

Properties of Optimum Binary Message-Passing Decoders

Masoud Ardakani, *Member, IEEE*, and
Frank R. Kschischang, *Senior Member, IEEE*

Abstract—We consider a class of message-passing decoders for low-density parity-check (LDPC) codes whose messages are binary valued. We prove that if the channel is symmetric and all codewords are equally likely to be transmitted, an optimum decoding rule (in the sense of minimizing message error rate) should satisfy certain symmetry and isotropy conditions. Using this result, we prove that Gallager’s Algorithm B achieves the optimum decoding threshold among all binary message-passing decoding algorithms for regular codes. For irregular codes, we argue that when the nodes of the message-passing decoder do not exploit knowledge of their decoding neighborhood, optimality of Gallager’s Algorithm B is preserved. We also consider the problem of designing irregular LDPC codes and find a bound on the achievable rates with Gallager’s Algorithm B. Using this bound, we study the case of low error-rate channels and analytically find good degree distributions for them.

Index Terms—Gallager’s algorithm B, irregular codes, low-density parity-check (LDPC) codes, message-passing decoders.

I. INTRODUCTION

Low-density parity-check (LDPC) codes—first introduced by Gallager [1] and rediscovered by MacKay *et al.* [2], [3]—have shown excellent performance in many channels [4]–[7]. The design and analysis of irregular LDPC codes under different decoding algorithms, have been of great interest, e.g., [4]–[9]. Richardson *et al.* [7] introduced density evolution as a tool by which the convergence behavior of LDPC codes may be studied. For a given channel condition, given decoding algorithm and given code ensemble, density evolution allows one to study the evolution of the probability density of messages iteration by iteration and one may determine how successful the given decoding algorithm is.

In this work, we consider a class of message-passing decoding algorithms for LDPC codes with binary-valued messages, a case that we refer to as binary message-passing (BMP). As with other message-passing algorithms, we assume that each message in the decoder carries a belief about the adjacent variable node. Notice that for binary-valued messages, the probability distribution can be expressed by a single parameter. Different parameters can be used to express such a density and these parameters can be translated uniquely to each other. The evolution of this parameter can be plotted in a manner similar to the EXIT charts of [9]. While in the literature, the term EXIT chart is used to describe the evolution of mutual information, we will use error probability to describe the density of messages and call the resulting graphs EXIT charts nevertheless.

The main results of this correspondence are the following. We prove that variable and check node update rules of the optimum BMP decoder (in the sense of minimizing the message error rate) must satisfy the symmetry conditions introduced in [7]. We also propose an isotropy

Manuscript received November 17, 2003; revised December 13, 2004. The material in this correspondence was presented in part at the International Symposium on Turbo Codes and Related Topics, Brest, France, September 2003.

M. Ardakani was with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada. He is now with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: ardakani@ece.ualberta.ca).

F. R. Kschischang is with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: frank@comm.utoronto.ca).

Communicated by A. E. Ashikhmin, Associate Editor for Coding Theory.
Digital Object Identifier 10.1109/TIT.2005.855611

condition and show that the optimum decoder has to satisfy it. We then prove that a decoding algorithm due to Gallager [1], and referred to as Gallager’s Algorithm B in [7], has the optimum threshold for regular LDPC codes among all BMP algorithms. For irregular LDPC codes also, we show that Gallager’s Algorithm B is optimal among all BMP algorithms when the nodes in the factor graph [10] of the code have no knowledge of node degrees in their local neighborhood. If such a knowledge exists, we discuss the possibility of better decoding algorithms. In addition, we find a bound on achievable rates with Gallager’s Algorithm B. Using this bound, for check degree distributions, we find a sufficient condition for convergence on low-error-rate channels, when all the variable nodes are chosen to be degree three. We show that the optimum check degree distribution (associated with the maximum-rate code) which satisfies this sufficient condition is concentrated on one or two degrees.

For fixed (irregular/regular) check degree distribution, the problem of irregular code design with a particular convergence behavior can be reduced to shaping an EXIT chart from the EXIT charts corresponding to regular variable degree codes [11]. This can be done effectively by solving a linear program. To guarantee convergence, the EXIT chart should be “open.” We conjecture that EXIT chart openness can be determined by testing only the switching points of Gallager’s Algorithm B.

The remainder of this correspondence is organized as follows. In Section II, we introduce our assumptions on channel, codewords, decoding algorithms, and messages. In Section III, we prove the necessity of symmetry and isotropy properties for the optimum BMP decoder. In Section IV, we show that for regular LDPC codes, Gallager’s Algorithm B is optimal among all BMP algorithms. In Section V, we consider the case of irregular codes. We introduce the irregular code design procedure and present some design examples. We also discuss the optimality of Gallager’s decoding Algorithm B for irregular codes, and present analytical results concerning optimum degree distributions in the case of low-error-rate binary-symmetric channels. Finally, in Section VI, we present some conclusions.

II. ASSUMPTIONS AND DEFINITIONS

An LDPC code is usually represented by its factor graph and the decoding algorithm is defined as a set of message update rules at variable nodes and check nodes of this graph. If the outgoing message on an edge is independent of the incoming message on the same edge, the message-passing algorithm can be described in a computation tree [7]. We focus on this class of decoders and define a binary message-passing algorithm as one which uses binary-valued messages. That is to say, the message from the channel as well as the messages which are passed between variable and check nodes are restricted to the set $\{0, 1\}$. A famous example of a BMP algorithm is Gallager’s Algorithm B [1], [7], which will be referred to as Algorithm B throughout this correspondence.

We assume that the messages coming into any vertex of the computation tree are independent. As shown in [7], this assumption becomes true with probability approaching one as the length of the code approaches infinity.

We consider a binary-symmetric channel and assume that all the codewords are equally likely to be chosen at the encoder. Thus, assuming a nondegenerate code, it will be equally likely to have a “1” or a “0” at the channel input.

Given the transmitted codeword, every variable node has a true value. The messages to/from this variable node can either be correct (equal to the true value) or incorrect. We define the message error rate as the probability of a message being incorrect. Given a variable node with the

true value of v , for a message m to/from this variable node, we define $\epsilon = m \oplus v$, where \oplus is the modulo two sum, as the error of m about v . We also say v is the true value of m , by which we mean, in a no-error situation, all the incoming and outgoing messages to a variable node should be equal to the true value of that node. Our goal is to find the *optimum* decoding rule in the sense of minimizing the message error rate at each iteration.

A. Symmetric/Isotropic BMP Decoders

In [7], two symmetry conditions for message-passing decoders are proposed that make the error probability of decoding at each iteration independent of the transmitted codeword over a symmetric channel.

Changing the message alphabet to $\{0, 1\}$ and the arithmetic to logic, the symmetry conditions for binary message decoders can be rewritten as follows.

