Therefore, from Lemma 1 and from (32)–(34), we have

$$\mathcal{E}P_{e,\text{Ed}}(C,\mathcal{C}_o) \le P_e(C) + \exp\{-NE_r(R_o)\}.$$

Then, the second and third assertion follow since there must exist a C_o which satisfies

$$\sum_{\boldsymbol{y}} \prod_{\boldsymbol{x}\in C_o} \chi[\boldsymbol{y} \notin S(\boldsymbol{x}, N\delta_o)] \le 2^{n+1} \exp\{-(1-M/M_o)e^{N(\eta/2-\eta_N)}\}$$

and

$$P_{e,\text{Ed}}(C,C_o) - P_e(C) \le 2\exp\{-NE_r(R_o)\}$$

simultaneously.

V. Proof of Corollary 3.1

Let η_1 be a given positive number and let C be an approximately optimal code which satisfies

$$P_{\text{ers,Th}}(C) \le \exp\{-NE_{sp}(R_o)\}$$

and

$$P_{\text{uer,Th}}(C) \le \exp\{-N[E_{\text{Th}}(R,R_o) - \eta_1]\}.$$

From the assumption on the exponent functions, we may assume that $P_{\text{uer,Th}}(C) \le P_{\text{ers,Th}}(C)$. Thus from Lemma 1, we have

$$P_e(C) \le 2P_{\text{ers,Th}}(C). \tag{35}$$

Combining (35) and Theorem 3, we have, for a given η_2

$$P_{\text{uer,Ed}}(C,C_o) \le \exp\{-N[E_{\text{Th}}(R,R_o) - \eta_1]\}$$

and

$$P_{\text{ers,Ed}}(C,C_o) \le \exp\{-N[E_{sp}(R_o) - \eta_2]\}$$

for a sufficiently large N, where we used $E_{sp}(R_o) = E_r(R_o)$ for $R_o \ge R_{cr}$. This completes the proof of the corollary.

Acknowledgment

The authors wish to thank Prof. S. Hirasawa and Dr. T. Niinomi of Waseda University for inspiring discussions of erasures-and-errors decoding. They also wish to thank the members of their laboratory, especially H. Tsuchiya, for providing a good computer environment, and anonymous referees for providing good suggestions and opinions.

References

[1] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic, 1981.

[2] P. Delsarte and P. Piret, "Algebraic constructions of Shannon codes for regular channels," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 593–599, July 1982.

[3] G. D. Forney, Jr., *Concatenated Codes*. Boston, MA: MIT Press, 1966.

[4] ——, "Exponential error bounds for erasure, list, and decision feedback scheme," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 206–220, Mar. 1968.

[5] M. P. C. Fossorier and S. Lin, "Computationally efficient soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol. 42, pp. 738–750, May 1996.

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

[7] T. Hashimoto, "On the coded ARQ with the generalized Viterbi algorithm," in *Proc. IEEE Information Theory Workshop* (Salvador, Brazil, June 21–26, 1992), pp. 45–47.

[8] ——, "A coded ARQ scheme with the generalized Viterbi algorithm," *IEEE Trans. Inform. Theory*, vol. 39, pp. 423–432, Mar. 1993.

[9] ——, "On the error exponent of convolutionally coded ARQ", *IEEE Trans. Inform. Theory*, vol. 40, pp. 567–575, Mar. 1994.

[10] F. Jelinek, *Probabilistic Information Theory: Discrete and Memoryless Models*. New York: McGraw-Hill, 1968.

[11] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum likelihood decoding of linear block codes with algebraic decoder," *IEEE Trans. Inform. Theory*, vol. 40, pp. 320–327, Mar. 1994.

[12] T. Kløve and M. Miller, "The detection of errors after error-correction decoding," *IEEE Trans. Commun.*, vol. COM-32, pp. 511–517, May 1984.

[13] ——, "Using codes for error correction and detection," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 868–870, Nov. 1984.

[14] T. Kløve and V. I. Korzhik, *Error-Detecting Codes*. Norwell, MA: Kluwer, 1995.

[15] B. D. Kudryashov, "Viterbi algorithm for convolutional code decoding in a system with decision feedback," *Probl. Pered. Inform.*, vol. 20, pp. 18–26, no. 2, 1984. (Also see B. D. Kudryashov, "Error probability for repeat-request systems with convolutional codes," *IEEE Trans. Inform. Theory* vol. 39, pp. 1680–1684, Sept. 1993.)

[16] ——, "Convolutional-block coding in channels with decision feedback," *Probl. Pered. Inform.*, vol. 21, no. 1, pp. 17–27, 1985.

[17] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[18] L. K. Rasmussen and S. B. Wicker, "Trellis-coded, type-I hybrid-ARQ protocols based on CRC error-detecting codes," *IEEE Trans. Commun.*, vol. 43, pp. 2569–2575, Oct. 1995.

