Codes for Symbol-Pair Read Channels

Yuval Cassuto
Hitachi Global Storage Technologies
San Jose Research Center
3403 Yerba Buena Rd.
San Jose, CA 95135, U.S.A.
yuval.cassuto@hitachigst.com

Mario Blaum
Grupo de Análisis, Seguridad y Sistemas (GASS)
Facultad de Informática, Despacho 431
Universidad Complutense de Madrid (UCM)
C/ Profesor José García Santesmases s/n, 28040 Madrid, Spain
mario.blaum@fdi.ucm.es

Abstract— **A new coding framework is established for channels whose outputs are overlapping pairs of symbols. Such channels are motivated by storage applications in which the spatial resolution of the reader may be lower than that of the process that was used to store the data. Reading symbols as pairs changes the error model from the standard bounded number of symbol errors to a bounded number of pair errors. Starting from the most basic coding-theoretic questions, the paper studies codes that protect against pair-errors. It provides answers on pair-error correctability, code construction and decoding, and lower and upper bounds on code sizes.**

I. Introduction

The central theme of information theory is to manipulate and reason about information when it is containerized in quanta called symbols. The basic information unit is usually fixed at the outset, and the behavior of these units is examined through channels, processing units, and other liabilities, which are discretized correspondingly. In particular, the theory of error-control codes aims at recovering the original information units when some bound is given on their corruption. These corruption bounds can be defined at the code-block level, like a certain number of errors in Hamming-metric codes, or at the individual-symbol level, like symbol-transition restrictions in asymmetric or uni-directional error-correcting codes. The alphabet on which the information unit is defined may change throughout the coding problem, like in soft-decoding, but still it is typically the same unit that is tracked and analyzed.

There are cases where it is incumbent upon the code designer to depart from the coupling of information units with channel uses. In such cases, physical constraints may enforce representing information as one unit, while the channel or impairment may operate on a different unit. This paper's subject is one such instance that is motivated by the application of high-density data storage technologies. In essence, the model treated here is of information units defined over some discrete symbol alphabet, but whose reading from the channel is performed as (potentially corrupted) overlapping *pairs* of symbols. So while the codes are defined as usual over some q-ary symbol alphabet, their design objective is to protect against a certain number of *pair* errors, rather than a certain number of symbol errors. A pair-error is defined as a pair-read in which one or more of the symbols is read in error. The main motivation for the pair-error model is to address scenarios where, due to physical limitations, individual symbols cannot be read off the channel, and therefore, each channel read contains contributions from two adjacent symbols. Such a scenario can be manifested, for example, when information is written to the surface of storage media by a high-resolution write process (e.g. lithography), and later read by a lower-resolution read process (e.g. a magnetic read head). Such a case is described pictorially in Figure 1. Note that even though the reader is not able to spatially isolate two adjacent symbols, it can still provide hypotheses on the symbols themselves (and not a mixed function thereof), assuming the two read symbols have orthogonality properties with respect to the read process[1]. However, since two hypotheses are obtained in a single channel use, a single read error event affects either one or both of the symbol reads in the pair. The overlap between adjacent pair reads suggested by the model is advantageous over the standard method of partitioning the symbols to disjoint pairs, since it provides two observations of each symbol, for the same traversal of the reader over the stored symbols. Even though the main motivation comes from data storage, we attempt to maintain an application-agnostic discussion, and use the storage terms written/read interchangeably with the standard coding theory terms of transmitted/received, respectively.

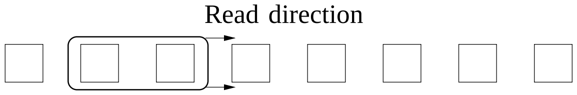


Figure 1. Read process performed in pairs due to resolution deficiency of reader.

The proposed model is related to the model of multiple-burst errors, since it treats a pair-error as the error unit, whether one or both of the symbols were received in error. Nevertheless, the model of burst errors still has the property that the transmitted and received units are identical, unlike the pair-error model. While of no direct applicability to the pair-error model, some relevant results from the theory of multiple-burst error correction can be found in [4] and [1].