Check Node Symmetry:

$$\text{CHK}(x_1, \dots, x_{i-1}, \bar{x}_i, x_{i+1}, \dots, x_{d_c-1}) = \overline{\text{CHK}(x_1, \dots, x_{d_c-1})} \quad (1)$$

where CHK is a binary function representing the update rule at the check node, d_c is the check node degree, x_i 's are the binary extrinsic messages to the check node, and \bar{x} is the complement of the binary message x . A function satisfying this symmetry property is a function of the Hamming weight, reduced modulo two, of its arguments.

Variable Node Symmetry:

$$\text{VAR}(\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{d_v-1}) = \overline{\text{VAR}(x_0, x_1, \dots, x_{d_v-1})}$$

where VAR is a binary function representing the update rule at the variable node, d_v is the variable node degree, x_0 is the channel message to the variable node and x_i 's, $i > 0$ are the input messages to the variable node.

As a direct result of [7, Lemma 1], the conditional bit-error rate after the l th decoding iteration is independent of the transmitted codeword if the channel and the update rules are symmetric.

We find the following definitions also useful in our discussion.

Definition 1: Let X and Y be binary random variables. Y is "symmetric" with respect to X if $P(Y = 0|X = 1) = P(Y = 1|X = 0)$.

If Y is symmetric with respect to X , then Y may be considered as the output of a binary-symmetric channel with crossover probability $\delta = P(Y = 1|X = 0)$, where X is the channel input. If Y is symmetric with respect to X we write $X \bowtie Y$.

Lemma 1: $X \bowtie Y$ if and only if $X \oplus Y$ is independent of X .

Proof: Suppose $X \bowtie Y$. Then

$$P(X \oplus Y = 1 | X = 0) = P(Y = 1 | X = 0) = P(Y = 0 | X = 1) = P(X \oplus Y = 1 | X = 1).$$

Conversely, suppose that $X \oplus Y$ is independent of X . Then

$$P(Y = 1 | X = 0) = P(X \oplus Y = 1 | X = 0) = P(X \oplus Y = 1 | X = 1) = P(Y = 0 | X = 1). \quad \square$$

Lemma 2: If $X \bowtie Y$ and $P(X = 1) = P(X = 0) = 1/2$ then $Y \bowtie X$.

Proof: Notice that $P(Y = 1) = P(Y = 0)$ because

$$\begin{aligned} P(Y = 1) &= \sum_{i=0,1} P(Y = 1 | X = i)P(X = i) \\ &= \frac{1}{2} \sum_{i=0,1} P(Y = i | X = 1) = \frac{1}{2}. \end{aligned}$$

As a result

$$P(X = 0 | Y = 1) = P(X = 1 | Y = 0). \quad \square$$

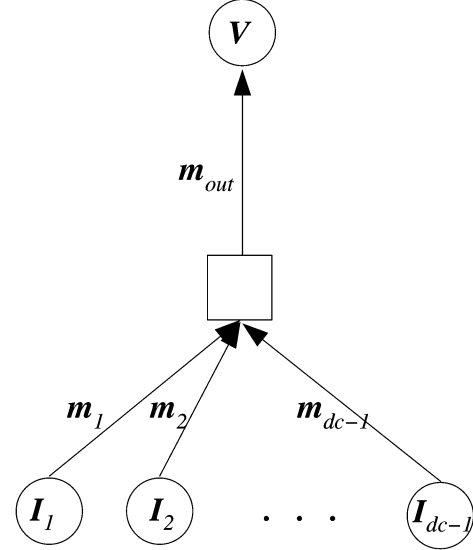


Fig. 1. Messages at a check node.

When $P(X = 0) = P(X = 1) = 1/2$ and $X \bowtie Y$, it can be concluded that $X \oplus Y$ is also independent of Y . This is a direct result of Lemma 2 and Lemma 1.

Definition 2: We call a multivariable function isotropic with respect to two variables x_i and x_j if

$$f(x_0, x_1, \dots, x_i, \dots, x_j, \dots, x_{d-1}) = f(x_0, x_1, \dots, x_j, \dots, x_i, \dots, x_{d-1}).$$

Example 1: The function $f(a, b, c) = ab + bc + ac + c^2$ is isotropic with respect to a and b , but is not isotropic with respect to a and c .

Remark: Isotropic is usually referred to as "symmetry" in mathematics. We use the term isotropic because following [7], [8] the term symmetric is widely used for describing another property of the update rule.

III. NECESSITY OF SYMMETRY AND ISOTROPY CONDITIONS

In this section, we show that among all possible binary update rules, only symmetric, isotropic rules are of interest.

Theorem 1: At a check node, if all the patterns of input messages are equally likely and each message is symmetric with respect to its true value, then the optimum update rule is symmetric.

Proof: Consider the check node of Fig. 1. Assume that the optimum update rule CHK satisfies

$$m_{\text{out}} = \text{CHK}(m_1, \dots, m_{d_c-1})$$

for some specific binary values of m_1 to m_{d_c-1} , where m_i is the input message on the i th input edge. Since, by definition, CHK provides the minimum message error rate, the following inequality holds:

$$P(V = m_{\text{out}} | m_1, \dots, m_{d_c-1}) \geq P(V = \overline{m_{\text{out}}} | m_1, \dots, m_{d_c-1})$$

equivalently

$$P(V = m_{\text{out}} | m_1, \dots, m_{d_c-1}) \geq \frac{1}{2} \quad (2)$$

where V is the true value of the variable node adjacent to the output edge of the check node.

Now consider the i th input message m_i and denote its true value with I_i . By definition, $I_1 \oplus \dots \oplus I_{d_c-1} \oplus V = 0$. Each m_i can be written as

$I_i \oplus \epsilon_i$, where $\epsilon_i = I_i \oplus m_i$ is a binary-valued quantity, representing the error in m_i and (as a result of Lemma 1) independent of m_i . Hence,

$$\begin{aligned} P(V = m_{\text{out}} | m_1, \dots, m_{d_c-1}) \\ = P(m_{\text{out}} = m_1 \oplus \epsilon_1 \oplus \dots \oplus m_{d_c-1} \oplus \epsilon_{d_c-1} | m_1, \dots, m_{d_c-1}). \end{aligned}$$

Similarly

$$\begin{aligned} P(V = \overline{m_{\text{out}}} | m_1, \dots, \overline{m_i}, \dots, m_{d_c-1}) \\ = P(\overline{m_{\text{out}}} = m_1 \oplus \epsilon_1 \oplus \dots \oplus \overline{m_i} \oplus \epsilon_i \oplus \dots \\ \oplus m_{d_c-1} \oplus \epsilon_{d_c-1} | m_1, \dots, \overline{m_i}, \dots, m_{d_c-1}). \end{aligned}$$