[19] H. Yamamoto and K. Itoh, "Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 540–547, Sept. 1980.

[20] G. W. Zeoli, "Coupled decoding of block-convolutional concatenated codes," *IEEE Trans. Commun.*, vol. COM-21, pp. 219–226, Mar. 1973.

The Zero-Guards Algorithm for General Minimum-Distance Decoding Problems

Yunghsiang S. Han, *Member, IEEE*, and
Carlos R. P. Hartmann, *Fellow, IEEE*

***Abstract*— In this correspondence we present some properties of an improved version of the Zero-Neighbors algorithm—the Zero-Guards algorithm. These properties can be used to find a Zero-Guards. A new decoding procedure using a Zero-Guards is also given.**

***Index Terms*—Decoding, linear block codes, minimum-distance decoding.**

I. Introduction

Minimum-distance decoding for a linear block code has been proved to be an NP-hard computational problem [1]. The complexity of the best known decoding algorithms is determined by $\min(2^k, 2^{n-k})$, where n is the code length and k is the number of information bits [3]. The Zero-Neighbors algorithm [3] provides a better method for solving the problem. The algorithm uses the concept

Manuscript received June 24, 1996; revised March 21, 1997. This work was supported in part by National Science Council ROC under Grant NSC 84-2213-E-211-004.

Y. S. Han is with the Department of Electronic Engineering, Hua Fan College of Humanities and Technology, Taipei Hsien, Taiwan, ROC.

C. R. P. Hartmann is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244-4100 USA.

Publisher Item Identifier S 0018-9448(97)05614-9.

of Zero-Neighbors—a special set of codewords. Only these codewords need to be stored and used in the decoding procedure. The size of a Zero-Neighbors is very small compared with $\min(2^k, 2^{n-k})$ for $n \gg 1$ and a wide range of code rates $R = k/n$. Recently, an improvement of the Zero-Neighbors algorithm, the Zero-Guards algorithm (ZGA), was presented in [2] and [4]. The ZGA further reduces the number of codewords to be stored. The special set of these codewords is called a Zero-Guards. Thus the size of a Zero-Guards is the main factor that determines the complexity of the algorithm.

In this correspondence, we investigate the properties of a Zero-Guards. These properties can be used to find a Zero-Guards efficiently. Moreover, we also presented a new decoding procedure using a Zero-Guards that is much simpler than the one given in [3]. In Section II, we briefly review the Zero-Neighbors algorithm. In Section III, we give a description of the Zero-Guards algorithm and a new decoding procedure using a Zero-Guards. In the next section, properties of a Zero-Guards are presented. Simulation results and conclusions are given in Section V.

II. The Zero-Neighbors Algorithm

In this section, we briefly describe the Zero-Neighbors algorithm. First we give some definitions.

Let $\boldsymbol{Z}$ be the set of all the binary vectors of length n, and let $\boldsymbol{C} \subset \boldsymbol{Z}$ be a binary linear block code. If $\boldsymbol{x} \in \boldsymbol{Z}$, we call $\boldsymbol{x}$ a vector in the space $\boldsymbol{Z}$. Let $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ denote the Hamming distance between $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{Z}$. Let $w(\boldsymbol{x}) = d(\boldsymbol{x}, \boldsymbol{0})$ denote the Hamming weight of $\boldsymbol{x}$ and let $\oplus$ denote the modulo-2 addition. Furthermore, let $d_{\min}$ be the minimum nonzero weight of codewords in $\boldsymbol{C}$.

Definition 1: The domain $D(\boldsymbol{c})$ of a codeword $\boldsymbol{c} \in \boldsymbol{C}$ is the set of all $\boldsymbol{x} \in \boldsymbol{Z}$ such that $d(\boldsymbol{x}, \boldsymbol{c}) \le d(\boldsymbol{x}, \boldsymbol{c}')$ for all $\boldsymbol{c}' \in \boldsymbol{C}$.

Definition 2: The vicinity $B(\boldsymbol{x})$ of $\boldsymbol{x} \in \boldsymbol{Z}$ is the set of all $\boldsymbol{y} \in \boldsymbol{Z}$ such that $d(\boldsymbol{x}, \boldsymbol{y}) = 1$. The domain frame $G(\boldsymbol{c})$ of a codeword $\boldsymbol{c} \in \boldsymbol{C}$ is the set

$$G(\boldsymbol{c}) = \bigcup_{\boldsymbol{x} \in D(\boldsymbol{c})} B(\boldsymbol{x}) - D(\boldsymbol{c}).$$

Definition 3: A Zero-Neighbors is a set N_0 of codewords such that

$$G(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in N_0} D(\boldsymbol{c})$$

where

$$|N_0| = \min\left\{|N| \,|\, N \subset \boldsymbol{C},\, G(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in N} D(\boldsymbol{c})\right\}.$$

