Suboptimal Decoding of Linear Codes: Partition Technique

Ilya Dumer, Member, IEEE

Abstract—General symmetric channels are introduced, and near-maximum-likelihood decoding in these channels is studied. First, we define a class of suboptimal decoding algorithms based on an incomplete search through the code trellis. It is proved that the decoding error probability of suboptimal decoding is bounded above for any $q$-ary code of length $n$ and code rate $r$ by twice the error probability of its maximum-likelihood decoding and tends to the latter as $n$ grows. Second, we design a suboptimal trellis-like algorithm, which reduces the known decoding complexity of the order of $q^m \min (r, 1-r)$ operations to that of $q^{m(r-1)}$ operations for all cyclic codes and virtually all long linear codes. We also consider the corresponding bounds for concatenated codes. An important corollary is that this suboptimal decoding can provide complexity below the lower bounds on trellis complexity at a negligible expense in terms of decoding error probability.

Index Terms—Trellis, symmetric channels, suboptimal decoding, information subsets, output-matched weights, shortest paths.

I. INTRODUCTION

A. Summary

In this paper we address a problem of designing general decoding algorithms, which can achieve the performance of maximum-likelihood (ML) decoding with reduced complexity. The problem is now extensively studied, and a number of papers effectively utilize near-maximum-likelihood (NML) decoding algorithms specially designed for some short codes. However, such an approach is mostly verified by an extensive computer simulation performed for bounding either decoding error probability or decoding complexity, or both. Also, newly considered algorithms designed in this way provide insufficient evidence as to how decoding complexity and performance depend on code length. Therefore, in this paper we use a different approach and tackle the problem of general NML decoding design in an analytical way. We start with the following definition.

Definition 1: Let $\Psi(C)$ be an ML decoding algorithm for a given $q$-ary code $C$ of length $n$ and code rate $r$ in a given channel $U$. We say that $\Psi$ is a suboptimal (NML) decoding algorithm for a sequence of codes $C_i(q, n_i, r_i)$ of growing lengths $n_i$, and rates $r_i \rightarrow r$, if the asymptotic equality $\lim [P_\Psi(C_i)/P_\Psi(C_i)] = 1$ holds for the decoding error probabilities $P_\Psi(C_i)$ and $P_\Psi(C_i)$ of the algorithms $\Psi$ and $\Psi$, as $i \rightarrow \infty$.

In the sequel, we introduce a class of symmetric channels, which can be continuous or discrete, additive or nonadditive, with memory or without it. Then we define a class of decoding algorithms, which provide suboptimal performance for any sequence of linear or nonlinear codes used over any symmetric channel. This gives us an analytical tool for estimating decoding error probability for long codes and guarantees that asymptotic performance is as good as that of ML decoding. As for short codes, the decoding error probability will be bounded by twice the error probability of ML decoding for any code and any symmetric channel.

Second, we address the problem of designing suboptimal algorithms with sufficiently low complexity. In doing this, we consider only memoryless symmetric channels and focus on cyclic or linear codes. An advantage of such an approach is that we do not utilize code structure and consider general classes of codes in a purely combinatorial way. The drawback of this approach directly stems from its advantage in a sense that code structure is not used for further reductions in decoding complexity. However, this study already yields some encouraging results. Namely, we construct a suboptimal trellis-like algorithm, whose complexity is bounded above by the exponential order of $q^{n(r(1-r)}$ operations for all cyclic codes and virtually all long linear codes. In other words, a substantial complexity reduction relative to the known upper bound of $q^m \min (r, 1-r)$ operations is obtained at a negligible expense in terms of decoding performance. Also, the achieved complexity already falls off below the known lower bounds on conventional trellis complexity in a wide interval of code rates. We also use similar tools to obtain greater reductions in decoding complexity for concatenated codes.

In essence, this paper can be divided into three parts. In the introductory part (Sections I–IV) we first give a review of known results relevant to this paper and proceed with more detailed exposition of our technique in Section I-B. Then in Sections II and III we give the basic definitions and exhibit a simple example of suboptimal soft-decision decoding for a code of length 32 and code rate 0.5 used over an AWGN channel. This example is later used as a benchmark for more general discussions and explicit proofs. Finally, Section IV serves as a transition to the sequel of the paper. Here we present an updated version of the Evseev's results [19] for minimum-distance (MD) decoding, which are most relevant to our discussion. This is done not only to pay tribute to a great paper, whose significance is likely underestimated.
by the scientific community, but mostly because the sequel of our paper can well be treated as generalizations of [19] to NML decoding. These generalizations include arbitrary symmetric channels, nonlinear codes, and also decoding lists of varying size, which allow us to trade code performance for decoding complexity. Our first minor divergence with [19] already appears in Section IV, where we employ a simple tool of a sliding window instead of a more sophisticated technique of arbitrary partitions used in [19]. This allows us to simplify the proofs, include cyclic codes in our consideration, and eliminate a subexponential factor of the order \( \exp (\sqrt{n}) \) from the complexity of MD decoding.

The second part of our paper (Sections V and VI) concerns the output decoding error probability for codes used over general symmetric channels. First, in Section V we introduce these channels, verify their properties, and give some examples. Then in Section VI we consider decoding algorithms, which achieve suboptimal performance for any sequence of codes. The developed tools are different from those used in Section IV.

The last part (Sections VII–X) focuses on decoding complexity of suboptimal decoding. We develop combinatorial algorithms for linear, cyclic, and concatenated codes. Throughout the paper we also raise a few open problems. Finally, the last Section X attempts to bring somewhat a deeper insight into the problem. We also mention here some directions for further research.

**B. Review of Results**

The problem of designing sufficiently simple suboptimal algorithms (SA) is far from solution to date. In this regard we mention three groups of algorithms relevant to our discussion. First, decoding design is well developed for some short codes used over the AWGN channels. This approach was launched in [10] and [13] and then employed in a great number of publications (see recent papers [1], [24], [26], [38], [39], [41], and references therein). In particular, the Chase’s algorithms [10] exhibit good results for moderate signal-to-noise ratio in terms of both the complexity and performance, and still can serve as a benchmark for newly devised algorithms. Some of the later designed algorithms make use of highly decomposable structure of the codes considered (like the Golay code and RM codes). On the other hand, neither simulation results, nor specific code structure can be exploited for general sets of long linear codes. More to the point, for \( q \)-ary linear codes of rate \( 0 < r < 1 \) complexity of ML decoding grows exponentially with code length \( n \) as \( q^{n \delta (r)} \), so that complexity exponent \( c (r) \) is bounded away from 0. Therefore, we are mostly interested in the complexity exponent \( c (r) \) as function of rate \( r \), and will later decrease the known complexity exponent of ML decoding for general linear codes by allowing a marginal increase in their decoding error probability. This problem was extensively studied for minimum-distance decoding in the second group of papers.