Since we assume that all the patterns of input are equally likely and

$$m_{\text{out}} = m_1 \oplus \epsilon_1 \oplus \dots \oplus m_{d_c-1} \oplus \epsilon_{d_c-1}$$

is equivalent to

$$\overline{m_{\text{out}}} = m_1 \oplus \epsilon_1 \oplus \dots \oplus \overline{m_i} \oplus \epsilon_i \oplus \dots \oplus m_{d_c-1} \oplus \epsilon_{d_c-1}$$

it is clear that

$$\begin{aligned} P(V = m_{\text{out}} | m_1, \dots, m_{d_c-1}) \\ = P(V = \overline{m_{\text{out}}} | m_1, \dots, \overline{m_i}, \dots, m_{d_c-1}). \end{aligned} \quad (3)$$

Using (2) and (3) it becomes clear that

$$P(V = \overline{m_{\text{out}}} | m_1, \dots, \overline{m_i}, \dots, m_{d_c-1}) \geq \frac{1}{2}.$$

This proves the necessity of symmetry for CHK. \square

Theorem 2: If the message error rate of each input of a check node is less than half, then the optimum update rule is the modulo-two sum of the input messages.

Proof: Since the symmetry property (1) is satisfied if and only if CHK is a function of the Hamming weight of its arguments modulo two, only two symmetric update rules exist, namely: modulo-two sum of the inputs and its complement.

Consider the check node of Fig. 1. Each input message m_i is equal to $I_i \oplus \epsilon_i$, where ϵ_i is the error of m_i and I_i is its true value. Define ϵ_{\oplus} as the error of m_{out} about its true value V , when the modulo-two sum rule is used. Therefore, $\epsilon_{\oplus} = \epsilon_1 \oplus \dots \oplus \epsilon_{d_c-1}$. Now, define the function $w : \{0, 1\} \mapsto \{-1, 1\}$, $w(x) = (-1)^x$ and let $w_{\oplus} = w(\epsilon_1 \oplus \dots \oplus \epsilon_{d_c-1})$. From the definition of w we have $w_{\oplus} = \prod_{i=1}^{d_c-1} w(\epsilon_i)$. Using the independence of ϵ_1 to ϵ_{d_c-1} we have

$$E(w_{\oplus}) = \prod_{i=1}^{d_c-1} E(w(m_i)) \quad (4)$$

where $E(\cdot)$ represents the expected value. Letting $p_i = P(\epsilon_i = 1)$ and $p_{\oplus} = P(\epsilon_{\oplus} = 1)$, from (4) we have

$$1 - 2p_{\oplus} = \prod_{i=1}^{d_c-1} (1 - 2p_i).$$

Since each $p_i < 1/2$, the right-hand side of this equation is positive which results in $p_{\oplus} < 1/2$. Thus, modulo-two sum has a lower output error rate compared to its complement. \square

It is worth mentioning that modulo-two sum is isotropic with respect to all of its inputs. So the optimum update rule at the check nodes is symmetric and isotropic with respect to all inputs.

Theorem 3: At a variable node V whose input messages are symmetric with respect to V and are independent when conditioned on the value of V , the binary update rule which provides the minimum output message error rate is symmetric.

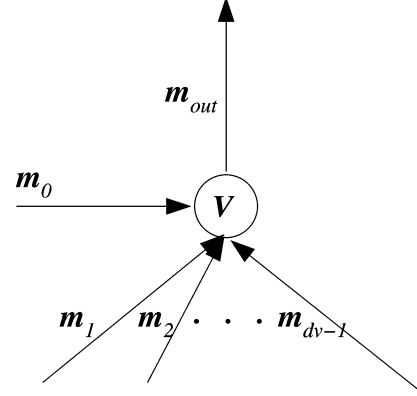


Fig. 2. Messages at a variable node.

Proof: Consider the variable node of Fig. 2. Assume that the optimal update rule VAR satisfies

$$\text{VAR}(m_0, \dots, m_{d_v-1}) = m_{\text{out}}$$

for some specific binary values of m_0 to m_{d_v-1} . Since VAR provides the minimum message error rate, we have

$$P(V = m_{\text{out}} | m_0, \dots, m_{d_v-1}) \geq P(V = \overline{m_{\text{out}}} | m_0, \dots, m_{d_v-1})$$

where V is the true value of the variable node. Under the assumption that each variable node in the code has an equal chance of being a “0” or a “1,” the above inequality can be rewritten as

$$P(m_0, \dots, m_{d_v-1} | V = m_{\text{out}}) \geq P(m_0, \dots, m_{d_v-1} | V = \overline{m_{\text{out}}}).$$

Since the messages are assumed to be independent

$$\prod_{i=0}^{d_v-1} P(m_i | V = m_{\text{out}}) \geq \prod_{i=0}^{d_v-1} P(m_i | V = \overline{m_{\text{out}}}). \quad (5)$$

Using the fact that $V \bowtie m_i$ for all i , (5) can be written as

$$\prod_{i=0}^{d_v-1} P(\overline{m_i} | V = \overline{m_{\text{out}}}) \geq \prod_{i=0}^{d_v-1} P(\overline{m_i} | V = m_{\text{out}})$$

which can be reformatted as

$$P(V = \overline{m_{\text{out}}} | \overline{m_0}, \dots, \overline{m_{d_v-1}}) \geq P(V = m_{\text{out}} | \overline{m_0}, \dots, \overline{m_{d_v-1}})$$

which proves the necessity of symmetry for VAR. \square

Theorem 4: Consider a variable node V whose input messages are symmetric with respect to the value of V and are independent when conditioned on the value of V . If two input messages have equal message error rate, the optimum update rule is isotropic with respect to those inputs.