It can be shown that if $\boldsymbol{x} \notin D(\boldsymbol{0})$, then there exists a $\boldsymbol{c} \in N_0$ such that $w(\boldsymbol{x} \oplus \boldsymbol{c}) < w(\boldsymbol{x})$. Thus the Zero-Neighbors algorithm is as follows:

Algorithm: Let $\boldsymbol{y} = \boldsymbol{y}_0 \in \boldsymbol{Z}$ be the received vector to be decoded. At the ith step of the algorithm we calculate $w(\boldsymbol{y}_{i-1} \oplus \boldsymbol{c})$ for all $\boldsymbol{c} \in N_0$. If there exists a $\boldsymbol{c}_i \in N_0$ such that $w(\boldsymbol{y}_{i-1} \oplus \boldsymbol{c}_i) < w(\boldsymbol{y}_{i-1})$, we set $\boldsymbol{y}_i = \boldsymbol{y}_{i-1} \oplus \boldsymbol{c}_i$ and go to the next step; otherwise, the algorithm terminates. If the algorithm terminates at the $(m+1)$th step, then

$$\boldsymbol{y}_m = \boldsymbol{y} \oplus \sum_{i=1}^{m} \boldsymbol{c}_i \in D(\boldsymbol{0})$$

and can be taken as a coset leader of minimum weight, while

$$\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}_i \in \boldsymbol{C}$$

is a codeword that is one of the closest to $\boldsymbol{y}$.

We need only to store the codewords in a Zero-Neighbors to accomplish this algorithm. It can be shown that the number of steps $m \le n$. A more complex decoding procedure based on the syndrome of the received vector is given in [3]. The number of decoding steps is $m \le n - k - \lfloor d_{\min}/2 \rfloor$ for this decoding procedure, where $\lfloor a \rfloor$ denotes the integral part of a. We do not describe the procedure here since we will give a new decoding procedure in the next section, which also has the number of decoding steps $m \le n - k - \lfloor d_{\min}/2 \rfloor$.

III. The Zero-Guards Algorithm

In this section, we will describe a minimum-distance decoding algorithm that is similar to the Zero-Neighbors algorithm except that we use the concept of Zero-Guards instead of Zero-Neighbors.

Definition 4: A vector $\boldsymbol{v}$ is an immediate descendant of $\boldsymbol{x}$ if and only if $\boldsymbol{v}$ can be obtained from $\boldsymbol{x}$ by changing one nonzero component to zero. A vector $\boldsymbol{v}$ is a descendant of $\boldsymbol{x}$ if and only if there is a chain $\boldsymbol{x}_0 = \boldsymbol{x}, \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n = \boldsymbol{v}$ such that, for each i, $\boldsymbol{x}_i$ is an immediate descendant of $\boldsymbol{x}_{i-1}$. Furthermore, if $\boldsymbol{v} \ne \boldsymbol{x}$, then $\boldsymbol{v}$ is a proper descendant of $\boldsymbol{x}$ [5].

Definition 5: The frontier $F(\boldsymbol{0})$ of $\boldsymbol{0}$ is the set of all $\boldsymbol{x} \in \boldsymbol{Z}$ such that all its proper descendants belong to $D(\boldsymbol{0})$ and $\boldsymbol{x} \notin D(\boldsymbol{0})$.

Definition 6: A Zero-Guards (ZG) is a set $RN_0 \subset \boldsymbol{C}$ of codewords such that

$$F(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in RN_0} D(\boldsymbol{c})$$

where

$$|RN_0| = \min\left\{|N| \,|\, N \subset \boldsymbol{C},\, F(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in N} D(\boldsymbol{c})\right\}.$$

In other words, the set of domains of codewords in RN_0 forms a minimum covering of $F(\boldsymbol{0})$. It is easy to see that RN_0 always exists, since the number of all such subsets $N \subset \boldsymbol{C}$ is finite.

It is not difficult to see that $F(\boldsymbol{0}) \subset G(\boldsymbol{0})$. Consequently, the number of codewords in Zero-Guards is less than or equal to that in Zero-Neighbors.

Next we give the main theorem that the ZGA is based on.

Lemma 1: Let $\boldsymbol{x} \in \boldsymbol{Z}$ and $\boldsymbol{x} \notin D(\boldsymbol{0})$. Then there exists a descendant $\boldsymbol{v}$ of $\boldsymbol{x}$ such that $\boldsymbol{v} \in F(\boldsymbol{0})$.