These papers develop techniques of hard-decision decoding for general linear codes. For a given received vector \( a^* \) such a decoding retrieves the closest codeword in the Hamming metric, no matter how far it is, and thus provides full MD decoding. It is proved in [5] that MD decoding of general linear codes is NP-complete. It is also proved in [8] and [34] that general MD decoding is yet likely to be exponential even if the code may be preprocessed for any amount of time prior to transmission (this statement is explicit provided that the conventional assumptions on the polynomial hierarchy hold; see [8]). Now for given \((q, n, r)\) let \( D \) denote the corresponding Gilbert distance

\[
D = \max \left\{ d: \sum_{i=0}^{d-1} (q-1)^i \left( \begin{array}{c} n \\ i \end{array} \right) \leq q^{n(1-r)} \right\}.
\]

It is proved by Blinovskii [6] that virtually all long linear \((q, n, r)\) codes (with the exception of exponentially decreasing fraction of codes as \( n \rightarrow \infty \)) have covering radius \( \rho = D + O(\ln n) \). In other words, correction of slightly more than \( D \) errors necessarily provides full MD decoding for virtually all long linear codes. Another important result by Evseev [19] implies that correction of only \( D \) or fewer errors at most doubles the decoding error probability of MD decoding for any linear \((q, n, r)\) code. As for the complexity \( q^{n \delta} \) of MD decoding, the following bounds on the exponent \( \delta \) are currently known.

Let

\[
H_q(z) = -z \log_q z - (1 - z) \log_q (1 - z) + z \log_q (q - 1)
\]
denote the \( q \)-ary entropy, \( 0 < z < 1 \), and \( \delta = H_q^{-1}(1-r) \) be the asymptotic relative GV distance \( D/n \) for given \( q, r \) (here also \( \delta \leq (q - 1)/q \)). First, note that the basic bound \( c_0 = \min (r, 1-r) \) can be obtained for all linear codes via the full search either through all \( q^{nr} \) codewords or all \( q^{n(1-r)} \) coset leaders. Then it was proved in [19] that hard-decision decoding of virtually all \( q \)-ary linear codes of rate \( r \) can be executed with complexity exponent \( c_1 = r(1-r) \). Complexity exponents \( c_2 = \min (r, H_q(\delta) - 1 + r) \) (see [33]), \( c_3 = (1-r)/2 \) from [15], and \( c_4 = (H_q(\delta) - (1-r) H_q(\delta/1-r))/\log_q 2 \) from [28], [11] also hold for virtually all \( q \)-ary linear codes of rate \( r \). Finally, the algorithm of [16] combines the algorithms of [11], [15], [19], and [28], as special cases and provides a slightly better estimate \( c_5 \) relative to \( c_4 \). For binary codes, these bounds \( c_0, \ldots, c_5 \) are plotted versus code rate \( r \) in Fig. 1(a).

In essence, the algorithms in [15], [16], and [19] are based on a common “correct a lightly corrupted subblock—and—re-encode” strategy. In particular, the following heuristic arguments can be used to justify the bound \( c_1 \) from [19]. The received block \( a^* \) is supposedly corrupted by \( D \) or fewer errors. Then for some \( j = 1, \ldots, n \) the subblock \( \bar{L}(j, s) = \{j, \ldots, j + s - 1\} \), which consists of \( s = \tau r n \) cyclically consecutive positions, is corrupted by \( r D \) or fewer errors. In

\[1\]

We call this the Gilbert distance to distinguish it from the Varshamov distance

\[
D = \max \left\{ d: \sum_{i=0}^{d-2} (q-1)^i \left( \begin{array}{c} n \\ i \end{array} \right) < q^{n(1-r)} \right\}.
\]

Obviously, for any length \( D - D \geq 1 \) with asymptotic equality.
this way, execute
\[ K = \sum_{i=0}^{rD} (q-1)^i \binom{rn}{i} \]

trials of "correcting" \( rD \) or fewer errors in each subblock \( L(j, s) \) and re-encode it as an information subset in every trial. In this process of multistage testing of every subblock \( L(j, s) \) we necessarily find any codeword \( c \) such that \( d(c, a^*) \leq D \). The number of trials \( K \) is readily verified to yield the complexity exponent \( c_1 \).

Finally, the third group of papers concerns the problem of ML soft-decision decoding for general linear codes. Most papers develop ML decoding via trellis realization of a code. Such a description was introduced in [2], [20], [35], and [42]. In [42], trellis ML decoding was first explored for block and concatenated codes. For arbitrary memoryless channel trellis decoding of any \( g \)-ary linear code of rate \( r \) is proved in [42] to yield the complexity exponent \( c_0 = \min (r, 1-r) \). In [14], decomposable trellises of concatenated codes were used to obtain the complexity exponent \( c_2 = \min (r, H_2(2\delta) - 1 + r) \) for the best binary concatenations (meeting the asymptotic Gilbert–Varshamov bound). The same exponent \( c_2 \) was also derived in [29], [30] for the block codes obtained from convolutional codes. It is shown in [43] that \( c_2 \) is the tight lower bound on trellis complexity of binary codes, provided that the GV bound is tight.

Basically, a trellis \( T(C) \) of a linear code \( C \) is a finite directed graph with \( n+1 \) levels. Let \( H \) denote the parity-check matrix of code \( C \) with the columns \( h_1, \ldots, h_n \). Every input vector \( (x_1, \ldots, x_n) \) is represented by the unique \textit{path} through the trellis. For any level \( j = 1, \ldots, n \) let \textit{subpath} \( \overline{x}_j = (x_1, \ldots, x_j) \) be a vertex mapped by the syndrome \( s_j = \sum_{i=1}^{j} x_i h_i \) of \( \overline{x}_j \). In other words, all the subpaths \( \overline{x}_j \) with the same syndrome \( s_j \) merge in the same vertex. Any \textit{edge} \( x_j \) of a path is ascribed by the weight \( w_j(x_j) \), which depends on the \textit{a posteriori} probability \( p(x_j \mid y_j) \) of the symbol \( x_j \) (explicit definitions are given in Section II below). In decoding we seek a shortest code path terminating at a null vertex, and for every consecutive level \( j \) leave only the shortest subpath arriving at any vertex \( s_j \). So the number of paths to be looked through is bounded above by the number \( q^k \) of code paths and by the number \( q^{n-k} \) of possible vertices. This is readily verified to give the complexity exponent \( c_0 \).