Proof: Without loss of generality, we prove isotropy of the optimum update rule with respect to m_1 and m_2 , assuming they have equal message error rate. We show that VAR must satisfy

$$\text{VAR}(m_0, 0, 1, m_3, \dots, m_{d_v-1}) = \text{VAR}(m_0, 1, 0, m_3, \dots, m_{d_v-1}).$$

Now, assume that

$$\text{VAR}(m_0, 0, 1, m_3, \dots, m_{d_v-1}) = m_{\text{out}}.$$

Since VAR is the optimum update rule

$$\begin{aligned} P(V = m_{\text{out}} | m_0, 0, 1, m_3, \dots, m_{d_v-1}) \\ \geq P(V = \overline{m_{\text{out}}} | m_0, 0, 1, m_3, \dots, m_{d_v-1}). \end{aligned} \quad (6)$$

Similar to the Proof of Theorem 3, (6) can be written in product form as

$$\begin{aligned} & \prod_{k=0, m_1=0, m_2=1}^{k=d_v-1} P(m_k | V = m_{\text{out}}) \\ & \geq \prod_{k=0, m_1=0, m_2=1}^{k=d_v-1} P(m_k | V = \overline{m_{\text{out}}}). \end{aligned} \quad (7)$$

Now, let $\epsilon_1 = V \oplus m_1$ and $\epsilon_2 = V \oplus m_2$. By assumption, m_1 and m_2 have equal error rate, hence, $P(\epsilon_1 = 1) = P(\epsilon_2 = 1)$. Also notice that ϵ_1 and ϵ_2 are independent of V due to Lemma 1. Thus,

$$\begin{aligned} & P(\epsilon_1 = m_{\text{out}} | V = m_{\text{out}}) P(\epsilon_2 = \overline{m_{\text{out}}} | V = m_{\text{out}}) \\ & = P(\epsilon_1 = \overline{m_{\text{out}}} | V = m_{\text{out}}) P(\epsilon_2 = m_{\text{out}} | V = m_{\text{out}}) \end{aligned}$$

or

$$\begin{aligned} & P(m_1 = 0 | V = m_{\text{out}}) P(m_2 = 1 | V = m_{\text{out}}) \\ & = P(m_1 = 1 | V = m_{\text{out}}) P(m_2 = 0 | V = m_{\text{out}}). \end{aligned} \quad (8)$$

Using (8), we may rewrite (7) as

$$\begin{aligned} & \prod_{k=0, m_1=1, m_2=0}^{k=d_v-1} P(m_k | V = m_{\text{out}}) \\ & \geq \prod_{k=0, m_1=1, m_2=0}^{k=d_v-1} P(m_k | V = \overline{m_{\text{out}}}), \end{aligned}$$

which is equivalent to

$$\begin{aligned} & P(V = m_{\text{out}} | m_0, 1, 0, m_3, \dots, m_{d_v-1}) \\ & \geq P(V = \overline{m_{\text{out}}} | m_0, 1, 0, m_3, \dots, m_{d_v-1}). \quad \square \end{aligned}$$

Notice that if all the extrinsic messages to a variable node have equal message error rate, under symmetry and independence assumption on these messages, the optimum variable node update rule must be isotropic with respect to these inputs. The channel message, however, usually has a different message error rate and the optimum update rule may treat it differently than the extrinsic messages.

So far, we have shown that, under some assumptions for the inputs to the check/variable nodes, the optimum update rule has to be symmetric and isotropic. Notice that at the first iteration (channel messages sent by the variable nodes to the check nodes) these assumptions are fulfilled by the channel messages, i.e., all patterns of input messages at the check nodes are equally likely and each message is symmetric with respect to its true value. Therefore, necessity of symmetric/isotropic decoding at the first iteration at the check nodes is proved. We now show that the assumptions made at the check/variable nodes are fulfilled and preserved by symmetric decoding. This shows the necessity of symmetric/isotropic decoding for all iterations of the decoding.

Theorem 5: If the input messages to a check node are symmetric with respect to their true value, using a symmetric update rule the output message is also symmetric with respect to its true value. Moreover, if the input messages are equally “0” or “1,” so is the output message.

Proof: There are two symmetric check node update rules. We finish the proof for modulo-two sum, which is of our interest. The proof for its complement follows the same lines and is omitted here.

Consider the check node of Fig. 1. By definition

$$V = I_1 \oplus I_2 \cdots \oplus I_{d_c-1}.$$

Let $\epsilon_i = m_i \oplus I_i$ and notice that $I_i \bowtie m_i$ and $P(m_i = 0) = P(m_i = 1) = 1/2$. Hence, using Lemma 2, $m_i \bowtie I_i$, which in turn shows

that I_i and ϵ_i are independent. Now, assuming a decoding tree, i.e., a cycle-free local neighborhood, all ϵ_i 's and I_i 's are mutually independent. As a result, $\epsilon_1 \oplus \cdots \oplus \epsilon_{d_c-1}$ is independent of $I_1 \oplus \cdots \oplus I_{d_c-1}$. In other words, $V \oplus m_{\text{out}}$ is independent of V , which means that m_{out} is symmetric with respect to V (Lemma 1).

Since the check node update rule is symmetric, of the 2^n possible patterns at the input, half give rise to an output of “0” and the other half to an output of “1.” Since all the input patterns are equally probable, the output is equally likely to be “0” or “1.” \square

This theorem shows that after message-update at the check nodes at the first iteration, the messages are still symmetric with respect to their true values and equally “0” or “1,” hence, the optimum variable node update rule at this iteration has to be symmetric. The following theorem shows that the symmetry of messages with respect to their true value is preserved by symmetric update rule at the variable nodes as well.

Theorem 6: If the input messages to a variable node V are symmetric with respect to their true value V , using a symmetric update rule the output message is also symmetric with respect to V . Moreover, if the input messages are equally “0” or “1,” so is the output message.

Proof: At a variable node whose true value is V and whose output message is m_{out} , we show that $P(m_{\text{out}} = 0 | V = 1)$ and $P(m_{\text{out}} = 1 | V = 0)$ are equal. To see this, consider two cases. In the first case, the true value of V is “0.” Now assume a pattern of input messages m_0, \dots, m_{d_v-1} at the input of V . In the second case, assume that the true value of V is “1.” Since $P(m_i = 0 | V = 0) = P(m_i = 1 | V = 1)$, the complement of the above pattern, i.e., $\overline{m_0}, \dots, \overline{m_{d_v-1}}$, happens at the input of the variable node with the same probability as the pattern m_0, \dots, m_{d_v-1} in the first case. Due to the symmetry of variable node update rule, the output messages in these two cases are complements of each other. In other words

$$P(m_{\text{out}} = 0 | V = 1) = P(m_{\text{out}} = 1 | V = 0).$$

Since the variable node update rule is symmetric, of the 2^n possible patterns at the input, half give rise to an output of “0” and the other half to an output of “1,” so the output is equally “0” or “1.” \square

As a result of this theorem, the messages to the next round of decoding have the conditions assumed in Theorems 1 and 3. Thus, the necessity of symmetric decoding at all iterations is proved.

As a direct result of symmetric decoding, at all iterations the input messages to a variable node are symmetric with respect to their true values. Therefore, using Theorem 4, the necessity of isotropic decoding is also proved.

IV. OPTIMALITY OF ALGORITHM B FOR REGULAR CODES

In this section, we show that for decoding regular codes, Algorithm B has the optimum threshold among all symmetric/isotropic BMP algorithms. Since we have already proved necessity of symmetry and isotropy properties, this proves that Algorithm B is the optimum BMP decoding rule.