Proof: Let $M(\boldsymbol{x}) = \{\boldsymbol{v} | \boldsymbol{v} \in \boldsymbol{Z}, \boldsymbol{v}$ be a descendant of $\boldsymbol{x}$ and $\boldsymbol{v} \notin D(\boldsymbol{0})\}$. Thus $M(\boldsymbol{x}) \ne \emptyset$, since at least $\boldsymbol{x} \in M(\boldsymbol{x})$. Then any vector of minimum weight in $M(\boldsymbol{x})$ belongs to $F(\boldsymbol{0})$. □

Theorem 1: $\boldsymbol{x} \notin D(\boldsymbol{0})$ if and only if there exists a $\boldsymbol{c} \in RN_0$ such that $w(\boldsymbol{x} \oplus \boldsymbol{c}) < w(\boldsymbol{x})$.

Proof: First, assume that $\boldsymbol{x} \notin D(\boldsymbol{0})$. From Lemma 1, there exists a descendant of $\boldsymbol{x}$, named $\boldsymbol{v}$, $\boldsymbol{v} \in F(\boldsymbol{0})$. Consider a $\boldsymbol{c} \in RN_0$ such that $\boldsymbol{v} \in D(\boldsymbol{c})$. Hence

$$w(\boldsymbol{x} \oplus \boldsymbol{c}) = d(\boldsymbol{x}, \boldsymbol{c}) \le d(\boldsymbol{x}, \boldsymbol{v}) + d(\boldsymbol{v}, \boldsymbol{c}) < d(\boldsymbol{x}, \boldsymbol{v}) + d(\boldsymbol{v}, \boldsymbol{0}) = w(\boldsymbol{x}).$$

Next, assume $\boldsymbol{x} \in D(\boldsymbol{0})$. Then $d(\boldsymbol{x}, \boldsymbol{0}) \le d(\boldsymbol{x}, \boldsymbol{c})$ for all $\boldsymbol{c} \in \boldsymbol{C}$. Thus $w(\boldsymbol{x}) \le w(\boldsymbol{x} \oplus \boldsymbol{c})$. Therefore, there is no $\boldsymbol{c} \in RN_0$ such that $w(\boldsymbol{x} \oplus \boldsymbol{c}) < w(\boldsymbol{x})$. □

Obviously, if $w(\boldsymbol{c})$ is even, then $w(\boldsymbol{x}) - w(\boldsymbol{x} \oplus \boldsymbol{c})$ is even, too. Thus we have the following corollary.

Corollary 1: If all codewords in a Zero-Guards are of even weight and $\boldsymbol{x} \notin D(\boldsymbol{0})$, then there exists a $\boldsymbol{c} \in RN_0$ such that $w(\boldsymbol{x} \oplus \boldsymbol{c}) \le w(\boldsymbol{x}) - 2$.

The following algorithm and arguments are similar to those in [3] except that we use the concept of Zero-Guards instead of Zero-Neighbors.

Algorithm 1: Let $\boldsymbol{y} = \boldsymbol{y}_0 \in \boldsymbol{Z}$ be the received vector to be decoded. At the ith step of the algorithm we calculate $w(\boldsymbol{y}_{i-1} \oplus \boldsymbol{c})$ for all $\boldsymbol{c} \in RN_0$. If there exists a $\boldsymbol{c}_i \in RN_0$ such that $w(\boldsymbol{y}_{i-1} \oplus \boldsymbol{c}_i) < w(\boldsymbol{y}_{i-1})$, we set $\boldsymbol{y}_i = \boldsymbol{y}_{i-1} \oplus \boldsymbol{c}_i$ and go to the next step; otherwise, the algorithm terminates. If the algorithm terminates at the $(m+1)$th step, then

$$\boldsymbol{y}_m = \boldsymbol{y} \oplus \sum_{i=1}^{m} \boldsymbol{c}_i \in D(\boldsymbol{0})$$

and can be taken as a coset leader of minimum weight, while

$$\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}_i \in \boldsymbol{C}$$

is a codeword that is one of the closest to $\boldsymbol{y}$.

We call the algorithm the Zero-Guards algorithm. Only the codewords in a Zero-Guards must be stored in order to accomplish this algorithm. The algorithm will stop when $\boldsymbol{y}_m \in D(\boldsymbol{0})$. Thus if $w(\boldsymbol{y}_m) \le \lfloor d_{\min}/2 \rfloor$, the algorithm stops immediately. Since at each step of the algorithm the weight of $\boldsymbol{y}$ decreases at least by one, the number of steps is

$$m \le w(\boldsymbol{y}) - \lfloor d_{\min}/2 \rfloor \le n - \lfloor d_{\min}/2 \rfloor.$$

Moreover, it follows from Corollary 1 that for codes with only even-weight codewords, each step of the algorithm decreases the weight of $\boldsymbol{y}$ at least by 2; therefore, in this case

$$m \le \lfloor (w(\boldsymbol{y}) - \lfloor d_{\min}/2 \rfloor)/2 \rfloor \le \lfloor (n - \lfloor d_{\min}/2 \rfloor)/2 \rfloor.$$

Next, we give another algorithm that is simpler than that presented in [3], which is based on the syndrome of the received vector.