Apart from the upper bound \( c_0 \), the lower bounds on trellis complexity were derived in [32], [37], and [43] for all codes. In general, code design with small trellis complexity has been of great interest (see [9], [21–23], [36], [37], and many more references therein). Much less is known about suboptimal soft-decision decoding. The major problem is that for soft-decision decoding a set of the most probable error patterns depends on a received vector \( y \), in contrast to its hard-decision counterpart, for which this set forms the ball of radius \( D \) regardless of \( y \). Yet, the significant reductions in decoding complexity described above for MD and near-MD decoding, raise the following natural question. Are these reductions only appropriate for MD decoding, or can they also be achieved for ML and NML decoding? More precisely, we formulate the following problem.

\textit{Open Problem 1:} Is it possible to obtain complexity exponents \( c_1, \ldots, c_5 \) for near-maximum-likelihood soft-decision decoding of linear codes?

In this paper, we give a partial answer to this problem and prove that virtually all linear codes and all cyclic codes admit suboptimal decoding with complexity exponent \( c_1 = r(1-r) \). We also develop general techniques for bounding the decoding error probability, which is useful for various suboptimal algorithms. In particular, this technique will also be used in our next correspondence, where a different algorithm will be employed to obtain the exponent \( c_3 = (1-r)/2 \) for all linear codes, and the better bound \( c_4 = r(1-r)/(1+r) \) for virtually all \( q \)-ary linear codes. However, more sophisticated tools, used in deriving \( c_6 \), do not allow us to include cyclic codes in our consideration. These complexity exponents \( c_1, c_3, c_4 \), and \( c_5 \) are plotted in Fig. 1(b) along with the known upper and lower bounds for ML decoding of binary codes. For completeness we also present a complexity bound for suboptimal decoding of binary concatenated codes meeting the Gilbert–Varshamov bound.

In order to decrease decoding complexity we treat code trellises in a way different from the above description. Namely, any specific output \( y \) above affects only the lengths \( w_j(x_j) \) rather than trellis realization itself. In what follows, we employ the \textit{incomplete} trellises, which are constructed depending upon a received output and therefore include many fewer paths. In essence, we will prove that suboptimal performance is achieved as long as decoder keeps track of the sublist \( S \) including about \( N \approx q^{n(1-r)} \) shortest paths among all possible \( q^n \) paths. In other words, the error probability of ML decoding is being virtually unaffected by losing the shortest \textit{code path} whenever it is too long to be placed into the list \( S \) (Theorems 2 and 3). With these statements at hand, we are mostly interested in retrieving the shortest path from the whole space, and can proceed without considering a code structure. In order to decrease decoding complexity, we then study the projections \( \textit{(subpaths)} \) of these \( N \) shortest paths onto any subset \( L \) of \( s \approx nr \) consecutive positions. For \( s \mid n \) we prove that any path \( x \in S \) produces at least one "good" subpath, which belongs to the list of about \( K = N^{s/n} \approx q^{nr(1-r)} \) shortest subpaths on some subset \( L \). In the general case, we prove that at least one subpath of length \( s \) is "well decomposable" in a sense that it consists of a few good subpaths of shorter lengths \( s_1, \ldots, s_f \). The latter form the sublists of sizes \( N^{s_1/n}, \ldots, N^{s_f/n} \) and again yield at most \( K \) subpaths of length \( s \), when linked together. In addition, these \( K \) subpaths can be designed with complexity of the order \( K \approx q^{nr(1-r)} \) (Theorem 4). Basically, we run at most \( n \) trials and consider about \( nr \) consecutive positions in every trial. Then suboptimal performance is attained by seeking about \( q^{nr(1-r)} \) "well decomposable" subpaths in each subtrellis. Each considered subpath is re-encoded, and the shortest path
\[ \delta = H_2^{-1}(1 - r) \text{-relative GV distance for binary codes} \]

\[ c_0 = \min(r, 1 - r) \]

\[ c_1 = r(1 - r) \]

\[ c_2 = \min(r, H_2(2\delta) - 1 + r) \]

\[ c_3 = (1 - r)/2 \]

\[ c_4 = H_2(\delta) - (1 - r)H_2(\frac{\delta}{1 - r}) \]

\[ c_5 = \min_{\alpha, \beta} \max\{(1 - \alpha)(1 - H_2\left(\frac{\beta}{1 - \alpha}\right)), 1 - r - (1 - \alpha)H_2\left(\frac{\beta}{1 - \alpha}\right) - \frac{\delta}{2} H_2\left(\frac{\beta}{\alpha}\right)\} \]

\[ \alpha, \beta : r \leq \alpha \leq 1, \max(0, \alpha - \beta + 1) \leq \beta \leq \min(\alpha, \delta) \]

\[ c_6 = r(1 - r)/(1 + r) \]

\[ c_{\text{con}} = -(H_2^{-1}(1 - r)) \log_2(2^{1-r} - 1) \]

\[ \mathcal{C}' \text{ and } \mathcal{C}'' \text{ - lower bounds on trellis complexity (from [32] and [43])} \]

Fig. 1. (a) Complexity exponents for MD decoding (versus code rate). (b) Complexity exponents for near-ML decoding (versus code rate).

on the length \( n \) is chosen. The overall complexity is then bounded above by the same order of \( q^{nr(1-r)} \) operations.

II. NOTATIONS AND DEFINITIONS

Let \( X = \{a^{(1)}, \ldots, a^{(q)}\} \) be a q-ary input alphabet and \( Y \) be an output alphabet, which may be infinite. When transmitted, input sequences \( x = (x_1, \ldots, x_n) \in X^n \) of length \( n \) are converted into sequences \( y = (y_1, \ldots, y_n) \) of the output space \( Y^n \). For given \( X, Y, \) and \( n \) define a channel by a conditional probability \( P(y|x) \) of receiving a sequence \( y \in Y^n \) provided that a sequence \( x \in X^n \) is transmitted (here \( P \) is a conditional probability density for continuous sets \( Y \)). We use a code \( C(q, n, M) \subseteq X^n \) with \( M \) input sequences. For an arbitrary decoding algorithm \( \Psi \) let the set \( \Psi^{-1}(c) = \{y : \Psi(y) = c\} \) denote the decoding domain for a codeword \( c \in C \), and let \( \Psi^{-1}(c) \) be the complementary set in \( Y^n \). For a given output \( y \), ML decoding \( \psi(y) = c \) retrieves a codeword \( c \in C \) such that \( P(y|c) \geq P(y|c') \) for any codeword \( c' \neq c \). In the sequel, we suppose that all \( M \) messages are equiprobable. Then ML decoding retrieves a codeword \( c \in C \) with the maximum \( a \) posteriori probability

\[ P(c|y) \geq P(c'|y), \quad c \neq c \]  \hspace{1cm} (2)

and minimizes the decoding error probability

\[ P_{\text{err}}(C) = \frac{1}{M} \sum_{c \in C} P(\Psi^{-1}(c)|c). \]

among all decoding algorithms \( \Psi \) (see [25, sec. 5.2]).

