jan., 1956.

20. Otter, R.; "The Multiplicative Process"; Annals of Mathematical Statistics; Vol XX; 1949.

21. Gnedenko, B.E. and Kolmogorov, A.N.; Limit Distributions for Sums of Independent Random Variables; Translated from the Russian; Addison-Wesley Pub. Co.; Cambridge, Mass.; 1956.