Algorithm 2: Let $\boldsymbol{G}$ be a generating matrix of a systematic code $\boldsymbol{C}$. Let $\boldsymbol{y} = (y_0, y_1, \cdots, y_{n-1})$ and

$$\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1}) = (y_0, y_1, \cdots, y_{k-1})\boldsymbol{G}.$$

Then $c_i = y_i$ for $0 \le i \le k-1$. Take

$$\boldsymbol{x} = \boldsymbol{c} \oplus \boldsymbol{y} = (x_0, x_1, \cdots, x_{n-1}).$$

Thus $x_i = 0$ for $0 \le i \le k-1$. It is clear that $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same coset. Instead of decoding $\boldsymbol{y}$ directly, we decode $\boldsymbol{x}$ as the process in Algorithm 1. Assume the process in Algorithm 1 terminates at the $(m+1)$th step; we then have

$$\boldsymbol{y}_m = \boldsymbol{x} \oplus \sum_{i=1}^{m} \boldsymbol{c}_i \in D(\boldsymbol{0}).$$

Then

$$\boldsymbol{c} \oplus \sum_{i=1}^{m} \boldsymbol{c}_i$$

is a codeword that is one of the closest to $\boldsymbol{y}$.

Since $w(\boldsymbol{x}) \le n-k$, the number of decoding steps

$$m \le n - k - \lfloor d_{\min}/2 \rfloor.$$

IV. Some Properties of a Zero-Guards

In this section, we present some properties of the frontier $F(\boldsymbol{0})$ and the Zero-Guards that can help to find RN_0.

Lemma 2: Let

$$S(\boldsymbol{x}, a) = \{\boldsymbol{v} | \boldsymbol{v} \in \boldsymbol{Z},\ w(\boldsymbol{v}) = a \text{ and } \boldsymbol{v} \text{ be a descendant of } \boldsymbol{x}\}.$$

Then $\boldsymbol{x} \in F(\boldsymbol{0})$ if and only if $\boldsymbol{x} \notin D(\boldsymbol{0})$ and $S(\boldsymbol{x}, w(\boldsymbol{x})-1) \subset D(\boldsymbol{0})$.

Proof: If $\boldsymbol{x} \in F(\boldsymbol{0})$, then by definition $\boldsymbol{x} \notin D(\boldsymbol{0})$ and $S(\boldsymbol{x}, w(\boldsymbol{x})-1) \subset D(\boldsymbol{0})$. Assume now that $\boldsymbol{x} \notin D(\boldsymbol{0})$ and $S(\boldsymbol{x}, w(\boldsymbol{x})-1) \subset D(\boldsymbol{0})$. By [5, Theorem 3.9], if $\boldsymbol{v} \in D(\boldsymbol{0})$, then all its descendants also belong to $D(\boldsymbol{0})$. Thus $\boldsymbol{x} \notin D(\boldsymbol{0})$ and all its descendants belong to $D(\boldsymbol{0})$. □

Lemma 3: If $\boldsymbol{x} \in F(\boldsymbol{0})$, then there exists at least one $\boldsymbol{c} \in \boldsymbol{C}$ such that $\boldsymbol{x} \in D(\boldsymbol{c})$ and $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$.

Proof: Because $\boldsymbol{x} \notin D(\boldsymbol{0})$ there exists at least one $\boldsymbol{c} \in \boldsymbol{C}$ such that $\boldsymbol{x} \in D(\boldsymbol{c})$. Suppose $\boldsymbol{x}$ is not a descendant of $\boldsymbol{c}$. Then $\boldsymbol{x}$ should at least differ from $\boldsymbol{c}$ in a position where $\boldsymbol{c}$ has 0. Let $\boldsymbol{v}$ be the immediate descendant of $\boldsymbol{x}$ that differs from $\boldsymbol{x}$ in the position just mentioned. Then $d(\boldsymbol{x}, \boldsymbol{c}) > d(\boldsymbol{v}, \boldsymbol{c})$ and $w(\boldsymbol{v}) = w(\boldsymbol{x}) - 1$. Since $d(\boldsymbol{x}, \boldsymbol{c}) < w(\boldsymbol{x})$, $w(\boldsymbol{v}) \ge d(\boldsymbol{x}, \boldsymbol{c})$. Thus $w(\boldsymbol{v}) > d(\boldsymbol{v}, \boldsymbol{c})$ which contradicts $\boldsymbol{v} \in D(\boldsymbol{0})$. Therefore, $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$. □

Lemma 4: Let $\boldsymbol{x} \in F(\boldsymbol{0})$. If $d(\boldsymbol{x}, \boldsymbol{c}) < w(\boldsymbol{x})$, then $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$.