We have already shown that at the check nodes, modulo-two sum is the optimum update rule. We now focus on the update rule at the variable nodes. We let p_0 be the probability of a channel message being in error and p be the probability of a check-to-variable message being in error. Since the code is assumed to be regular and the update rules are symmetric and isotropic, p is the same for all the messages.

At a variable node of degree d_v , we receive a message from the channel and $d_v - 1$ messages from the neighboring variable nodes. Assuming that the channel message is m_0 and that b of the extrinsic

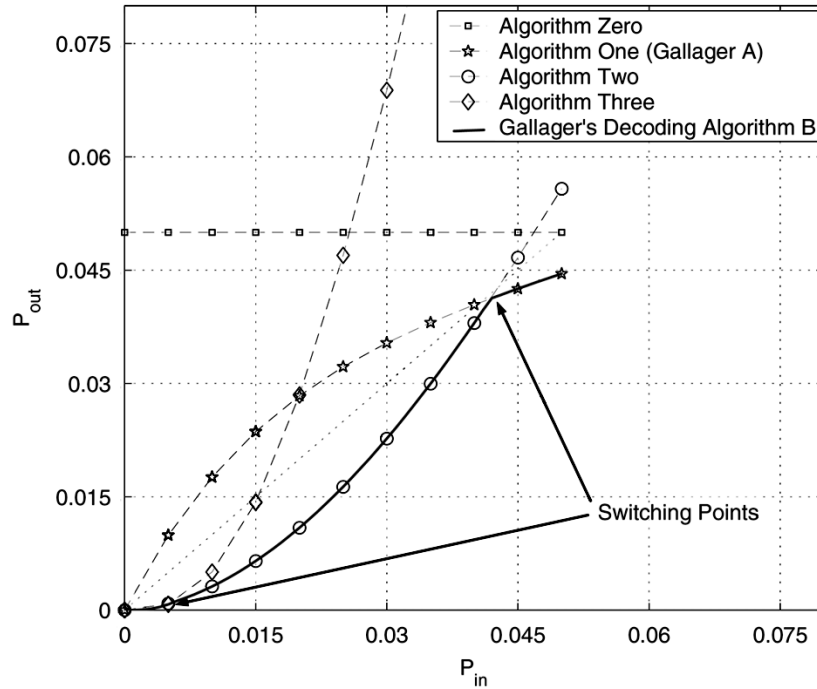


Fig. 3. EXIT charts for different decoding algorithms in the case of a regular (6, 10) code.

messages are 1's and $d_v - 1 - b$ are 0's, or in other words, the weight w of the extrinsic messages is b , it can be concluded that

$$\begin{aligned} \frac{P(V=0|m_0, w=b)}{P(V=1|m_0, w=b)} &= \frac{P(m_0, w=b|V=0)}{P(m_0, w=b|V=1)} \\ &= \frac{P(m_0|V=0)p^b(1-p)^{d_v-1-b}}{P(m_0|V=1)(1-p)^b p^{d_v-1-b}}. \end{aligned} \quad (9)$$

Using the symmetry of the channel, (9) simplifies to

$$\frac{P(V=0|m_0, w=b)}{P(V=1|m_0, w=b)} = \frac{P(m_0|V=0)}{1 - P(m_0|V=0)} \left(\frac{1-p}{p} \right)^{d_v-1-2b}.$$

Hence, the optimum decision for a given b, p, p_0 , and m_0 is obvious. It is easy to verify that this decision rule is equivalent to Algorithm B.

V. IRREGULAR CODES

Algorithm B can be viewed as switching between a set of elementary algorithms. Every elementary algorithm has a fixed b and the outgoing message at a variable node is the same as the intrinsic message, if it has at least $d_v - 1 - b$ extrinsic messages in agreement with the intrinsic message. A graphical view of this fact can be very useful. Fig. 3 shows the p_{in} versus p_{out} EXIT chart for these decoding algorithms in the case of a regular (6, 10) code when the channel crossover probability is 0.05. In this figure, Algorithm i ($0 \leq i \leq \lfloor \frac{d_v}{2} \rfloor$) is the algorithm which requires i agreements from the extrinsic messages to set its outgoing message equal to the intrinsic message. It is clear from Fig. 3 that Algorithm Zero is not used in any iteration. It can be understood here that, in the case of regular codes, decoding Algorithm Zero cannot be of any use. The reason is that a successful decoding will be made possible if for every p_{in} , $0 < p_{in} \leq p_0$, we have $p_{out} < p_{in}$. However, for Algorithm Zero, $p_{out} = p_0$.

For an irregular code, we can perform Algorithm B for each variable node depending on its degree. However, we note that the proof of optimality of Algorithm B for regular codes requires that the message error rate of the inputs to the variable nodes be equal. For irregular codes,

however, if variable nodes have a knowledge of their depth- l decoding tree, they can distinguish between more and less reliable messages and follow an update rule which may be more effective than an isotropic one. However, if such a knowledge does not exist or is not used, the input message error rates are equal and optimality of Algorithm B is preserved.

An irregular LDPC code is usually described by the fraction of edges incident on variable nodes and check nodes of different degrees. For example, the irregularity in the variable nodes can be specified by the so-called variable degree distributions $\{\lambda_2, \lambda_3, \dots\}$, where λ_i indicates the fraction of edges connected to variable nodes of degree i . Similarly, we can describe the irregularity in the check nodes, using a check degree distributions $\{\rho_2, \rho_3, \dots\}$.

As described in [11], when the EXIT charts are based on error probability of messages and the check degree distribution is fixed, the EXIT chart of the irregular code is a linear combination of the EXIT charts corresponding to different variable degrees. We will refer to the EXIT chart corresponding to any fixed variable degree i by g_i . The EXIT chart of the irregular code is a linear combination of g_i 's and the weights of this linear combination are determined by the variable degree distribution.

For a given p_0 , one design problem is to find the highest rate code whose threshold of decoding is greater than or equal to p_0 . That is, we wish to find a linear combination with an open EXIT chart which maximizes the code rate. The design rate of the code is

$$R = 1 - \frac{\sum_j \rho_j / j}{\sum_i \lambda_i / i} \quad (10)$$

where ρ 's and λ 's are the check and the variable edge degree distribution, respectively.

