Revised March 1980 ESL-P-800

GENERALIZATION OF HUFFMAN CODING TO MINIMIZE THE PROBABILITY OF BUFFER OVERFLOW

BY

Pierre A. Humblet*

Abstract

An algorithm is given to find a prefix condition code that minimizes the value of the moment generating function of its codeword length distribution for a given positive argument. This algorithm is used in an iterative way to yield a code that maximizes the rate of decay of the probability of buffer overflow as the buffer length increases.

*This paper was presented at the 1977 International Symposium on Information Theory, Ithaca, N.Y.
This work was supported in part by a Frank A. Cowan Scholarship and in part by the Advanced Research Projects Agency of the Department of Defense under Grant N00014-75-C-1183. It is part of a dissertation submitted to the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, in partial fulfillment of the requirements for the Ph.D. degree.
The author is with the Department of Electrical Engineering and Computer Science and the Laboratory for Information and Decisions Systems, Massachusetts Institute of Technology, Cambridge, MA 02139

I. Introduction

A source emits symbols drawn from the alphabet $\{1,2,\dots c\}$; symbol i has probability p_i. The source symbols are encoded into d-ary codewords. The codeword corresponding to symbol i has length m_i.

It is well known that Huffman's procedure [2] yields a prefix condition code minimizing $\sum_{i=1}^{c} p_i m_i$. We show in Section II that it can be generalized to yield a prefix condition code minimizing the moment generating function $\sum_{i=1}^{c} p_i e^{sm_i}$ for a given $s > 0$.

A drawback of transmitting variable length codewords over a synchronous line is the need for a buffer at the output of the encoder. This buffer may overflow. The probability that a source symbol causes an overflow is of the order of e^{-s^*B}, as can be shown by Wyner's results [6]. In this formula B denotes the buffer size and s* is the largest s (possibly ∞) such that

$$A(s) \sum_{i=1}^{c} p_i e^{sm_i} \leq 1 \tag{1}$$

where $A(s) \overset{\Delta}{=} E[\exp(-st)]$ denotes the Laplace-Stieltjes transform of the probability distribution of the source intermission times measured in units of encoded digit transmission time. This result requires the mutual independence of all source symbols and interemission times, and holds only if the mean interemission time is greater than the average codeword length.

It is thus desirable to use a code with s* as large as possible when the probability of buffer overflow is required to be small, so that the asymptotic approximation is good. The search for such a code is the subject of Section III. We consider only the problem of symbol by symbol encoding, asymptotic properties have been considered in [4], and variable length to block coding in [5]. The use of s* as a criterion was first suggested by Jelinek [4].

II. Minimization of $\sum_{i=1}^{c} p_i e^{sm_i}$, $s > 0$.

Without loss of essential generality we can assume $c = d + k(d-1)$ for some integer k (so that c is the number of terminal nodes in some d-ary trees) and $p_i \geq p_{i+1}$, $i = 1,2,\dots c-1$. As for every uniquely decodable code there is a prefix condition code with the same set of codeword lengths [1,p.49], no gain can be achieved by minimizing $\sum_{i=1}^{c} p_i e^{sm_i}$ over all uniquely decodable codes, rather than only over the prefix condition codes.

Because $s > 0$, e^{sm_i} increases with m_i, thus, analogously to the argument given by Huffman [2], there is an optimal prefix condition code where the codewords corresponding to symbols c-d+1 are the longest and differ only in the last character. If c=d this observation specifies an optimal code. If $c > d$ this reduces the problem to finding a prefix condition code of size $c-d+1 = d + (k-1)(d-1)$ minimizing

$$\sum_{i=1}^{c-d} p_i e^{sm_i} + (e^s \sum_{i=c-d+1}^{c} p_i) e^{sm_{c-d+1}} .$$

Thus the "merge" step in the Huffman construction is replaced here by "merge and scale by e^s".

Again we can make the same observation and continuing we will eventually reach the point where the code is completely specified.

While this paper was being revised, the preceeding algorithm has been obtained independently by [7].

For s close enough to 0, this algorithm yields a Huffman code, since $e^{sm_i} \simeq 1+sm_i$, so that minimizing $\sum_{i=1}^{c} p_i e^{sm_i}$ is equivalent to minimizing $\sum_{i=1}^{c} p_i m_i$. For s large enough, this algorithm assigns codewords of length

$\lceil \log_d c \rceil -1$ to symbols 1 to $\dfrac{d^{\lceil \log_d c \rceil} - c}{d-1}$, and codewords of length $\lceil \log_d c \rceil$ to the others; by definition we say that such a code is generated by this algorithm for $s=\infty$.

One might wonder if "merge and scale" algorithms can be used to minimize more general cost functions of the form $\sum_{i=1}^{c} p_i \, g(m_i)$. Unfortunately, the answer is no. For the algorithm to work, g must be such that $g(m+1) = a\, g(m)+b$. This limits g to being linear or exponential. Note however that the algorithm given here can also be used to find a code maximizing $\sum_{i=1}^{c} p_i \, e^{sm_i}$, $s < 0$.