Proof: The proof is similar to that in Lemma 3. □

Lemma 5: For every $\boldsymbol{c} \in \boldsymbol{C}$ and $\boldsymbol{c} \ne \boldsymbol{0}$ there exists a descendant $\boldsymbol{v}$ of $\boldsymbol{c}$ such that $\boldsymbol{v} \in F(\boldsymbol{0})$.

Proof: For any $\boldsymbol{c} \in \boldsymbol{C}$, $\boldsymbol{c} \notin D(\boldsymbol{0})$, and $\boldsymbol{c} \in D(\boldsymbol{c})$. From Lemma 1 there exists a descendant $\boldsymbol{v}$ of $\boldsymbol{c}$ such that $\boldsymbol{v} \in F(\boldsymbol{0})$. □

Lemma 6: If $\boldsymbol{c} \in RN_0$, then there exists an $\boldsymbol{x} \in F(\boldsymbol{0})$ such that $\boldsymbol{x} \in D(\boldsymbol{c})$ and $\boldsymbol{x} \notin D(\boldsymbol{c}')$, $\boldsymbol{c}' \ne \boldsymbol{c}$, $\boldsymbol{c}' \in RN_0$.

Proof: Assume that there is no $\boldsymbol{x} \in F(\boldsymbol{0})$ such that $\boldsymbol{x} \in D(\boldsymbol{c})$ and $\boldsymbol{x} \notin D(\boldsymbol{c}')$, $\boldsymbol{c}' \ne \boldsymbol{c}$. Then for every $\boldsymbol{x} \in D(\boldsymbol{c})$ and $\boldsymbol{x} \in F(\boldsymbol{0})$ there exists at least one $\boldsymbol{c}' \in RN_0$, $\boldsymbol{c}' \ne \boldsymbol{c}$ such that $\boldsymbol{x} \in D(\boldsymbol{c}')$. Therefore, if we remove $\boldsymbol{c}$ from RN_0 we also have

$$F(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in RN_0} D(\boldsymbol{c}).$$

This contradicts the fact that RN_0 is the minimum set with this property. Therefore, the lemma holds. □

Lemma 7: Let r be the covering radius of the code $\boldsymbol{C}$. If $\boldsymbol{c} \in \boldsymbol{C}$ and $w(\boldsymbol{c}) > 2r+1$, then $\boldsymbol{c} \notin RN_0$.

Proof: Assume $\boldsymbol{c} \in RN_0$. From Lemma 6, there exists an $\boldsymbol{x} \in D(\boldsymbol{c})$ and $\boldsymbol{x} \notin D(\boldsymbol{c}')$, $\boldsymbol{c}' \ne \boldsymbol{c}$. Since $\boldsymbol{x} \in D(\boldsymbol{c})$, $d(\boldsymbol{x}, \boldsymbol{c}) \le r$ and since $\boldsymbol{x} \in F(\boldsymbol{0})$, $w(\boldsymbol{x}) \le r+1$. Hence, $w(\boldsymbol{c}) = w(\boldsymbol{x}) + d(\boldsymbol{x}, \boldsymbol{c}) \le 2r+1$. Thus if $w(\boldsymbol{c}) > 2r+1$, then $\boldsymbol{c} \notin RN_0$. □

Lemma 8: If $\boldsymbol{x} \in F(\boldsymbol{0})$ and there exists a $\boldsymbol{c} \in \boldsymbol{C}$ such that $d(\boldsymbol{x}, \boldsymbol{c}) < d(\boldsymbol{x}, \boldsymbol{c}')$, $\boldsymbol{c} \ne \boldsymbol{c}'$ for all $\boldsymbol{c}' \in \boldsymbol{C}$, then $\boldsymbol{c} \in RN_0$.

Proof: Assume $\boldsymbol{c} \notin RN_0$. Then

$$\boldsymbol{x} \notin \bigcup_{\boldsymbol{c}' \in RN_0} D(\boldsymbol{c}').$$

Hence

$$F(\boldsymbol{0}) \not\subset \bigcup_{\boldsymbol{c}' \in RN_0} D(\boldsymbol{c}')$$

which is a contradiction. □

Theorem 2: Let $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \boldsymbol{C}$ where $\boldsymbol{c}_1$ is a descendant of $\boldsymbol{c}_2$. Then $\boldsymbol{c}_2 \notin RN_0$.