The initial treatment of pair-errors in this paper pursues some questions that are well known and/or well studied for traditional error-control coding. At times the results for the pair-metric are similar to known ones for the Hamming metric, but at other times the behavior of the new metric turns out more surprising and counter-intuitive. In section II, a relevant distance metric is defined, and is used to provide necessary

[1] A simple example may be symbols that are taken from some orthogonal set of sequences.

and sufficient conditions for pair-error correctability. In section III, the more "constructive" issues of code construction and decoding are discussed, where known art from standard coding theory is found useful if properly adapted to address the new model. Section IV provides upper and lower bounds on code sizes by combinatorially enumerating spheres in the pair-metric. Finally, in section V a graph-theoretic view of the pair-error model is presented. This view is used for additional code constructions and treatment of the case where the assignment of symbols into pairs is restricted.

II. Symbol-Pair Read Channels

The standard regime treated by coding theory is where decoder inputs are corrupted versions of symbols output by the encoder. So a typical coding problem defines some alphabet Ξ, a block size n, and specifies the code $\mathcal{C} \subset \Xi^n$, and the decoder $dec : \Xi^n \to \mathcal{C}$. This is obviously a very useful framework to combat noisy memoryless channels that arise in a broad variety of communication and storage applications. In some specific applications, however, this coding-problem definition does not adequately describe the constraints of the physical system, in which case a refinement is needed to get better optimized coding solutions. In the current paper, the subject of study are codes over an alphabet Ξ, with block size n, whose decoder inputs are n (potentially corrupted) *pairs* of adjacent code symbols from Ξ. More precisely, the code is defined, as usual, as a subset $\mathcal{C} \subset \Xi^n$, but the decoder is now a function $pair\_dec : (\Xi, \Xi)^n \to \mathcal{C}$. So the inputs to the decoder are n *pairs* of symbols, where each symbol is independently read in two adjacent pairs. To distinguish pair vectors from standard, symbol vectors, we will over-mark them with the symbol $\leftrightarrow$.

$$\overset{\leftrightarrow}{u} = [(\lhd u_0, \rhd u_0), \ldots, (\lhd u_{n-1}, \rhd u_{n-1})]$$

In each pair, $\lhd$ and $\rhd$ designate the left and right symbols, respectively. Each symbol vector in Ξ^n can be represented as a symbol-pair vector as defined below.

Definition 1.*(Symbol-Pair Read Vector)*
Let $\boldsymbol{x} = [x_0, \ldots, x_{n-1}]$ be a vector in Ξ^n. The symbol-pair read vector of $\boldsymbol{x}$ is defined as

$$\pi(\boldsymbol{x}) = [(x_0, x_1), (x_1, x_2), \ldots, (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)].$$

Every vector $\boldsymbol{x} \in \Xi^n$ has a pair representation $\pi(\boldsymbol{x}) \in (\Xi, \Xi)^n$. However, not all pair vectors in $(\Xi, \Xi)^n$ have a corresponding vector in Ξ^n, because they may have two different readings of the same symbol in two adjacent pairs. Pair-vectors that have corresponding symbol vectors will be called *consistent*.
We will hereby be interested in cases where some of the read pairs are corrupted versions of the true symbol pairs. The main error model considered for symbol pairs is when the number of pair-errors is bounded by an integer t, defined as *t-pair error* below.

Definition 2.*(t-Pair Error)*
Let $\boldsymbol{x} = [x_0, \ldots, x_{n-1}]$ be a vector in Ξ^n. A pair vector $\overset{\leftrightarrow}{u} = [(\lhd u_0, \rhd u_0), \ldots, (\lhd u_{n-1}, \rhd u_{n-1})]$ is the result of a t-pair error from $\boldsymbol{x}$ if $|\{i : (\lhd u_i, \rhd u_i) \neq (x_i, x_{i+1})\}| \leqslant t$. Indices are taken modulo n and $(a, b) = (c, d)$ if both $a = c$ and $b = d$.

A. Conditions for symbol-pair error correctability

After defining the symbol-pair error model, the next natural step is to prove necessary and sufficient conditions on the code for achieving correctability of symbol-pair errors. A central element in the characterization of correctability is the *pair distance*, $D_p(\cdot, \cdot)$ defined below.