III. Finding A Prefix Condition Code With Largest s*.

Following [4] we first note that it is possible to upperbound s* over all uniquely decodable codes. By Holder's inequality, for s > 0,

$$\left(\sum_{i=1}^{c} p_i e^{sm_i}\right)^{\frac{\ln d}{\ln d+s}} \left(\sum_{i=1}^{c} d^{-m_i}\right)^{\frac{s}{\ln d+s}} \geq \sum_{i=1}^{c} p_i^{\frac{\ln d}{\ln d+s}} .$$

For a uniquely decodable code, $\sum_{i=1}^{c} d^{-m_i} \leq 1$ [1, p. 47]; thus

$$A(s) \sum_{i=1}^{c} p_i e^{sm_i} \geq A(s) \left(\sum_{i=1}^{c} p_i^{\frac{\ln d}{\ln d+s}} \right)^{\frac{\ln d+s}{\ln d}} . \qquad (2)$$

Consequently the s* corresponding to a uniquely decodable code is not greater than s_u, defined as the largest s such that the right member of (2) is less than or equal to 1. In general s_u is not achievable because the m_i's must take integer values. The right member of (2) is a convex function of s; at s=0, its value is 1, while the value of its derivative is equal to the entropy (base d) of the source alphabet minus the mean intermission time. Thus if this latter quantity is negative then s_u is positive, and conversely except in the degenerate case where the right member of (2) is identical to 1.

For a given code, C, we denote the corresponding $A(s) \sum_{i=1}^{c} p_i e^{sm_i}$ by f(C,s). f(C,s) is the Laplace-Stieltjes transform of the probability distribution of an intermission time minus a codeword length, thus it is convex in s and $f(C,0) = 1$.

The iterative algorithm to find a prefix condition code with largest s^* is as follows (see also Figure 1).

```
1. Choose any $s_0$ in $[0,\infty]$ (a good choice is $s_0 = s_u$)
2. $j \leftarrow 0$
3. $j \leftarrow j+1$
4. Use the algorithm of Section II to find a code $C_j$ minimizing
   $\sum_{i=1}^{c} p_i e^{sm_i}$ for $s = s_{j-1}$.
5. Compute the $s^*$ corresponding to $C_j$. Denote it by $s_j$.
6. If $s_j \neq s_{j-1}$ then go to 3, else stop
```

Of course we must show that this algorithm terminates and that the last code generated is optimal.

First we note that $s_{j+1} \geq s_j$, $j \geq 1$, because

(line 5) $f(C_j, s_j) \leq 1$ and

(line 4) $f(C_{j+1}, s_j) \leq f(C_j, s_j)$

and the definition of s_{j+1}. Secondly, we observe that the number of codes that can possibly be generated by the algorithm of Section II is finite, if only because all codeword lengths must be smaller than c. These two remarks insure that the algorithm will terminate in a finite time.

Let C_ℓ and s_ℓ denote respectively the last generated code and its s^*, while $\hat{s}$ denotes the largest achievable s^*. We must show that $s_\ell = \hat{s}$.

For the sake of brevity (the complete proof appears in [3]) we assume here that $f(C,s)$ is not identical to 1 when C is an optimal code; this happens when the intermission times are equal with probability one to an integer t, and when the number of source symbols with non zero probabilities lies between $d^t - d + 2$ and d^t. With this restriction $f(C,s)$ is strictly convex, $f(C,0) = 1$ and $f(C,\hat{s}) \leq 1$ when C is an optimal code. If $s_\ell = \infty$, then C_ℓ is optimal. If $s_\ell < \infty$, then $f(C_\ell, s_\ell) = 1$. If $s_\ell > 0$ and $s_\ell < \hat{s}$ then by strict convexity $f(C, s_\ell) < 1$ when C is an optimal code, and C_ℓ may not be the last generated code. If $s_\ell = 0$ then $\frac{d}{ds} f(C_\ell, s)\big|_{s=0} \geq 0$ and, as we have seen in Section II, C_ℓ is a Huffman code and therefore minimizes $\frac{d}{ds} f(C,s)\big|_{s=0}$ over all uniquely decodable codes. Thus $\frac{d}{ds} f(C,s)\big|_{s=0} \geq 0$ for all uniquely decodable codes, and by the strict convexity argument $\hat{s} = 0$.

This algorithm was tested on a typical 128 symbol alphabet, for Poisson and deterministic interemission processes, with $s_0 = s_u$. Convergence was fast (1-2 iterations) in the Poisson case, slower (3-10 iterations) in the deterministic case. The relative difference between $\hat{s}$ and s^* corresponding to an ordinary Huffman code ranges from ∞(deterministic, light traffic) to 10% (Poisson, light traffic), to 0 (heavy traffic).

Acknowledgements

The author is grateful to Professor R. G. Gallager and Professor J. L. Massey for their constructive comments on this research and to Professor J. Wolf for mentioning reference [7].

References

[1] R. G. Gallager, Information Theory and Reliable Communication, New York: Wiley, 1968.