Proof: Assume $\boldsymbol{c}_2 \in RN_0$ and $\boldsymbol{c}_3 = \boldsymbol{c}_1 \oplus \boldsymbol{c}_2$. Then, by Lemma 6, there exists an $\boldsymbol{x} \in F(\boldsymbol{0})$ such that $\boldsymbol{x} \in D(\boldsymbol{c}_2)$ and $\boldsymbol{x} \notin D(\boldsymbol{c}')$, $\boldsymbol{c}' \ne \boldsymbol{c}_2$, $\boldsymbol{c}' \in RN_0$. Furthermore, by Lemma 4, $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}_2$. By Lemma 4, if $d(\boldsymbol{x}, \boldsymbol{c}_1) < w(\boldsymbol{x})$, then $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}_1$. In this case, $\boldsymbol{x} \notin D(\boldsymbol{c}_2)$, which contradicts the fact that $\boldsymbol{x} \in D(\boldsymbol{c}_2)$. Therefore, $d(\boldsymbol{x}, \boldsymbol{c}_1) \ge w(\boldsymbol{x})$. Similarly, we have $d(\boldsymbol{x}, \boldsymbol{c}_3) \ge w(\boldsymbol{x})$. Therefore,

$$d(\boldsymbol{x}, \boldsymbol{c}_2) = d(\boldsymbol{x}, \boldsymbol{c}_1) + d(\boldsymbol{x}, \boldsymbol{c}_3) - w(\boldsymbol{x}) \ge w(\boldsymbol{x}).$$

This contradicts the fact that $\boldsymbol{x} \in D(\boldsymbol{c}_2)$. □

By [3, Theorem 3], if $\boldsymbol{c}_1$ and $\boldsymbol{c}_3$ are in N_0 then $\boldsymbol{c}_2 \notin N_0$; however, by the above theorem, any codeword will not be in RN_0 if any of its nonzero descendant is a codeword.

Theorem 3: All codewords of minimum weight belong to RN_0.

Proof: Let $\boldsymbol{c}$ be a codeword of minimum weight. From Lemma 5, there exists a $\boldsymbol{v} \in F(\boldsymbol{0})$ that is a descendant of $\boldsymbol{c}$. Thus

$$d(\boldsymbol{c}, \boldsymbol{c}') \le d(\boldsymbol{c}, \boldsymbol{v}) + d(\boldsymbol{v}, \boldsymbol{c}') = w(\boldsymbol{c}) - w(\boldsymbol{v}) + d(\boldsymbol{v}, \boldsymbol{c}')$$

where $\boldsymbol{c}' \ne \boldsymbol{c}$ and $\boldsymbol{c}' \in \boldsymbol{C}$. Hence

$$d(\boldsymbol{v}, \boldsymbol{c}') \ge w(\boldsymbol{v}) + [d(\boldsymbol{c}, \boldsymbol{c}') - w(\boldsymbol{c})].$$

Since $\boldsymbol{c}$ is of minimum weight, $d(\boldsymbol{c}, \boldsymbol{c}') - w(\boldsymbol{c}) \ge 0$. Thus $d(\boldsymbol{v}, \boldsymbol{c}') \ge w(\boldsymbol{v})$. But $\boldsymbol{v} \notin D(\boldsymbol{0})$, then $\boldsymbol{v} \in D(\boldsymbol{c})$, and $\boldsymbol{v} \notin D(\boldsymbol{c}')$. Therefore, $d(\boldsymbol{v}, \boldsymbol{c}') > d(\boldsymbol{v}, \boldsymbol{c})$. From Lemma 8, $\boldsymbol{c} \in RN_0$. □

TABLE I
THE COMPARISON OF ZERO-NEIGHBORS AND ZERO-GUARDS

(n,k)	d_{min}	r	2^k	$\|N_0\|$	$\|RN_0\|$
(15,6)	6	5	64	45	30
(15,6)	6	4	64	55	25
(15,7)	5	3	128	63	33
(15,8)	4	4	256	115	15

n: the code length
k: the number of information bits
d_{min}: the minimum distance
r: the covering radius
$|N_0|$: the number of codewords in a Zero-Neighbors
$|RN_0|$: the number of codewords in a Zero-Guards

Corollary 2: If $\boldsymbol{v} \in F(\mathbf{0})$ and $\boldsymbol{v}$ is a descendant of $\boldsymbol{c}$, which is a minimum-weight codeword, then $\boldsymbol{v}$ is not a descendant of any other minimum-weight codeword.

Proof: Suppose $\boldsymbol{v}$ is also a descendant of another minimum-weight codeword $\boldsymbol{c}'$. Then from the proof in Theorem 3, $d(\boldsymbol{v}, \boldsymbol{c}') > d(\boldsymbol{v}, \boldsymbol{c})$ and $d(\boldsymbol{v}, \boldsymbol{c}) > d(\boldsymbol{v}, \boldsymbol{c}')$ which is a contradiction. □

Theorem 4: All descendants of weight $\lfloor d_{\min}/2 \rfloor + 1$ of minimum-weight codewords belong to $F(\mathbf{0})$.