Definition 3.*(Pair Distance) Let $\overset{\leftrightarrow}{u}, \overset{\leftrightarrow}{v}$ be two pair-vectors in $(\Xi, \Xi)^n$. The pair distance between $\overset{\leftrightarrow}{u}$ and $\overset{\leftrightarrow}{v}$ is defined as*

$$D_p\left(\overset{\leftrightarrow}{u}, \overset{\leftrightarrow}{v}\right) = |\{i : (\lhd u_i, \rhd u_i) \neq (\lhd v_i, \rhd v_i)\}|.$$

So the pair-distance between two pair-vectors counts how many of the n symbol-pairs differ between them, or in other words, the pair-distance is the Hamming distance over the alphabet (Ξ, Ξ). For notational aesthetics, when a consistent pair-vector is used as an argument to the pair distance, its symbol-vector notation may appear instead of its pair-vector one, i.e.

$$D_p(\boldsymbol{x}, \boldsymbol{y}) \triangleq D_p(\pi(\boldsymbol{x}), \pi(\boldsymbol{y})).$$

The pair-distance is related to the Hamming distance in the following manner.

Proposition 1. *For $\boldsymbol{x}, \boldsymbol{y}$ in Ξ^n, let $0 < D_H(\boldsymbol{x}, \boldsymbol{y}) < n$ be the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. Then*

$$D_H(\boldsymbol{x}, \boldsymbol{y}) + 1 \leqslant D_p(\boldsymbol{x}, \boldsymbol{y}) \leqslant 2D_H(\boldsymbol{x}, \boldsymbol{y}).$$

In the extreme cases in which $D_H(\boldsymbol{x}, \boldsymbol{y})$ equals 0 or n, clearly $D_p(\boldsymbol{x}, \boldsymbol{y}) = D_H(\boldsymbol{x}, \boldsymbol{y})$.

Proof: Define the set $S_H = \{j : x_j \neq y_j\}$. If $D_H(\boldsymbol{x}, \boldsymbol{y}) = d$, then $|S_H| = d$. In addition, define the set $S_p = \{i : (y_i, y_{i+1}) \neq (x_i, x_{i+1})\}$. Each index $j \in S_H$ appears in exactly two pairs of S_p, which gives the upper bound. Since each pair has exactly two indices, it may seem that the tightest lower bound is d and not $d + 1$. However, if $d < n$ there is at least one pair with only one of its indices $i, i + 1$ in S_H. ■
Note that for the trivial code $\mathcal{C} = \Xi^n$, the minimum pair distance between distinct codewords is 2, hence it can detect a single pair-error.
Proposition 1 can be regarded as a corollary to the following theorem.

Theorem 2. *For two words $\boldsymbol{x}, \boldsymbol{y}$ in Ξ^n with $0 < D_H(\boldsymbol{x}, \boldsymbol{y}) < n$, define the set $S_H = \{j : x_j \neq y_j\}$. Let $S_H = \cup_{l=1}^{L} B_l$ be a minimal partition of the set S_H to subsets of consecutive*[2] *indices (Each subset $B_l = [s_l, e_l]$ is the sequence of all indices between s_l and e_l, inclusive, and L is the smallest integer that achieves such partition). Then*

$$D_p(\boldsymbol{x}, \boldsymbol{y}) = D_H(\boldsymbol{x}, \boldsymbol{y}) + L$$

Proof: The requirement that the partition be minimal guarantees that there are no two adjacent indices $i, i + 1$ that belong to different subsets of S_H (otherwise the two subsets can be merged resulting in a smaller partition). Therefore, the pair distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be calculated as the sum of the sizes of the pair subsets $\{(s_l - 1, s_l), (s_l, s_l + 1), \ldots, (e_l, e_l +$

[2] indices may wrap around modulo n

$1)\}$. The number of pairs in each pair subset l equals $|B_l|+1$, hence the sum equals $\sum_{l=1}^{L}|B_l| + L = D_H(\boldsymbol{x},\boldsymbol{y}) + L$. ■

Proposition 1 can be obtained from Theorem 2 by noting that $1 \leqslant L \leqslant D_H(\boldsymbol{x},\boldsymbol{y})$.

The pair-distance shares the following simple properties with the well-studied Hamming distance.

- $D_p(\boldsymbol{x},\boldsymbol{y}) \geqslant 0$ and $D_p(\boldsymbol{x},\boldsymbol{y}) = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{y}$
- $D_p(\boldsymbol{x},\boldsymbol{y}) = D_p(\boldsymbol{y},\boldsymbol{x})$ (symmetry).
- $D_p(\boldsymbol{x},\boldsymbol{y}) \leqslant D_p(\boldsymbol{x},\overset{\leftrightarrow}{u}) + D_p(\overset{\leftrightarrow}{u},\boldsymbol{y})$ (triangle inequality).