Let \( p(b|a) \) denote any conditional probability distribution defined over the pairs \( (b, a) \), \( b \in Y, a \in X \). Below we mostly consider memoryless channels, for which the equality

\[ P(y|x) = \prod_{j=1}^{n} p(y_j|x_i) \]
holds for any pair \( x, y \). Now for a given output \( y \) define the “output-matched weight” of any input symbol \( a \in X \) in any position \( j \) by

$$ w_j(a) = -\ln p(a|y_j) \tag{4} $$

and the total “weight” of any sequence \( x = (x_1, \ldots, x_n) \) by

$$ w_y(x) = \sum_{j=1}^{n} w_j(x_j). \tag{5} $$

Then the most probable codeword \( c \) is the lightest one in terms of the minimum weight \( w_y(c) \). Let also \( a_j^* \) be the most probable input in position \( j \), so that \( w_j(a_j^*) \geq w_j(a) \) for each input \( a \neq a_j^* ). Hard-decision \) decoder receives the most probable input sequence \( a^* = (a_1^*, \ldots, a_n^*) \) and retrieves the codeword \( c \) which is the nearest one to \( a^* \) in terms of the Hamming distance: \( d(a^*, c) \leq d(a^*, c'), c' \neq c \). Obviously, such a decoding provides ML performance in a q-ary discrete narrow-sense symmetric channel (see [25]), where \( Y = X \), and any one-symbol substitution in \( a^* \) is equiprobable. By contrast, soft-decision decoding corresponds to a general situation, where an output alphabet \( Y \) includes \( X \) and usually forms a continuous set (say, \( R^2 \) in an AWGN channel, or \( R^2 \) in a q-PSK channel).

Below we address a problem of how efficiently decoding complexity can be traded for decoding error probability. We define the complexity \( \Omega_{\Psi}(C) \) of an algorithm \( \Psi(C) \) by the maximum number of operations \( \Omega(y) \) required to terminate the decoding of any output \( y \)

$$ \Omega_{\Psi}(C) = \max_{y \in Y^n} \Omega(y). \tag{6} $$

(Note that the decoding result is not necessarily a codeword if \( \Psi \) is an incomplete algorithm.) Here the operations are taken over \( q \)-ary input symbols and real numbers. Also, for a given set \( C \) of the \( (q, n, M) \) codes consider the maximum complexity \( \Omega_{\Psi}(C) = \max_{C \subseteq \mathcal{C}} \Omega_{\Psi}(C) \) over all codes in the set \( C \). In order to consider a symmetric behavior of decoding algorithms we need the following definitions.

Definition 2: We say that the function \( f(n) \) has polynomial order \( n^{b} \) and write \( \deg(f) = b \), if \( \log(|f(n)|)/\log n \to b \), as \( n \to \infty \).

Note that \( \deg(f) \) above does not depend on the logarithm base used.

Definition 3: We say that the function \( f(n) \) has exponential order \( q^{n^{b}} \) and write \( \ord_q(f) = b \), if \( \log(|f(n)|)/\log n \to b \), as \( n \to \infty \).

Definition 4: We say that an infinite sequence of codes \( C_i(q, n_i, M_i) \) of growing lengths \( n_i \to \infty \) and code rates \( r_i(C_i) = (\log_q M_i)/n_i \) is a \((q, r)\) sequence, if \( r_i \to r \), as \( i \to \infty \).

Finally, the sets \( C_i \) of \( q \)-ary codes are called the \((q, r)\)-sets if the rates of all the codes \( C_i \in C_i \) tend to \( r \) as \( i \to \infty \).

III. EXAMPLE FOR SOFT-DECISION DECODING

Consider a linear \((2, 32, 2^{16})\)-code used over a binary AWGN channel. Suppose that both halves \( I_1 = \{1, \ldots, 16\} \) and \( I_2 = \{17, \ldots, 32\} \) form the information subsets. Let \( y \) be the received vector in the 32-dimensional Euclidean space \( R^{32} \). First, we use (4) to calculate 64 output-matched weights \( \{w_j(y)\} \) for each symbol \( a = 0, 1 \), and each position \( j = 1, \ldots, 32 \). Then we seek the shortest code path via the following incomplete trellis algorithm.

1) Design \( K = 256 \) shortest subpaths on the set \( I_1 \). First, all \( K \) subpaths \( x_0 = (x_1, \ldots, x_8) \) at the eighth level are present in the descendent order according to their weights

$$ w_y(x_0) = \sum_{j=1}^{8} w_j(x_j). $$

Then in each following step \( j = 9, \ldots, 16 \) we keep seeking \( K \) shortest subpaths \( x_j = (x_1, \ldots, x_j) \) arriving at the level \( j \). Namely, we take \( K \) shortest subpaths \( x_{j-1} = (x_1, \ldots, x_{j-1}) \) from the previous step, and consider \( 2K \) subpaths \( (x_{j-1}, 0) \) and \( (x_{j-1}, 1) \). We recalculate their weights \( w_y(x_{j-1} + x_j(0)) \) and \( w_y(x_{j-1} + x_j(1)) \), respectively, and sort out \( K \) shortest subpaths among \( 2K \) of them.

2) All \( K \) shortest subpaths found in the step 16 are re-encoded onto the length 32 and the shortest code path \( x' \) is chosen among them.

3) Repeat the above procedure for the set \( I_2 \) on the second half by designing \( K \) shortest subpaths on this set (as in step 1), and finding the shortest re-encoded code path \( x'' \) (as in step 2).

4) Choose the shortest path among \( x', x'' \).

The results below prove that the decoding error probability of full ML decoding is at most doubled by setting the threshold of \( K = q^{n(1-\epsilon)} \) subpaths that is, 256 in our case) regardless of a specific code structure and signal-to-noise ratio. In other words, in our example we use two incomplete trellis diagrams on 16 positions, with at most \( K \) subpaths looked through on either half, whereas the full trellis can include as many as \( q^{n(1-\epsilon)} = 65, 636 \) paths for an arbitrary \((2, 32, 2^{16})\)-code and about a thousand paths (see [31]) for a highly decomposable RM code. Also, the above defined lists \( (x_j, 0) \) and \( (x_j, 1) \) are already present in every step \( j \). Subsequently this will allow us to rule out the \( K/2 \) longest subpaths prior to any sorting and bound the number of presorted subpaths by \( 3K/2 \) in every step.