Fig. 4 shows the EXIT chart of different variable degrees for channel crossover probability of $p_0 = 0.05$, assuming (in the dashed curves) that each of the variable nodes uses Algorithm B without Algorithm Zero (as is the case for regular codes), and assuming (in the solid curves) that Algorithm Zero is also used. One can see that the message error rate at the output of variable nodes with degree two and three,

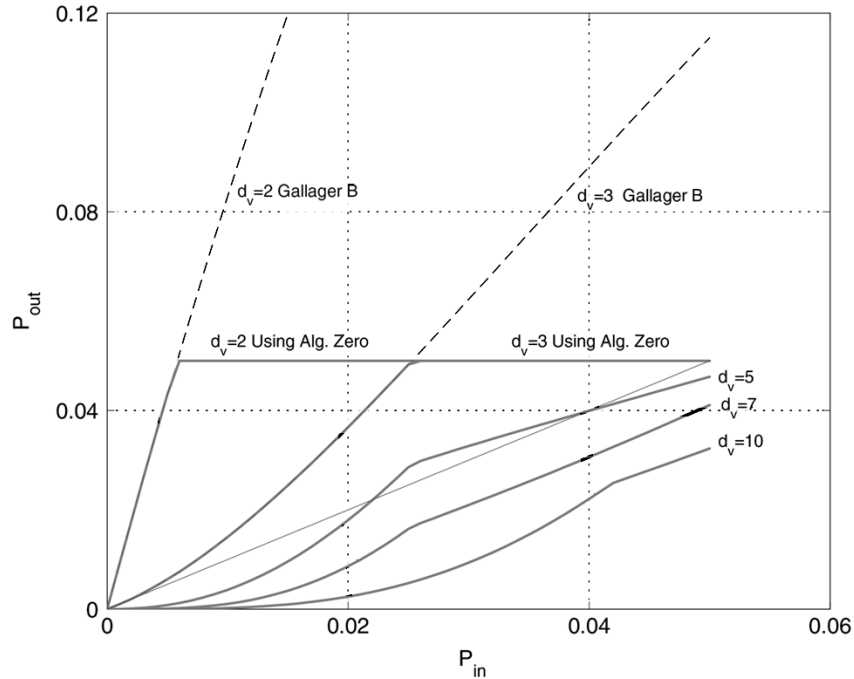


Fig. 4. EXIT charts for various d_v , fixing $d_c = 10$.

when Algorithm Zero is not used, are initially greater than p_0 . This suggests use of decoding Algorithm Zero for irregular codes.

In the design of the codes, openness of every point of the EXIT chart must, in principle, be tested. We have observed that testing openness only at the switching points of the algorithms used for different variable degree seems to result in an EXIT chart that is everywhere open. However, we have not yet proved that this is a sufficient condition for openness of the EXIT chart. Note that after finishing the design it is very easy to verify whether the EXIT chart is open or not because we are dealing with polynomial functions. This seems to be a more effective method compared to that of [12], which is based on a discrete version of P_{in} and needs a larger number of points to be examined yet results in some *conflict intervals* [12]. The shape of EXIT charts makes it intuitive that the switching point of Algorithm B should be critical points in the EXIT chart of the irregular designed code. Considering a discrete version of P_{in} without noticing this fact can result in a failure in design. As stated in [12], the design based on course discretization results in intervals in which the solution does not satisfy the inequalities of the linear program. The authors of [12] also mention that a fine discretization is computationally ineffective, so the threshold of their designed code p_0^* is slightly less than the crossover probability p_0 of the channel for which the code was designed. In our case, however, we have $p_0^* = p_0$ by solving the linear program only for a few points.

Table I shows the results of maximum rate code design for some given values of crossover probability p_0 . In the design, we have limited the variable node degree to a maximum of 50. The codes are regular at the check side and the value of d_c is chosen to allow the maximum code rate.

A. Stability Condition

It is useful to derive a stability condition similar to that of [8] for decoding Algorithm B. Near $p = 0$, all the variable nodes use Algorithm $\lfloor d_v/2 \rfloor$, so their EXIT charts approach zero with slope zero except for $d_v = 2$ and $d_v = 3$. It is easy to see that the slope for $d_v = 2$ is $\bar{d}_c - 1$ and for $d_v = 3$ is $2p_0(\bar{d}_c - 1)$, where \bar{d}_c is the average check node

TABLE I

p_0	variable degree distribution	d_c	rate	capacity
0.07	$\lambda_2=0.0390, \lambda_3=0.2158,$ $\lambda_{12}=0.4450, \lambda_{49}=0.3002$	14	0.4694	0.6341
0.05	$\lambda_2=0.0229, \lambda_3=0.1483,$ $\lambda_4=0.0750, \lambda_{10}=0.0914,$ $\lambda_{12}=0.2695, \lambda_{15}=0.0287,$ $\lambda_{49}=0.3642$	20	0.5853	0.7136

degree. Thus, in order to have a slope of less than one for the irregular code near zero, the degree sequence should satisfy

$$\frac{1}{\lambda_2 + 2p_0\lambda_3} > \bar{d}_c - 1. \tag{11}$$

For a fixed \bar{d}_c , this inequality puts an upper bound on the rate of an irregular code. As discussed in [11], for a fixed check degree distribution, in order to achieve the maximum rate code we wish to maximize $\sum_{i \geq 2} \frac{\lambda_i}{i}$. The maximum is achieved if we put as much weight as possible on the lower degrees. There is a limit on the maximum of λ_2 and λ_3 because of (11). Assuming we have λ_2 and λ_3 such that

$$\frac{1}{\lambda_2 + 2p_0\lambda_3} = \bar{d}_c - 1$$

and assuming that the rest of the total weight, i.e., $1 - \lambda_2 - \lambda_3$, can be placed on λ_4 we can compute an upper bound on the maximum rate achievable with Algorithm B for a given \bar{d}_c as

$$R \leq 1 - \frac{24p_0}{6p_0d_c + \lambda_2(12p_0 - d_c) + \frac{\bar{d}_c}{d_c - 1}}. \tag{12}$$

B. Low-Error-Rate Channels

The bound of (12) becomes more accurate when p_0d_c is small, allowing a code with variable degrees 2, 3, and 4 to converge. When $p_0d_c \ll 1$ and $d_c \gg 1$, the bound can be approximated as

$$R \leq 1 - \frac{24p_0}{1 - \lambda_2d_c}$$

suggesting that smaller λ_2 results in higher rates. In our presented results, also λ_2 tends to be small. Letting $\lambda_2 = 0$, we have $R \leq 1 - 24p_0$. Also notice that when $p_0 d_c \ll 1$ and $\lambda_2 = 0$, then $\lambda_3 = 1$ satisfies (11), hence, the code rate will be $R = 1 - \frac{3}{d_c}$. This code rate is achieved if the EXIT chart for a degree three variable node, given an average check node of \bar{d}_c , is open.

These observations give rise to an interesting practical question for code design over binary-symmetric channels (BSC) when the crossover probability of the channel is very small and the variable degree is fixed to three.