Hence the set Ξ^n with the pair-distance is a *metric space*. The first two properties are obvious. To prove the triangle inequality, observe that if for some i we have $(x_i, x_{i+1}) \neq (y_i, y_{i+1})$, then at least one of $(x_i, x_{i+1}) \neq (\triangleleft u_i, \triangleright u_i)$ and $(\triangleleft u_i, \triangleright u_i) \neq (y_i, y_{i+1})$ has to be satisfied.

These properties of the pair-distance enable its use in the statement of necessary and sufficient conditions for pair-error correctability. Define a code $\mathcal{C} \subset \Xi^n$, and let

$$d_p = \min_{\boldsymbol{x}\in\mathcal{C},\boldsymbol{y}\in\mathcal{C},\boldsymbol{x}\neq\boldsymbol{y}} D_p(\boldsymbol{x},\boldsymbol{y})$$

be the minimum pair-distance of $\mathcal{C}$. A necessary and sufficient condition for correctability of t pair-errors is provided in the following proposition.

Proposition 3. *A code $\mathcal{C}$ can correct t pair-errors if and only if $d_p \geqslant 2t+1$.*

The proof of this proposition is essentially the same as in the Hamming case.

As it turns out, the necessary and sufficient conditions no longer match when the received pair-vectors are consistent. In the case where decoder inputs are consistent, there is a gap of one between the necessary and sufficient conditions. We first prove a (weaker) necessary condition for that case, and later show that it is tight.

Theorem 4. *When decoder inputs are consistent pair-vectors, a code $\mathcal{C}$ can correct all t-pair errors only if $d_p \geqslant 2t$.*

A proof of the theorem readily follows from the following lemma.

Lemma 5. *If $D_p(\boldsymbol{x},\boldsymbol{y}) = 2t-1$, then there exists a word $\boldsymbol{z} \in \Xi^n$ such that $D_p(\boldsymbol{x},\boldsymbol{z}) = t$ and $D_p(\boldsymbol{z},\boldsymbol{y}) \leqslant t$.*

Proof: Define the set $S_H = \{j : x_j \neq y_j\}$, and let $S_H = \cup_{l=1}^{L} B_l$ be its minimal partition to subsets with consecutive indices. For the selection of the word $\boldsymbol{z}$, we now specify how to construct a set T_H from S_H. Define a counter κ and initialize it to $\kappa = t$. If there is a subset $B_l = [s_l, e_l]$ with size $\kappa - 1$ or less, add it to T_H, subtract its size plus one from κ, and repeat the step with the new κ value. When there is no subset smaller than κ, pick any subset $B_l = [s_l, e_l]$, add the subset $[s_l, s_l + \kappa - 2]$ (of size $\kappa - 1$) to T_H, and terminate. Now define the word $\boldsymbol{z}$ as follows:

$$z_j = \begin{cases} y_j & \text{if } j \in T_H \\ x_j & \text{otherwise} \end{cases}$$

Each subset added to T_H contributes a pair-distance increase between $\boldsymbol{x}$ and $\boldsymbol{z}$ amounting to its size plus one. Hence $D_p(\boldsymbol{x},\boldsymbol{z}) = t$, as a result of the initialization $\kappa = t$. Denote the number of subsets in the minimal partition of T_H by L_1. Denote the number of subsets in the minimal partition of $S_H \setminus T_H$ by L_2. Since at most one subset out of the L subsets of S_H is split between T_H and $S_H \setminus T_H$, then $L_1 + L_2 \leqslant L + 1$. That fact gives the last inequality in the following

$$D_p(\boldsymbol{z},\boldsymbol{y}) =$$

$$|S_H| - |T_H| + L_2 = (2t - 1 - L) - (t - L_1) + L_2 \leqslant t$$

The first equality is from Theorem 2, with $D_H(\boldsymbol{z},\boldsymbol{y}) = |S_H| - |T_H|$; the second equality is also from Theorem 2, with $D_p(\boldsymbol{x},\boldsymbol{y}) = 2t-1$ and $D_p(\boldsymbol{x},\boldsymbol{z}) = t$. ■

The weaker necessary condition is tight since there exist codes with $d_p = 2t$ that *can* correct all t-pair errors resulting in consistent pair-vectors, such as the code $\{00000, 01110\}$ with $d_p = 4$, which can correct all 2-pair errors. (For example, the received word 01000 in pair-distance 2 from the all-zero codeword is in pair-distance 3 from the codeword 01110, so can be decoded correctly.).