Finally, the above mentioned upper bound on decoding error probability was also checked against the simulation results prepared by Eiguren. In this simulation, the lists of size \( K = 16, 32, 64, \ldots, 512 \) were tested for the \((2, 32, 2^{16})\) RM code used over the AWGN channels with signal to noise ratio of 0–6 dB. It turned out that changing \( K \) from 512 to 64 left the decoding error probability virtually unaffected for any signal-to-noise ratio (with a divergence in the second–third significant digit). In other words, the effective size of suboptimal list turned out to be around 64 instead of 256 above, thus raising a question on the actual size \( K \) for AWGN channels.

IV. NEAR-MD DECODING

An important paper by Evseyev [19] was the first to consider suboptimal-wise hard-decision decoding. Let a \( q \)-ary input
alphabet $X$ form a group and let $Y = X$. Define the componentwise subtraction $y - x$ in $X^n$, and consider an additive channel $(X^n, Y^n, P(y|x))$ with conditional probability $P(y|x)$ defined as $P(y-x)$, where the function $P$ stands for any probability distribution in $X^n$. Let also $C = C_0(q, n, M)$ be a group code in $X^n$. Then $X^n$ is fully decomposed into $T = q^n/M$ disjoint cosets $C_0(q), C_1(q), \ldots, C_{T-1}(q)$. Define a coset leader $\varepsilon_i$ as the most probable vector in $C_i(q)$ (such that $p(\varepsilon_i) \geq p(\varepsilon)$ for any $\varepsilon \in C_i(q)$). Let $e(y)$ denote the coset leader of the coset that includes the received vector $y$. Then ML decoding $\psi(y)$ is derived as $\psi(y) = y - e(y)$. In other words, coset leaders form the set $E = \{e_0, e_1, \ldots, e_{T-1}\}$ of error vectors chosen by ML decoder as $y$ runs through the cosets $C_0(q), C_1(q), \ldots, C_{T-1}(q)$.

Now let $B = \{b_0, b_1, \ldots, b_{T-1}\}$ denote a set of $T$ most probable vectors in the whole set $X^n$, so that $p(b') \geq p(b'')$ for any $b' \in B$ and any $b'' \notin B$. Define the decoding algorithm $\Psi$ by

$$\psi(y) = \begin{cases} \psi(y), & \text{if } e(y) \in B \\ \emptyset, & \text{otherwise}. \end{cases} \tag{7}$$

In other words, algorithm $\Psi$ fails to produce any codeword, if an error vector does not belong to the set $B$. Then the important Evseev's lemma gives the following upper bound on the decoding error probability $P_\Phi(C)$ of the algorithm $\Psi$.

**Lemma 2** Evseev [19]: The inequality

$$P_\Phi(C) \leq 2P(E) \tag{8}$$

holds for any group code $C$ in a discrete additive channel.

**Proof:** First, for any input $c \in C$ the ML decoder improperly decodes any error pattern taken from $X^n \setminus E$. So, the equality (3) implies that $P_\Phi(C) = P(X^n \setminus E)$, where $P$ denotes the total probability of a taken set. In addition, (7) implies that $\Psi(y)$ also fails to decode any error vector from $E \setminus B$. So

$$P_\Phi(C) = P(X^n \setminus E) + P(E \setminus B).$$

By definition of the set $B$, $P(B) \geq P(E)$. So

$$P(E \setminus B) \leq P(X^n \setminus B) \leq P(X^n \setminus E)$$

and (8) follows. $\square$

Now apply Lemma 1 to a $q$-ary narrow-sense symmetric channel. Then ML decoding is converted into the MD decoding in the Hamming space. In addition, $T$ most probable error vectors belong to the ball of radius $D$ centered at 0, in accordance with (1). In other words, the algorithm $\Psi(y)$ provides MD decoding within the sphere of radius $D$ centered at the received vector, and fails to decode whenever no codewords belong to this sphere.

The following basic lemma simplifies the original algorithm of [19] and will be used throughout this paper for further generalizations. Let

$$L = L(j, s) = \{j, j + 1 \pmod{n}, \ldots, j + s - 1 \pmod{n}\}$$

be a subset of $s$ cyclically consecutive positions starting from the position $j$. We will also consider the collection of $n$ samples $L(j, s)$ as $j$ runs through the set $0, \ldots, n-1$ and call it the sliding window $L(s)$. $L(j, s)$ is said to be an information subset, if any two different codewords disagree on $L(j, s)$.

Let $k = \log_q M$ be the number of information symbols and $\tau = k/n$ be the code rate.

**Lemma 2:** For virtually all linear $(q, n, M)$-codes the sliding window of length $s = k + \lceil 2 \log_q n \rceil$ forms $n$ information subsets.

**Proof:** Consider the set of linear codes generated by all $q$-ary $(k \times n)$ matrices $G$. For a given $G$, the subset $L \in L(s)$ is an information subset if the truncated $(k \times s)$ submatrix $G_L$ is nonsingular. First, count the fraction $P_L$ of nonsingular submatrices $G_L$ for a given $L$. First $\tau$ linearly independent rows in $G_L$ rule out $q^\tau$ options for the next row, and therefore

$$P_L = \frac{q^{k-s} - q^\tau}{q^{k-1}} > 1 - \sum_{i=0}^{k-1} q^{-s+i} > 1 - q^{-s+k}.$$  

Here we repeatedly use the obvious inequality $(1-\alpha)(1-\beta) > 1 - \alpha - \beta$ for positive $\alpha$, $\beta$. So the fraction of matrices $G$ with at least one singular submatrix $L$ is bounded above by $nq^{-s+k}$ and tends to zero for the above given $s = k + \lceil 2 \log_q n \rceil$. Similarly, the fraction of singular $(k \times n)$ matrices $G$ is bounded above by

$$1 - \sum_{i=0}^{k-1} q^{-s+i}/q^\tau < \sum_{i=0}^{k-1} q^{-n+i} < q^{-n+k},$$

and also tends to zero. Finally, the declining fraction of singular matrices $G$ yields the declining fraction of degenerate linear codes. So the above considered matrices generate virtually all $(q, n, M)$-codes of dimension $k$ with each $L \in L(s)$ as an information subset. $\square$

Note also that $s = k$ can be taken for cyclic codes. Now consider all cyclic codes and virtually all linear codes satisfying Lemma 2. Then any possible error pattern of weight $D$ or less produces a subblock $e$ of weight $t = \lfloor Da/n \rfloor$ or less for at least one subset $L \in L(s)$. So we take every subset $L$ and re-encode each information subblock $y_L - e$, as long as all $e$ are exhausted. Re-encoding takes at most $O(n^2)$ operations and produces at most one codeword in every trial. Every newly re-encoded codeword replaces the best codeword found before if it is closer to the $y$. For the given $s = k + \lceil 2 \log_q n \rceil$ the overall number of trials

$$F = n \sum_{i=0}^{s} (q-1)^i$$

is readily verified to be of the exponential order

$$\text{ord}_q(F) = \text{ord}_q((q-1)^s) = n r (1 - r)$$

according to (1). Hence this algorithm $\Psi(y)$ justifies the following theorem.