Now, consider a check degree distribution $\{\rho_i, 2 \leq i \leq i_{\max}\}$, $\sum_i \rho_i = 1$ and define $\rho(x) = \sum_i \rho_i x^{i-1}$. It is easy to see that the error rate of messages at the output of check nodes is

$$p_c(p) = \frac{1 - \rho(1 - 2p)}{2}$$

where $0 \leq p \leq p_0$ is the input message error rate to the check nodes. The output error rate of a degree three variable node whose channel message has an error rate of p_0 and whose input messages have an error rate of p_c is $p_c(p)^2 + 2p_0(1 - p_c(p))p_c(p)$.

Since $p_c(p)$ is a concave function of p , an upper bound for it is the tangent line at the origin, i.e.,

$$p_c(p) \leq p_c^{(b)}(p) = p \left(\sum_i i \rho_i - 1 \right). \quad (13)$$

A sufficient (but not necessary) condition for convergence is then

$$\forall p \in (0, p_0], \quad \left(p_c^{(b)}(p) \right)^2 + 2p_0 \left(1 - p_c^{(b)}(p) \right) p_c^{(b)}(p) < p. \quad (14)$$

The left-hand side of (14) is an increasing convex function of p and, hence, the inequality is satisfied if and only if it is satisfied at $p = p_0$. After some manipulation and simplification, we obtain the following sufficient condition for convergence. We must have

$$\sum_i i \rho_i \leq \frac{1}{\sqrt{p_0}} \frac{\sqrt{1 - p_0} - 2p_0 \sqrt{p_0}}{1 - 2p_0} \quad (15)$$

which for small p_0 simplifies to

$$\sum_i i \rho_i \leq \frac{1}{\sqrt{p_0}}.$$

Notice that (15) is a sufficient condition and not a necessary condition for convergence. Hence, the optimum (in the sense of maximizing the code rate) check degree distribution which satisfies (15) is not necessarily the optimum check degree distribution for which the convergence occurs. However, as p approaches zero, the inequality of (13) becomes equality, and hence, (15) will be a necessary and sufficient condition. Therefore, for very small p_0 we expect a very close agreement between the two optimum check degree distributions.

To have the maximum code rate for a fixed variable degree distribution, according to (10), we wish to minimize $\sum_i \rho_i / i$. This quantity will be minimized if we put more weight on higher degrees. Thus, if i_{\max} satisfies (15), then all the weight should be placed on i_{\max} . Otherwise, the following theorems show that the optimum check degree distribution is concentrated on one or two degrees.

Theorem 7: Given an integer N , the minimum of $\sum_i \rho_i / i$ subject to $\rho_i > 0$, $\sum_i \rho_i = 1$ and $\sum_i i \rho_i \leq N$ is achieved when $\rho_N = 1$.

Proof: Assume that another degree distribution has achieved the minimum. Call $I = \{i, \rho_i > 0\}$ the support of this degree distribution.

It is clear that all the members of I are not greater than N or $\sum_i i \rho_i > N$. If all the members of I are less than N , it is clear that $\sum_i \rho_i / i > 1/N$ which contradicts the assumption that this set of weights achieves the minimum, because $\rho_N = 1$ achieves $1/N$.

As a result, we have to assume that there are elements greater than N and also elements less than N present in I . Consider two of these elements, $N - k$ and $N + m$, where k and m are positive integers. Change the degree distribution as follows. Set $\rho'_i = \rho_i$ if $i \notin \{N - k, N, N + m\}$, $\rho'_{N-k} = \rho_{N-k} - \epsilon m$, $\rho'_{N+m} = \rho_{N+m} - \epsilon k$, and $\rho'_N = \rho_N + \epsilon(m + k)$, where $\epsilon > 0$ is chosen to avoid negative ρ' weights.

It is straightforward to see that $\sum_i \rho'_i = 1$, $\sum_i i \rho'_i \leq N$, and $\sum_i \rho'_i / i < \sum_i \rho_i / i$, which contradicts the assumption that the set of ρ 's achieves the minimum. So the minimum is achieved by $\rho_N = 1$ and is equal to $1/N$ \square

In this proof, we devised a method for trading weights larger than N with weights less than N , allowing a higher weight on N and a higher rate code. This method can be extended to prove the following theorem.

Theorem 8: Given a noninteger number R , the minimum of $\sum_i \rho_i / i$ subject to $\rho_i > 0$, $\sum_i \rho_i = 1$, and $\sum_i i \rho_i \leq R$ is achieved when $\rho_{\lceil R \rceil} = R - \lfloor R \rfloor$, $\rho_{\lfloor R \rfloor} = \lceil R \rceil - R$, and $\rho_i = 0$ whenever $i \notin \{\lfloor R \rfloor, \lceil R \rceil\}$.

Proof: Following the lines of the proof of the previous theorem, it is evident that by considering any set of weight satisfying the conditions, all the weights below $\lfloor R \rfloor$ can be traded with weights greater than $\lfloor R \rfloor$ in order to allow a higher rate code. Repeating a similar proof, one can argue that the weights more than $\lceil R \rceil$ should be traded for some of the weight on $\lceil R \rceil$ to allow for a higher weight on $\lceil R \rceil$ and hence a higher rate code. Notice that our proof in the previous theorem did not use the fact that the bound on $\sum_i i \rho_i$ is an integer number. Hence, the same lines of proof are valid here.

As a result, the whole weight is on $\lfloor R \rfloor$ and $\lceil R \rceil$. We wish as large as possible $\rho_{\lceil R \rceil}$, which can be computed as $\rho_{\lceil R \rceil} = R - \lfloor R \rfloor$. Hence, $\rho_{\lfloor R \rfloor} = \lceil R \rceil - R$. \square

Using these two theorem for BSCs with low crossover probability, one can design an irregular code with $d_v = 3$. For instance, when $p_0 = 0.001$, the suggested solution is $\lambda_3 = 1$, $\rho_{31} = 0.3773$, and $\rho_{32} = 0.6227$ and the rate of this code is 0.9051. The maximum-rate code which can be designed with a maximum node degree of 32 has a rate of 0.9076.

VI. CONCLUSION

In this correspondence, we have shown that the optimum binary message-passing decoder, when the factor graph of the code is a tree, has certain symmetry/isotropy attributes. We also have shown that in the case of regular codes, Algorithm B is the optimum binary message-passing algorithm in the sense of minimizing message error rate. In the case of irregular codes, when variable nodes do not exploit structural knowledge of their local decoding neighborhood, Algorithm B remains optimum.

We propose a design method for irregular codes with the maximum possible rate by solving a linear program only for the switching points of Algorithm B. An interesting observation, easily verified analytically, is that the switching points for a variable node of degree d_v are a subset of the switching points for a variable node of degree $d_v + 2$. This means that the switching points can be determined by considering only the maximum degree variable node and one degree less. In practice, most of these points are greater than p_0 and need not be considered.