Proof: Assume $d_{\min}$ is even and $d_{\min} = 2t$. Then $\lfloor d_{\min}/2 \rfloor + 1 = t + 1$. For any vector $\boldsymbol{v}$ that is a descendant of a minimum codeword $\boldsymbol{c}$, if the weight of $\boldsymbol{v}$ is $t+1$, then $\boldsymbol{v} \in D(\boldsymbol{c})$. Moreover, any vectors of weight t belong to $D(\mathbf{0})$. Thus $\boldsymbol{v} \in F(\mathbf{0})$. The argument that $d_{\min}$ is odd is similar to that above. □

Theorem 5: If there are m codewords of minimum weight, then there are at least

$$\binom{d_{\min}}{\lfloor d_{\min}/2 \rfloor + 1} \times m$$

vectors of weight $\lfloor d_{\min}/2 \rfloor + 1$ belonging to $F(\mathbf{0})$.

Proof: The theorem follows directly from Corollary 2 and Theorem 4. □

Theorem 6: For an odd minimum-weight linear code, any vector $\boldsymbol{v}$ of weight $\lfloor d_{\min}/2 \rfloor + 1$ that belongs to $F(\mathbf{0})$ is a descendant of a minimum-weight codeword.

Proof: From Lemma 3 we can conclude that the vector $\boldsymbol{v}$ is a descendant of a codeword $\boldsymbol{c}$ and $\boldsymbol{v} \in D(\boldsymbol{c})$. Then

$$w(\boldsymbol{c}) \leq \lfloor d_{\min}/2 \rfloor + 1 + \lfloor d_{\min}/2 \rfloor.$$

Thus $w(\boldsymbol{c}) \leq d_{\min} - 1 + 1 = d_{\min}$. Therefore, $\boldsymbol{c}$ is a minimum-weight codeword. □

From Theorems 5 and 6 we have the following corollary.

Corollary 3: For an (n, k) Hamming code, the number of minimum-weight codewords is $\binom{n}{2} / \binom{3}{2}$.

V. SIMULATION RESULTS AND CONCLUSIONS

Even though up to now we were not able to find a good method with which to estimate the size of RN_0, from the simulation results, $|RN_0|$ is dramatically less than $|N_0|$. Some results are shown in Table I. Just as with the Zero-Neighbors algorithm presented in [3], the ZGA can also be easily generalized for linear codes over GF(p), $p > 2$. To implement the ZGA, we must find a Zero-Guards. As expected, to find a Zero-Guards is an NP-hard problem. However, it needs to be found only once for a given code.

ACKNOWLEDGMENT

The authors wish to thank the reviewer and the Associate Editor T. Kløve for their invaluable suggestions, which we believe have helped to improve the presentation of our correspondence. In addition, the authors wish to thank E. Weinman for her invaluable help on the language check.

REFERENCES

[1] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, “On the inherent intractability of certain coding problems,” *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384–386, May 1978.
[2] C. R. P. Hartmann and L. B. Levitin, “An improvement of the zero-neighbors minimum distance decoding algorithm: The zero-guards algorithm,” in *IEEE Int. Symp. on Information Theory* (Kobe, Japan), 1988.
[3] L. B. Levitin and C. R. P. Hartmann, “A new approach to the general minimum distance decoding problem: The zero-neighbors algorithm,” *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 378–384, May 1985.
[4] L. B. Levitin, M. Naidjate, and C. R. P. Hartmann, “Generalized identity-guards algorithm for minimum distance decoding of group codes in metric space,” in *IEEE Int. Symp. on Information Theory* (San Diego, CA), 1990.
[5] W. W. Peterson, *Error-Correcting Codes*, 2nd ed. Cambridge, MA: MIT Press, 1972.

A Probability-Ratio Approach to Approximate Binary Arithmetic Coding

Linh Huynh and Alistair Moffat

***Abstract*— We describe an alternative mechanism for approximate binary arithmetic coding. The quantity that is approximated is the ratio between the probabilities of the two symbols. Analysis is given to show that the inefficiency so introduced is less than 0.7% on average; and in practice the compression loss is negligible.**

***Index Terms*—Approximate arithmetic coding, bilevel coding, binary arithmetic coding, data compression.**

I. BINARY ARITHMETIC CODING

The need for binary arithmetic coding arises in many applications, including bilevel image compression [1] and general bit-based data compression [2]. In this correspondence, a novel mechanism for approximating the various calculations involved in binary arithmetic coding is described, and error bounds limiting the inaccuracy of the resulting representation are given. We also report experimental results

Manuscript received March 2, 1995; revised February 1, 1997. This work was supported by the Australian Research Council and the Collaborative Information Technology Research Institute. The material in this correspondence was presented in part at the 1994 IEEE Data Compression Conference, Snowbird, UT, March 1994.
L. Huynh is with the Department of Computer Science, RMIT, GPO Box 2476V, Melbourne 3001, Australia, and with the Department of Computer Science, University of Melbourne, Parkville, Victoria 3052, Australia.
A. Moffat is with the Department of Computer Science, University of Melbourne, Parkville, Victoria 3052, Australia.
Publisher Item Identifier S 0018-9448(97)05409-6.