[2] D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes", Proc. IRE, Vol. 40, pp. 251-252, September 1952.

[3] P. A. Humblet, "Source Coding for Communication Concentrators", Ph.D. Dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, Mass., 1978.

[4] F. Jelinek, "Buffer Overflow in Variable Length Coding of Fixed Rate Sources", IEEE Trans. Inform. Theory, Vol. IT-14, pp. 490-501, 1968.

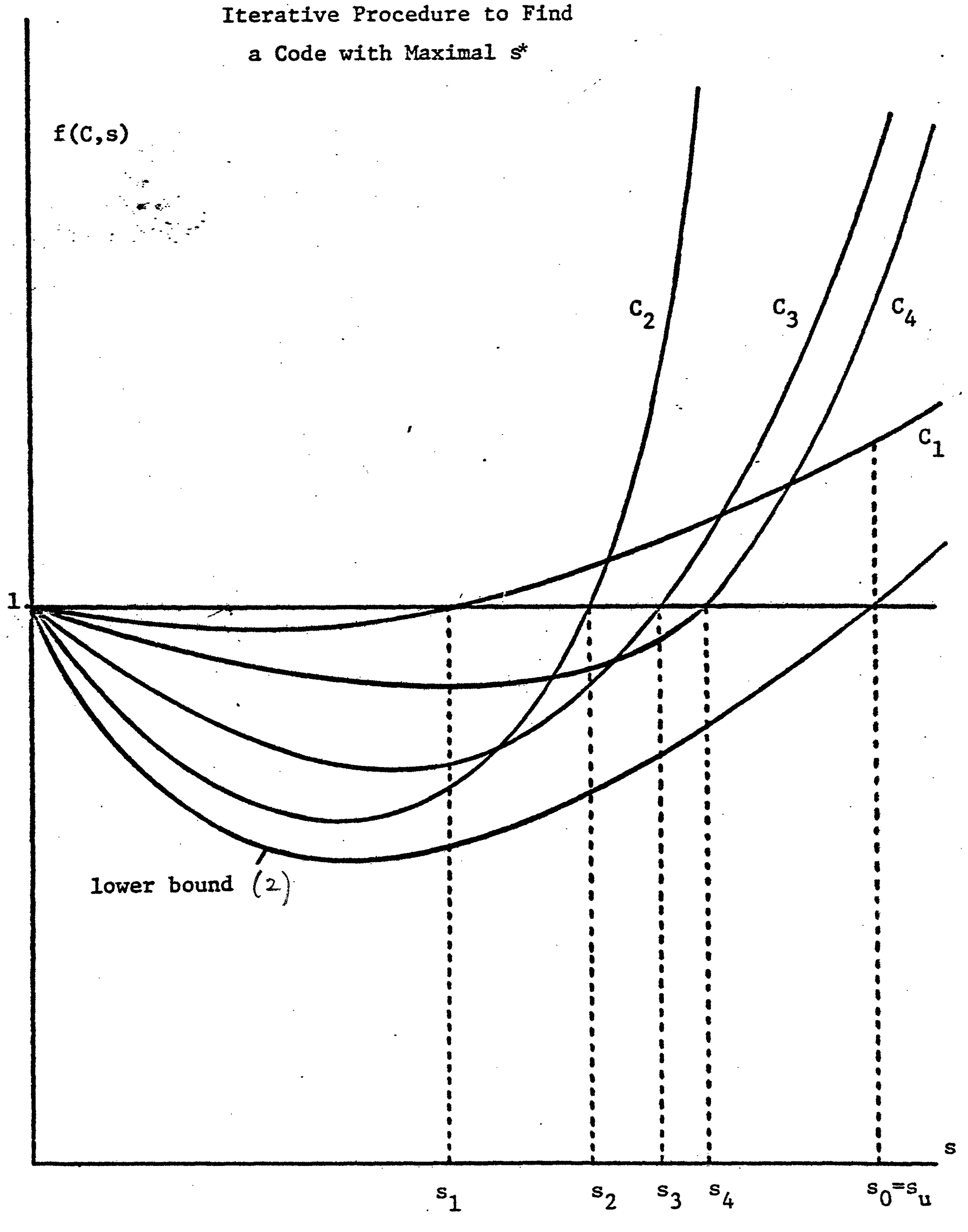
[5] F. Jelinek and K. Schneider, "On Variable Length to Block Coding", IEEE Trans. Inform. Theory, Vol. IT-18, pp. 765-774, November 1972

[6] A. D. Wyner, "On the Probability of Buffer Overflow Under an Arbitrary Bounded Input-Output Distribution", SIAM J. Appl. Math., Vol. 27, pp. 544-570, December 1974.

[7] T. C. Hu, D. J. Kleitman and J. T. Tamaki, "Binary Trees Optimum Under Various Criteria", SIAM J. Appl. Math., Vol. 37, No. 2, October 1979.

FIGURE 1

Iterative Procedure to Find a Code with Maximal s*


f(C,s)
C_2
C_3
C_4
C_1
lower bound (2)
s
s_1
s_2
s_3
s_4
$s_0=s_u$