III. Code Constructions and Decoding

A. Constructions from Hamming-metric codes

The treatment of adjacent symbols as pairs is reminiscent of the well studied problem of correcting error bursts. Therefore, it is not surprising that *interleaving*, a standard method for error-burst correction, is found useful for the symbol-pair problem as well. In Proposition 1, a potential factor-two gap between the Hamming distance and the pair distance is shown. As a consequence, using codes in the Hamming metric for pair-error correction is sub-optimal. To close the factor-two gap, while still using codes in the Hamming metric, the method of interleaving can be invoked.

Theorem 6. *Let $\mathcal{C}^{(1)}$ be an (n, M_1, d_H) code, and $\mathcal{C}^{(2)}$ be an (n, M_2, d_H) code, using the standard notation of (N, M, D) to denote a length N code with M codewords and minimum Hamming distance D. Then the code obtained by interleaving codewords of $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$ as in equation* (1) *is a $(2n, M_1M_2, d_H)$ code with $d_p = 2d_H$.*

$$\begin{array}{|c|c|c|c|c|c|c|}\hline \mathcal{C}_0^{(1)} & \mathcal{C}_0^{(2)} & \mathcal{C}_1^{(1)} & \mathcal{C}_1^{(2)} & \cdots & \mathcal{C}_{n-1}^{(1)} & \mathcal{C}_{n-1}^{(2)} \\ \hline \end{array} \tag{1}$$

Proof: For any two codewords of the interleaved code, the Hamming distance on $\mathcal{C}^{(1)}$'s or $\mathcal{C}^{(2)}$'s coordinates (or both) is at least d_H. Since none of the differing coordinates of the same $\mathcal{C}^{(i)}$ are in consecutive locations, the number of consecutive subsets is at least d_H, and by Theorem 2, $d_p \geqslant d_H + d_H = 2d_H$. The reverse inequality $d_p \leqslant 2d_H$ is proved by fixing a codeword of $\mathcal{C}^{(1)}$ and taking two codewords of $\mathcal{C}^{(2)}$ at distance d_H as two codewords of the interleaved code. ■

B. Direct cyclic-code construction

While codes constructed via interleaving obtain optimal pair-distance for their Hamming distance (factor 2), they are in general inferior to directly constructed codes, even if the constituent Hamming-metric codes are themselves optimal. As an example, we take the shortened $[30, 22]$ cyclic code generated by the polynomial $1 + x^2 + x^3 + x^8$, whose minimum pair-distance is $d_p = 7$. This code can correct 3 pair-errors, one more than the $[30, 22]$ code obtained from interleaving the $[15, 11]$ Hamming code with itself, whose minimum pair-distance is $d_p = 6$. This example also proves that, at least for these parameters, more pair-errors can be corrected than Hamming errors, since there is no $[30, 22]$ code that corrects 3 errors in the Hamming metric.

C. Decoding

So far, for the correctability proofs of sub-section II-A, a decoding function in the pair-metric was assumed, without regard to the algorithmic aspects of achieving such a decoder (enumerating the pair-distances from the received pair-vector to all codewords and finding the closest one is always possible, though not practical). Similarly to decoding in the Hamming metric, there is a need to devise efficient decoding algorithms that will allow the implementation of coding schemes in the pair-metric. A first such attempt is to reduce the pair-decoding function to a Hamming-decoding function, as specified in Algorithm 1 below.

Algorithm 1. *Let* $\overset{\leftrightarrow}{u} = [(\lhd u_0, \rhd u_0), \ldots, (\lhd u_{n-1}, \rhd u_{n-1})]$ *be the received pair-vector. Define the* n *symbols of the vector* z *as*

$$z_i = \begin{cases} \lhd u_i & \text{if } \lhd u_i = \rhd u_{i-1} \\ * & \text{otherwise} \end{cases}$$

The symbol $*$ *represents symbol erasure and is used when symbol hypotheses from the two pairs are in conflict. The vector* z *is now input to an error/erasure decoder in the Hamming metric.*

While any code (either interleaved or direct) for the pair-error model can be decoded using Algorithm 1, the question to ask at this instance is whether this decoder provides the decoding guarantees of Proposition 3. The answer turns out to be *no* in general, and *yes* for interleaved codes. To prove that the algorithm is inferior, in general, to a bounded-distance pair decoder we show an example. Suppose a single pair-error correcting code with the two codewords $\{00000, 01100\}$ (minimum pair-distance 3) is used, and the pair-vector $\overset{\leftrightarrow}{u} = [(0,0), (1,1), (0,0), (0,0), (0,0)]$ is received. Then Algorithm 1 will transform $\overset{\leftrightarrow}{u}$ into $z = [0, *, *, 0, 0]$, and a Hamming decoder will fail to decode (both codewords are equally likely given the decoder input). On the other hand, a pair-decoder will detect that $\overset{\leftrightarrow}{u}$ is at pair-distance 1 to 00000 and at pair-distance 2 to 01100.