**Theorem 1** Evseev [19]: Virtually all linear codes of any code rate $\tau$ can be decoded with decoding error probability bounded by twice the error probability of minimum-distance decoding, and decoding complexity of the exponential order of $q^{nr(1-r)}$.

Obviously, the algorithm $\Psi(y)$ is also valid for all cyclic $(q, n, M)$-codes. We note without proof that a few nonexponential improvements can be provided for this algorithm. First,
if the ratio $s/n$ is fixed as a constant $p/q$, then only $q$ subsets $L(j, s)$ may be chosen. In particular, this ratio $s/n = r$ is fixed for cyclic codes. Also, it can be proved that more restrictive constraints on vectors $e$ can be imposed thus reducing the number of trials for each subset $L(j, s)$ by a linear factor. For example, cyclic shifts of the received vector allow us to consider only those vectors $e$, whose weight on the first $i$ positions is bounded above by $|D_{i/n}|$ simultaneously for all $i = 1, \ldots, n$ (more details can be found in [4], where similar consideration was employed for bounding weight spectra of some cyclic codes).

V. SYMMETRIC CHANNELS

In Sections V and VI, we generalize Lemma 1 for:

- an arbitrary symmetric channel with either discrete or continuous output;
- an arbitrary nonlinear $(q, n, M)$-code (in essence, any subset of $M$ input signals);
- a class of algorithms $\Psi_i$, which trade the decoding error probability $P_{\Psi_i}(C)$ for the number of the designed paths.

First, in this section we define symmetric channels and set some conditions to verify whether a channel is symmetric. We also present some important examples. For a given channel $(X^n, \Omega^n, P(y|x))$ consider any finite output subset $Y_{\alpha} \subseteq \Omega^n$ of the size $Q_{\alpha} = |Y_{\alpha}|$. Let $P_{\alpha} = |\{P(y|x)\}|$ denote the corresponding matrix of transition probabilities (densities) taken over all the entries $x \in X^n$, $y \in Y_{\alpha}$.

**Definition 5**: A channel $(X^n, \Omega^n, P(y|x))$ is called symmetric if the output space $\Omega^n$ can be decomposed into disjoint finite subsets $Y_{\alpha}$ in such a way that in each matrix $P_{\alpha}$ each row is a permutation of any other row and each column is a permutation of any other column.

Note that for a discrete channel this definition coincides with the classical one given in [25, sec. 4.5]. The lemma below also shows that a memoryless channel is symmetric, if so is the corresponding alphabet channel $(X, Y, P(b|a))$.

**Lemma 3**: A symmetric alphabet channel $(X, Y, P(b|a))$ generates the symmetric memoryless channel $(X^n, \Omega^n, P(y|x))$.

**Proof**: Let the output alphabet $Y$ be decomposed into the sets $Y_{\alpha}$ satisfying the above definition, with $\beta$ running through some set $\Omega$. Consider the set $\Omega^n$ of $n$-tuples $\alpha = (\beta_1, \ldots, \beta_n)$, where $\beta_i \in \Omega$ for each $i = 1, \ldots, n$. First, note that $\Omega^n$ splits into the disjoint subsets $Y_{\alpha} = \{y_1, \ldots, y_n\} \subseteq \Omega^n$ as $\alpha$ runs through $\Omega^n$. In other words, we define every subset $Y_{\alpha}$ as the direct product $Y_{\beta_1} \times \cdots \times Y_{\beta_n}$ of the corresponding alphabet subsets, and induce the decomposition $\Omega^n = \{y_1, \ldots, y_n\} = \cup_{\alpha} Y_{\alpha}$.

Second, for each symbol $y_{\alpha} \in Y_{\alpha}$, the set of probabilities $p(y_{\alpha}|a)$ takes on the same set of values as $a$ runs through $X$. Hence, so does their product

$$P(y|x) = \prod_{j=1}^{n} p(y_j|x_j)$$

as $x$ runs through $X^n$. Finally, the above arguments also show that for each $x \in X^n$ the set of probabilities $P(y|x)$ takes on the same set of values as $y$ runs through any $Y_{\alpha}$. So all three conditions of Definition 5 hold for the induced partition $\Omega^n = \cup_{\alpha} Y_{\alpha}$.

The following examples show how the symmetric properties of an $n$-dimensional memoryless channel can be verified at the alphabet level.

**Example 1**: Let BPSK with the set $X = \{-1, +1\}$ of two antipodal signals be used $n$ times over the memoryless channel with additive white Gaussian noise (AWGN). Then $X^n = \{-1, +1\}^n$, $Y = \mathbb{R}$, and $Y^n = \mathbb{R}^n$. Decompose the output alphabet into disjoint subsets $Y_{\beta} = \{y\mid \beta\}$ defined for any $\beta \in [0, +\infty)$. Then any subset $X \times Y_{\beta}$ satisfies Definition 5, which, in turn, proves that an AWGN channel is symmetric. Note also that the induced subsets $Y_{\alpha}$ have size $2^{w(\alpha)}$, where $w(\alpha)$ is the number of positive components of the vector $\alpha$.