We have also derived a stability condition and an upper bound on the code rate for Algorithm B. We also studied the case of low-error-rate channels and proved that in certain situations, the check degree

distribution of the maximum rate code tends to be concentrated on one or two degrees.

REFERENCES

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
 [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.
 [3] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
 [4] A. Shokrollahi, "New sequence of linear time erasure codes approaching the channel capacity," in *Proc. Int. Symp. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1999, vol. 1719, pp. 65–67.
 [5] A. Shokrollahi, "Capacity-achieving sequences," *IMA Volumes in Mathematics and Its Applications*, vol. 123, pp. 153–166, 2000.
 [6] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
 [7] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
 [8] T. J. Richardson, A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
 [9] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
 [10] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
 [11] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.
 [12] M. G. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

A Family of Highly Symmetric Codes

Jürgen Bierbrauer, Stefano Marcugini, and Fernanda Pambianco

Abstract—We define a class of codes admitting a large automorphism group. This family contains the binary extended Hamming code, the hexacode, the Golay codes, the Pless symmetry codes, as well as the $[16, 4, 12]_8$ -codes constructed by Marcugini *et al.* A computer search resulted in the construction of codes with new parameters $[28, 7, 18]_8$, $[32, 8, 20]_8$, and $[39, 13, 17]_4$ belonging to this family.

Index Terms—Linear codes, Galois fields, symmetry groups.

Manuscript received February 16, 2004; revised March 27, 2005. This work was supported by Italian MIUR, CNR, and GNSAGA.

J. Bierbrauer is with the Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931 USA (e-mail: jbiebra@mtu.edu).

S. Marcugini and F. Pambianco are with the Department of Mathematics and Informatics, Perugia University, 06123 Perugia, Italy (e-mail: gino@dipmat.unipg.it; fernanda@dipmat.unipg.it).

Communicated by C. Carlet, Associate Editor for Coding Theory. Digital Object Identifier 10.1109/TIT.2005.855607

I. INTRODUCTION

We define a class of linear codes in terms of the action of a group of symmetries. The group of symmetries is a direct product $G \times \langle \phi \rangle$, where G is a regular subgroup of the symmetric group S_k and ϕ is the Frobenius automorphism of the field extension $\mathbb{F}_{q^r} | \mathbb{F}_q$. The parameters of the codes in question are $[(r + 1)k, k, d]_{q^r}$. The group $G \times \langle \phi \rangle$ has two orbits, of lengths k and rk , respectively, on the coordinates of the code. In Section II, we give a precise definition of the action of the group and prove that some of the most famous codes, the Golay codes, the hexacode, the binary extended Hamming code, and the Pless symmetry codes, belong to the family. For the sake of illustration we use one of the $[16, 4, 12]_8$ near maximum-distance separable (NMDS) codes constructed in [5]. Section III reports the results of a computer search for codes with new parameters belonging to the family. The new code parameters are $[32, 8, 20]_8$, $[28, 7, 18]_8$, and $[39, 13, 17]_4$. The full automorphism groups of the new codes are described in Section IV. The presence of a large automorphism group tends to give a better understanding of the code. We demonstrate this in Section V by giving a synthetic construction of the more symmetric of the $[16, 4, 12]_8$ -codes.

II. A FAMILY OF CODES

Let a projective $[n, k, d]_q$ -code \mathcal{C} be described by a generator matrix \mathbf{A} . The set of n points in projective $(k - 1)$ -dimensional geometry $\text{PG}(k - 1, q)$ generated by the columns of \mathbf{A} describes the code geometrically. Define the automorphism group $\text{Aut}(\mathcal{C})$ as the stabilizer of the point set under the action of $\text{P}\Gamma\text{L}(k, q)$. Matrices from $\text{GL}(k, q)$ act from the left on column vectors. Denote conjugation by $g^h = hgh^{-1}$. Consequently, the image of matrix \mathbf{M} under a field automorphism ϕ is denoted by \mathbf{M}^ϕ . The reason is that \mathbf{M}^ϕ acts like the conjugate of \mathbf{M} under ϕ . Identify the permutation $\pi \in S_k$ with the permutation matrix, which has entry 1 in row i and column $\pi(i)$, for all i .

We describe a family of linear codes, as follows. Consider the field extension $\mathbb{F}_{q^r} | \mathbb{F}_q$. Let the Frobenius automorphism be defined by $\phi(x) = x^q$. It generates the Galois group $\text{Gal}(\mathbb{F}_{q^r} | \mathbb{F}_q) \cong \mathbb{Z}_r$. Let $G \subset S_k$ be a regular permutation group on k symbols and P a point in $\text{PG}(k - 1, q^r)$ generated by $s \in \mathbb{F}_{q^r}^k$ and \mathcal{O} the orbit of P under the action of the direct product $G \times \langle \phi \rangle$, where G permutes the indices and ϕ acts coordinatewise. Here, \mathcal{O} should be chosen different from the orbit of length k consisting of the elementary (weight 1) vectors. Let \mathbf{A} be the $(k, k + |\mathcal{O}|)$ -matrix whose first k columns form the identity matrix, whose remaining columns are representatives of the elements of \mathcal{O} .

Let $\mathcal{C} = \mathcal{C}(q, r, G, P)$ be the code with \mathbf{A} as generator matrix. Clearly, \mathcal{C} is a q^r -ary code of dimension k and length $k + |\mathcal{O}|$. The problem is to find $P \in \text{PG}(k - 1, q^r)$ such that the minimum distance d is large. Observe that the first k columns of \mathbf{A} (the identity matrix) form an orbit under the action of $G \times \langle \phi \rangle$. It follows that $G \times \langle \phi \rangle$ is a subgroup of the automorphism group of \mathcal{C} .

As an illustration, we use the $[16, 4, 12]_8$ NMDS codes constructed in [5]. We represent the field \mathbb{F}_8 in the form $\mathbb{F}_8 = \mathbb{F}_2(\epsilon)$, where $\epsilon^3 = \epsilon^2 + 1$. It was shown in [5] that there are exactly two nonequivalent $[16, 4, 12]_8$ -codes. The first of those codes, call it \mathcal{A}_1 , can be described as $\mathcal{C}(2, 3, E_4, P_1)$, where E_4 is the Klein group in its natural action on four points and $P_1 = (1 : \epsilon^3 : \epsilon^2 : \epsilon)$. The second $[16, 4, 12]_8$ -code \mathcal{A}_2 is $\mathcal{C}(2, 3, \mathbb{Z}_4, P_2)$, where \mathbb{Z}_4 is the cyclic group of order 4 and $P_2 = (1 : \epsilon : \epsilon^3 : \epsilon^2)$. Let

$$\mathbf{C}(\mathbf{a}) = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{pmatrix}$$

where $\mathbf{a} \in \mathbb{F}_8^4$.