A decoding algorithm for linear codes overs Z4

Manish Goel and B.Sundar Rajan
EE Department, Indian Institute of Technology, New Delhi, India. E-mail: bsrajan@ee.iitd.emet.in

Abstract — **A decoding algorithm for linear** codes **over** *Z4* = {0,1,2,3}, **the ring of integers modulo 4, is given which gives the codewords that is closest to the received vector in Lee distance.**

I. INTRODUCTION

A linear code over *Z4* is same as a group code over the 4-element cyclic group and can be defined by a check-matrix [1]. The algorithm proposed is similar to the one given in [2] for soft-decision decoding of binary linear codes. First a trellis is cons tructed using the check matrix of the linear code over *Z4* under consideration using Wolf's trellis construction[3]. There is one to one correspondence between the set of paths from the start node to the goal node and the set of codewords. Hence, the pr oblem of decoding is same as finding the path in the trellis which is closest to the received vector in Lee distance. The search is guided by an evaluating function / = *g + h* defined on each node, where g depends only on the past and h(called heuristic func tion) is an estimate on the set of possible futures. The nodes with minimum value of f is given the first priority for expanding. The most important factor in the efficeincy of the algorithm depends on the complexity of the heuristic function. We define a heuristic function which can be easily computed with the worst case complexity of 4n searches over Lee weight distribution , where n is the length of the code.

II. HEURISTIC FUNCTION '/I' AND COST FUNCTION '/'

Let r = (ro, ri,. .., r_n-*i)* be the received vector. A cost function *f(m,t)* for any node m at level t, (0 <_*t* <_*n* — 1) is defined by

$$f(m,t) = g(m,t) + h(m,t) \quad (1)$$

where *g(m,t)* and *h(m,t)* are defined as follows

$$g(m,t) = \sum_{i=0}^{t} LW(r_i - c_i) \quad (2)$$

where *LW(x)* = Lee weight of x and $c_p(t)$ = (co, ci,. . ., *ct)* is the path leading to that node and

$$h(m,t) = \min_{x \in X(t)} \left(\sum_{i=t+1}^{n-1} LW(n - x_t) \right) \quad (3)$$

where

$x(t) = \{(co,ci,...,c_b,xt+i,...,x_n\text{-}i)/x_i+i,...,x_n\text{-}i <= Zi,LW(x) \le L_s\}$ and L_s is the set of all Lee weights of the codewords. The decoding algorithm given below gives the codeword which is closest to the received vector in Lee distance.

III. THE DECODING ALGORITHM

Step 1: Create a list called OPEN and let start node be the only element in OPEN.

Step 2: Select and remove the first node from OPEN and call it node m. If m is the goal node exit successfully, and the path history of node m is the output of the decoder.

Step 3: Expand node m, generating next level nodes which are successors of the node m. This expanding operation consists of

(a) Obtaining all successor nodes and computing g and h values of all the successor nodes.

(b) For each of the successor storing the path followed so far (called path history) from the start node.

(c) Storing all successors in OPEN. *em[(d)] Arranging the nodes in OPEN in the increasing order of their f value. (For nodes with equal value of f arrange them in the decreasing order of the levels of the nodes. For nodes with equal values of f and in the same level arrange in the increasin g order of their g'value.)*

Step 4: Go to Step 2.

IV. A SIMPLE PROCEDURE TO CALCULATE *h(m,t)*

The following theorem leads to a simple procedure which gives the value of h(m,t) without actually carrying out the minimization.

Theorem 1 *For a chosen node m, which is say at level t, let* $\underline{ct}$ = (co,ci,... ,c_t), *and* l_c = *LW(*$\underline{ct}$*). Also let* $\underline{r}$± = (r_{t+1},. .. ,7v_i), *and* l_r = *LW(*$\underline{rt}$*). Then, h(m,t) =* \h*\, *(absolute value of h*) wh ere h* is the least integer such that* $l_c + l_r \pm h^* \in L_s$

For any node m, the possible values for h(m,t) are 0,1,2,. .. ,2(n-t-l). From Theorem 1, it follows that one can find h(m,t) for each node, by successively assuming values from 0 to 2(n-t-l) and matching with elements of L_s to check whether $l_c + l_r$ *pmh(m,t)* € L_a and stopping at the first value for which $l_c + l_r \pm h(m,t) \pounds L_s$ Clearly, the worst case for matching effort is for the start node for which the number of matching efforts may be 2n. The complexity of each search for matching depe nds on the Lee weight distribution of the code. If the code has codewords of specific weights only then the search becomes simple. For instance, for constant Lee weight codes the search is to test for that constant weight or zero. If minimum Lee weight is known then one checks only for zero and all weights starting from minimum Lee weight to four times the length of the code. In the absence of any knowledge of Lee weight distribution one is compelled to check for all weights from zero to twice the length of the code.

REFERENCES

[1] G.Caire and E.Biglieri, "Linear Codes over Cyclic Groups", (Private Communication).

[2] Y.S.Hahn, C.R.P. Hartmann and C.C. Chen, "Efficient priority-first search m axìmum-likelihood soft-decision decoding of linear block codes", IEEE Transactions on Information Theory, Vol. IT-35, No. 5, 1514-1523, 1993.

[3] J.K. Wolf, "Efficient maximum-likelihood decoding of linear block codes using a trellis", IEEE Transactions on Information Theory, Vol. IT-24, 76-80, 1978.