**Example 2**: More generally, let the $q$-ary alphabet be mapped onto the $q$-PSK set $X = \{e^{2\pi i/q}, \ldots, x^{q-1}\}$ and be used $n$ times over a memoryless channel with two-dimensional AWGN. Decompose the output set $Y$ of complex numbers into the disjoint subsets $Y_{\beta} = \{y\mid \beta\}$ defined for any complex $\beta = |\beta|e^{i\varphi}$ with $0 \leq \varphi < 2\pi/q$. In other words, $Y_{\beta}$ is obtained by rotating the point $\beta$ by the angle $2\pi/q$ and its multiples. Note that any $y \in Y_{\beta}$ gives the same set of Euclidean distances $D_y = \{d(y, x)\mid x \in X\}$. Also, any $x \in X$ gives the same set of Euclidean distances $D_x = \{d(y, x)\mid y \in Y\}$. Since the Euclidean distance fully determines probability density, the symmetric channel is obtained. Note that $|Y_{\alpha}| = q$, if $\beta \neq 0$, and $|Y_{\alpha}| = 1$, otherwise. So again the subsets $Y_{\alpha}$ have size $q^{w(\alpha)}$, where $w(\alpha)$ is the number of nonzero components in $\alpha$.

The following example shows that channels with memory can also provide symmetric properties.

**Example 3**: Let $X^n$ and $Y^n$ form a group and subgroup, respectively, $X^n \subseteq Y^n$. Consider an additive channel for which $P(y|x) = p(y-x)$ as above. Decompose the set $Y^n$ into the subgroup $X^n$ and its cosets $X^n(1), X^n(2), \ldots$ of equal size $q^n$. Obviously, this decomposition satisfies Definition 5. For example, the above defined discrete additive channel with $X^n = Y^n$ is symmetric.

**Example 4**: Consider a binary erasure channel with $X = \{0, 1\}$, $Y = \{0, 1, \ast\}$, defined by the matrix

$$P(y|x) = \begin{pmatrix} p' & p''' & 1 - p' - p''' \\ p'' & p'' & 1 - p'' \\ p' & p'' & 1 - p' \\ p'' & p'' & 1 - p'' \\ \end{pmatrix}$$

This nonadditive channel is well known to be a discrete-symmetric channel (see [25, sec. 4.5]) under the splitting $Y_1 = \{0, 1\}$ and $Y_2 = \{\ast\}$.

**Example 5**: In contrast to Example 2, amplitude modulation (AM) and quadrature amplitude modulation (QAM) give nonsymmetric channels, when used over the AWGN channels. In AM $q$ input points form the alphabet $X_n = \{i - h, |i| = 0, \ldots, q-1\}$ uniformly distributed within the interval $[-h, h]$ on $R^1$, where $h = (q-1)/2$. In QAM with $q = f^2$, input points form the square lattice $X_f \times X_f$ in $R^2$. Both channels necessarily include internal and external inputs for $q > 2$. This does not allow any output splitting, which is symmetric under the row permutations.

Another way to verify the symmetric properties of a channel is to define a set $F$ of transformations on $X$ and $Y$, which
are isomorphic with respect to the transition probabilities. For example, we can use the following “mapping” channels.

**Definition 6:** A memoryless channel is called a “mapping” channel if there exists a group $G = \{f_1, \ldots, f_q\}$ of $q$ mappings on $X$ and $Y$ such that the following conditions hold:

1. For any input $x \in X$ the set $F(x) = \{f_1(x), \ldots, f_q(x)\}$ forms $X$.
2. For any output $y \in Y$ the subset $F(y) = \{f_1(y), \ldots, f_q(y)\}$ consists of $q$ points and can be generated by any of them.
3. For any $x \in X$, $y \in Y$, and $f \in F$ the equality $P(y|x) = P(f(y)|f(x))$ holds.

**Lemma 4:** Any “mapping” channel is symmetric.

**Proof:** The set of mappings $F$ splits the output $Y$ into disjoint finite subsets $F(y)$ of size $q$. Consider the probability matrix $P(z|x)$, $z \in F(y)$, $x \in X$. For any pair $x_1, x_2 \in X$ a mapping $f \in F$ exists, such that $f(x_1) = x_2$. Therefore, the row $P(z|x_2)$ is obtained from $P(z|x_1)$ by the permutation $z \rightarrow f(z)$. The same holds for any pair of columns $P(z_1|x)$ and $P(z_2|x)$ by the permutation $x \rightarrow f(x)$, if $z_2 = f(z_1)$.

One can easily verify that the above Example 2 would represent a mapping channel once the origin is removed from the output set $Y = R^2$. Then the remaining set $R^2 \setminus 0$ and the input set $X$ satisfy the above properties when rotated by the angle $2\pi/q$ and its multiples. We will use this fact in the next section for estimating the decoding error probability of codes used over AWGN channels.

**VI. SUBOPTIMAL DECODING**

In this section, we introduce a class of decoding algorithms, and prove that they guarantee suboptimal performance for any symmetric channel and any code. We first consider general (nonlinear) codes used over additive discrete channels, and bound the decoding error probability depending on the size of a decoding list. Then we turn to general symmetric channels and prove a similar statement with a marginal degradation in the obtained decoding performance. We also explain this fact. Then we prove that mapping channels and AWGN channels provide the same bound as that derived for discrete additive channels. Finally, we briefly comment on some further refinements to the obtained bounds.

**A. Definition of Decoding Algorithms**

Consider a given channel $(X^n, Y^n, P(y|x))$. For any output $y \in Y^n$ order all $q^n$ inputs $x \in X^n$ into a set $X_y^n$ with nonincreasing a posteriori probabilities $P(x|y)$:

$$X_y^n = \{x(1), x(2), \ldots, x(q^n)\} | P(x(i)|y) \geq P(x(i+1)|y), \quad i = 1, \ldots, q^n-1. \quad (9)$$

Note that for memoryless channels the set $X_y^n$ gives the same descendant order of all the inputs $x$ with respect to their “weights” $w_y(x)$ (5). Also, $X_y^n$ is not unique if the equality $P(x'|y) = P(x'^*|y)$ holds for at least two inputs $x'$, $x'^* \in X^n$. At this point, we arbitrarily fix an ordering of these inputs. Let $N_y(x)$ denote the rank of $x$ in the set $X_y^n$. Define the set $X_y^n(N) = \{x(1), \ldots, x(N)\}$ of $N$ most probable inputs for given $y$ and $N$, $1 \leq N \leq q^n$. Again, $X_y^n(N)$ is unique, if (and only if) $P(x(N)|y) > P(x(N+1)|y)$. Consider a code $C(q, n, M) \subseteq X^n$. Then ML decoding of the output $y \in Y^n$ retrieves the codeword $c^* = \psi(y)$ with the least rank

$$N_y(c^*) = N_y(c), \quad c, c^* \in C, \quad c^* \neq c. \quad (10)$$

Now for the given code $C$ define the decoding algorithm $\Psi_N$ by the following rule:

$$\Psi_N(y) = \begin{cases} c^*, & \text{if } N_y(c^*) \leq N \\ \emptyset, & \text{otherwise.} \end{cases}$$

In other words, for a given $y$ the algorithm $\Psi_N$ either retrieves $c^* = \psi(y)$ if $c^*$ is among $N$ most probable inputs taken from the whole space $X^n$, or fails to decode, otherwise. Let $P_N$ be the error probability of $\Psi_N$ for the given code $C$. Obviously, $P_{N+1} \leq P_N$ for any $N$. Also, $\Psi_N$ is identical to ML decoding for any $N > q^n - M$, since then at least one codeword is being placed into the list $X_y(N)$. In general, the decoding domain $\Psi_N^{-1}(c)$ of a codeword $c$ is obtained from the domain $\psi^{-1}(c)$ of ML decoding by eliminating all the outputs $y$ such that $N_y(c) > N$. So

$$P_N = P_0 + \frac{1}{M} \sum_{c \in C} P\{y|c) : y \in \psi^{-1}(c), N_y(c) > N\} \quad (11)$$

where $P\{\cdot\}$ stands for the conditional probability of the subset within the braces.