To show equivalence to bounded-distance pair-decoding for interleaved codes, we observe that a chain of ℓ consecutive pair-errors induces up to 2 symbol erasures and $\ell - 1$ symbol errors. For odd ℓ, each constituent code $\mathcal{C}^{(i)}$ suffers one erasure and $(\ell - 1)/2$ errors. For even ℓ, one constituent code suffers two erasures and $\ell/2 - 1$ errors and the other suffers zero erasures and $\ell/2$ errors. When taking the weighted sum of $2 \cdot \#\text{errors} + \#\text{erasures}$, each code has a worst case sum of t, where t is the number of pair-errors. Substituting $d_p = 2d_H$ from Theorem 6 into Proposition 3 gives $t \leqslant \lfloor (2d_H - 1)/2 \rfloor = d_H - 1$, and from elementary coding theory the Hamming decoders of the constituent codes will be able to correct an error/erasure weighted sum of $d_H - 1$.

IV. Bounds on Code Sizes

The existence of necessary and sufficient conditions for pair-error correctability allows the derivation of upper and lower bounds, respectively, on the code size. A well known technique, used for both types of bounds, is to count the number of Ξ^n words in distance d from a given word. In the Hamming-distance metric, this counting task is very simple, and is used to derive the sphere-packing (upper) bound and the Gilbert-Varshamov (lower) bound, among many other bounds [2]. Given a word of Ξ^n, the pair-distance metric entails the complication of having part of the pair-error vectors (the consistent ones) result in words of Ξ^n, while others, non-consistent pair-error vectors, result in non-consistent pair-vectors. Thus the challenge is to count only the consistent pair-vectors at pair-distance d from the given word. Theorem 2 guides the solution toward solving the following combinatorial problem.

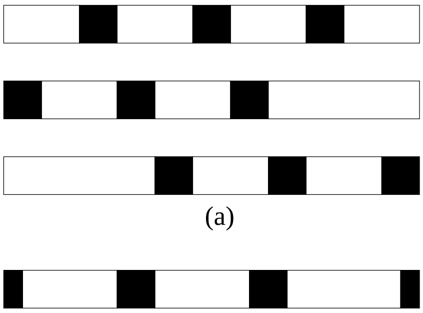
Problem 7. *Count how many of the subsets of the coordinate set* $[0, n-1]$ *have size* l*, and minimal partition of* $L = d - l$ *(cyclically) consecutive subsets.*

If this problem is solved, then all words that differ from the given word on these size-l subsets are known to be at pair-distance d from that word. Let $D(n, l, L)$ be the number of size l subsets of $[1, n]$ that occupy L cyclically consecutive subsets. The closed form formula for $D(n, l, L)$ is given in the following theorem.

Theorem 8. *For any triple* $n > l \geqslant L$,

$$D(n, l, L) = \binom{l-1}{L-1} \left[\binom{n-l-1}{L} + 2\binom{n-l-1}{L-1} \right] \quad (2)$$
$$+ \binom{n-l-1}{L-1}\binom{l-1}{L} \quad (3)$$

Proof: A subset that meets the (n, l, L) specification has one of the layouts depicted in Figure 2. The dark rectangles represent elements in the size-l subset. The white rectangles represent elements *not* in the subset. The three layouts in (a)


(a)

(b)

Figure 2. Layouts of (n, l, L) subsets. (a) Non-all-around layouts. (b) All-around layouts.

are ones that have no wrap around from $n-1$ to 0. The layout in (b) has a consecutive subset that wraps around. For any L, dark and white rectangles are placed in alternation (Figure 2 presents an example with $L=3$). Each rectangle represents a subset with size *strictly* larger than 0. The sizes of the dark rectangles in each layout sum to l. The sizes of the white rectangles sum to $n-l$. From elementary counting, the number of ways to place m identical elements in k numbered bins that are non-empty is $\binom{m-1}{k-1}$, [3, Ch.13]. The identical elements are the dark/white indices, and the bins are the dark/white rectangles of consecutive indices. We observe that the dark and the white elements can be independently grouped to rectangles, and each such grouping gives a distinct (n,l,L) subset. Therefore, the number of (n,l,L) subsets from each layout is the product of the number of dark groupings by the number of white groupings. In particular, the 4 layouts in Figure 2 have the following m,k parameters, listed from top to bottom.

1) Darks: $m=l$, $k=L$. Whites: $m=n-l$, $k=L+1$
2) Darks: $m=l$, $k=L$. Whites: $m=n-l$, $k=L$
3) Darks: $m=l$, $k=L$. Whites: $m=n-l$, $k=L$
4) Darks: $m=l$, $k=L+1$. Whites: $m=n-l$, $k=L$

Now the closed form expression immediately follows with (2) for the non-all-around layouts and (3) for the all-around layouts. ■

It is worth noting interesting special cases of $D(n,l,L)$ from Theorem 8.

- $D(n,l,1)=n$ (a single set with arbitrary shift)
- $D(n,l,L)=0$ if $L>l$ or $L>n-l$.
- For $D(n,l,l)$ the contribution of (3) is zero (singleton sets cannot go all around).

With a closed form formula for $D(n,l,L)$ it is possible to obtain a closed form formula for the number of Ξ^n words at pair-distance d from a given word. For that purpose we now define $\mathcal{S}_d(\boldsymbol{x})$, the radius-$d$ *pair-sphere* around a word $\boldsymbol{x}$.

Definition 4. *For a word $\boldsymbol{x}\in\Xi^n$, define the pair-sphere $\mathcal{S}_d(\boldsymbol{x})$ as the set of all $\boldsymbol{y}\in\Xi^n$ such that $D_p(\boldsymbol{x},\boldsymbol{y})=d$.*

The size of d-spheres is given in the following proposition.

Proposition 9. *For any $\boldsymbol{x}\in\Xi^n$, and $d>0$*

$$|\mathcal{S}_d(\boldsymbol{x})|=\sum_{l=\lceil d/2\rceil}^{d-1} D(n,l,d-l)(q-1)^l$$

where $q=|\Xi|$ is the size of the alphabet.

Note that $|\mathcal{S}_1(\boldsymbol{x})|=0$, as needed, and $|\mathcal{S}_2(\boldsymbol{x})|=n(q-1)$, which coincides with the Hamming sphere of radius 1. The *pair-ball* $\mathcal{B}_d(\boldsymbol{x})$ consists of all words with pair-distance d or less from $\boldsymbol{x}$, and clearly

$$|\mathcal{B}_d(\boldsymbol{x})|=1+\sum_{i=1}^{d}|\mathcal{S}_i(\boldsymbol{x})|.$$

The ability to enumerate pair-balls allows generalizing useful bounds to the pair-metric.

Proposition 10.*(Pair-Sphere Packing Bound) If $\mathcal{C}\subset\Xi^n$ is a code with M codewords that corrects all t-pair errors, then*

$$M|\mathcal{B}_t(\boldsymbol{x})|\leqslant q^n.$$

This sphere-packing bound in the pair-metric can be used to prove that the 3-pair-error correcting code in sub-section III-B is optimal, in the sense that there is no $[30,22]$ code that corrects 4 pair-errors. A similar generalization can be obtained for the Gilbert-Varshamov bound.

Proposition 11.*(Pair Gilbert-Varshamov Bound) There exists a code $\mathcal{C}\subset\Xi^n$ with M codewords and minimum pair-distance d if*

$$M|\mathcal{B}_{d-1}(\boldsymbol{x})|\leqslant q^n.$$

V. A Graph Theoretic Lens

As an aid to their study, codes for pair-read channels are now observed through a graph-theoretic lens. The symbols of the code alphabet can be regarded as *vertices* of a graph, while symbol-pair reads are regarded as *edges* of the graph. Pursuing such a view of the pair-error model is done for these two purposes.

1) To accommodate restrictions on code vectors, where due to reader limitations, not all symbol pairs are allowed at adjacent locations.
2) To potentially harness graph theoretic results for better code constructions.

Let $V=\{v_1,\ldots,v_q\}$ be a set of graph vertices. Let $E\subseteq V\times V$ be a set of directed edges, each denoted as an ordered pair of vertices $(v^{\text{out}},v^{\text{in}})$. A *walk* on a graph is a list of n, not necessarily distinct, edges from E: $[(v_1^{\text{out}},v_1^{\text{in}}),(v_2^{\text{out}},v_2^{\text{in}}),\ldots,(v_n^{\text{out}},v_n^{\text{in}})]$, where $v_i^{\text{in}}=v_{i+1}^{\text{out}}$, for all $1\leqslant i<n$. A walk is *closed* if in addition $v_n^{\text{in}}=v_1^{\text{out}}$. Now define the graph $\mathcal{G}$ to be the complete directed graph with self loops and q vertices. The problem of constructing codes over Ξ^n with minimum pair-distance d can be formulated as

Problem 12. *Given the graph $\mathcal{G}$, find a set of length n closed walks whose pairwise overlap is at most $n-d$ edges.*

As an example we draw in Figure 3 the graph that corresponds to the binary alphabet.

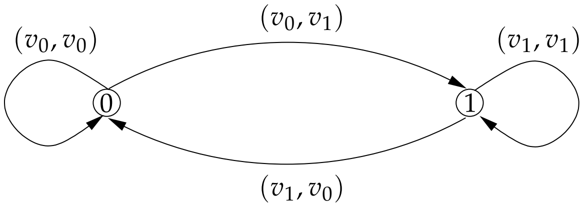


Figure 3. Complete directed graph for the binary alphabet.

Once the coding problem is formulated in graph theoretic terms, restrictions on symbol pairs can be accommodated by solving Problem 12 on different (non complete) graphs $\mathcal{G}$. For example, if the code symbols represent sequences from two mutually-orthogonal sets, then the complete *bipartite* graph should be considered as the underlying graph for the problem (symbol pairs are allowed across sets but not within the sets).

A. Walks with bounded edge overlap

We now turn to discuss possible methods to obtain sets of closed walks with bounded overlap as a way to construct codes for pair-errors. Two such methods are briefly described, omitting some of the details. The first one, *global-edge codes*, uses Hamming-metric codes over an alphabet whose size equals the number of edges in the graph $\mathcal{G}$. The second one, *local-edge codes*, uses Hamming-metric codes over an alphabet whose size equals the out-degree of the vertices of $\mathcal{G}$ (assuming that $\mathcal{G}$ is a regular graph).

1) Global-edge codes: As noted earlier in the section, each codeword in a pair-error correcting code is a closed walk on $\mathcal{G}$, and hence can be written as a tuple $[e_1, \ldots, e_n]$, where $e_i = (v_i^{\text{out}}, v_i^{\text{in}}) \in E$ is an edge of $\mathcal{G}$. Therefore, if we define the alphabet $\Sigma = \{1, \ldots, |E|\}$, then walks that are part of a code with minimum Hamming distance of d over Σ are guaranteed to have an edge overlap of at most $n - d$. Consequently, the pair-error correcting code will be the intersection over Σ^n of the Hamming-metric code and the set of closed walks.

2) Local-edge codes: Instead of using global identifiers for the edges in the walk, one can specify the walk by writing down the start vertex and indices to the local outgoing edges from the start vertex and from subsequent ones. So a walk can be written as $[v; \epsilon_1, \ldots, \epsilon_n]$, where $v \in V$ and $\epsilon_i \in \{1, \ldots, \delta\} \triangleq \mathcal{E}$, with δ being the out-degree of the vertices of $\mathcal{G}$. Taking the local edge indices ϵ_i from a code with minimum Hamming distance d, now over $\mathcal{E}$, can guarantee the required bound on edge overlap.

The global-edge code construction has the advantage of using codes over large alphabets, but on the flip side many of the codewords will have to be discarded since they do not represent valid walks on the graph. The local-edge construction uses codes over small alphabets, but all or most of its codewords represent valid walks and thus can be used by the pair-error code.

References

[1] G. Benelli, C. Bianciardi, and V. Capellini, “Redundancy bounds for multiple-burst error-correcting codes,” *Electronic Letters*, vol. 13, no. 13, pp. 389–390, June 1977.

[2] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[3] J. van Lint and R. Wilson, *A Course in Combinatorics, second edition*. Cambridge UK: Cambridge University Press, 2001.

[4] S. Wainberg and J. Wolf, “Burst decoding of binary block codes on q-ary output channels,” *IEEE Transactions on Information Theory*, vol. 18, no. 5, pp. 684–686, 1972.