**B. Suboptimal Decoding in Additive Discrete Channels**

The following statement generalizes Lemma 1 for an arbitrary code $C(q, n, M)$ used over an additive discrete channel, and for any size of the decoding list $N \geq T$, where $T = \lceil q^n/M \rceil$ as above.

**Theorem 2:** For a code $C(q, n, M)$ used over an additive discrete channel $(X^n, Y^n, P(y|x))$ the error probability of $\Psi_N$-decoding can be bounded above for any $N \geq T$ by

$$P_N \leq P_0 \left(1 + \frac{T}{N}\right). \quad (12)$$

**Proof:** Order all the (error) vectors $e \in X^n$ into a descendant order with respect to their probabilities $P(e)$, so that

$$P(e(1)) \geq P(e(2)) \geq \cdots \geq P(e(q^n)).$$

Any (say, lexicographic) order is used for vectors $e$ with equal probabilities. Then we generate the induced order

$$X^n_y = \{y - e(1), \ldots, y - e(q^n)\}$$

for any $y$. Let $p_i = P(e(i))$. Consider the $(q^n \times q^n)$ matrix $p(x, y)$ with rows enumerated by inputs $x$, columns enumerated by outputs $y$, and entries $p(x, y) = P(y|x)$. First, note that each (ordered) number $p_i$ occurs exactly once in every row $x$; namely, in the column $x + e(i)$. Also, $p_i$ occurs once in every column $y$ in row $y - e(i)$.

Now for any subset of pairs $W \subseteq \{(x, y) : x \in C, y \in Y^n\}$ consider the sum

$$P(W) = \sum_{(x, y) \in W} p(x, y).$$
Then ML decoding is successful on the set \( E = \{ \psi(y), y \} \) of \( q^n \) distinct pairs, and fails on the remaining set \( \overline{E} = \{ C \times Y^n \} \setminus E \). So

\[
P_\psi = P(\overline{E})/M. \tag{13}
\]

Consider also the set

\[
S_N = \{(x, x + e(i))|x \in C, \ i = 1, \ldots, N\}
\]

of \( NM \) pairs \((x, y)\) that give the \( N \) largest entries \( p_1, \ldots, p_N \) in the code rows \( x \in C \). Obviously

\[
P(S_N) = M \sum_{i=1}^N p_i. \tag{14}
\]

Then the algorithm \( \Psi_N(y) = x \) properly decodes all the pairs \((x, y)\) from the subset \( E \cap S_N \), fails on the subset \( \overline{E} \) (similarly to ML decoding) and, in addition, leaves the pairs from the set \( E \setminus S_N \) undecoded. So

\[
P_N = (P(\overline{E}) + P(E \setminus S_N))/M. \tag{15}
\]

Now compare the above probabilities \( P_\psi \) and \( P_N \). First, note that the set \( S_T \) gives all \( MT \geq q^n \) largest entries \( p_1, \ldots, p_T \) taken from the code rows. Therefore, \( P(S_T) \geq P(E) \). This implies that

\[
P(E \setminus S_T) \leq P(\overline{S_T}) \leq P(\overline{E}).\tag{16}
\]

So \( P_T \leq 2P_\psi \), according to (13) and (14). Second, for any \( N \geq T \) the subsets \( E \setminus S_N \) and \( S_N \setminus S_T \) belong to the set \( \overline{S_T} \) and do not intersect. So

\[
P(E \setminus S_N) + P(S_N \setminus S_T) \leq P(\overline{S_T}) \leq P(\overline{E}), \quad N \geq T. \tag{17}
\]

The subset \( E \setminus S_N \) includes at most \( q^n \) pairs whose entries are bounded above by \( p_N \), so that

\[
P(E \setminus S_N) \leq q^n p_N \leq MT p_N. \tag{18}
\]

On the other hand

\[
P(S_N \setminus S_T) = M(p_{T+1} + \cdots + p_N) \geq M(N - T)p_N. \tag{19}
\]

Then the inequality

\[
P(E \setminus S_N)/P(S_N \setminus S_T) \leq T/(N - T)
\]

combined with (15), gives the inequality

\[
P(E \setminus S_N) \leq P(\overline{E}) T/N. \tag{20}
\]

Now (12) immediately follows from (13), (14), and (16). □

Note that the above proof makes use of a descendant ordering \( P(e(1)) \geq \cdots \geq P(e(q^n)) \) of the error vectors \( e \leq X^n \) in an additive channel. This ordering may be not unique. However, the induced orderings \( X_n \) match each other in such a way that the \( i \)th ordered entry \( p_i \) equally likely occurs in each column \( y \). The latter can only be achieved if the output subsets \( Y_n \) include at least \( q^n \) elements. Otherwise, the result is the multiple equal entries in each column. Then in general, the \( \Psi_N(y) \)-decoder can rank these equal entries in such a way that the code rows are more frequently placed below the average rank and are less frequently selected into the list \( X_n \). For example, this can happen in the erasure channel of Example 4 or in AWGN channels of Examples 1 and 2, if an output \( y \) includes zero symbols. Therefore, the error probability bounds obtained for the general symmetric channel in the following Theorem 3, are slightly weaker.

C. General Symmetric Channels

**Theorem 3**: The error probability of \( \Psi_N \)-decoding of a code \( C(q, n, M) \) used over a symmetric channel \((X^n, Y^n, P(y|x))\), can be bounded above for any \( N > T \) as

\[
P_N \leq P_\psi \left(1 + \frac{T}{N - T}\right). \tag{21}
\]

**Proof**: Decompose the output set \( Y^n \) into disjoint finite subsets \( Y_n \) satisfying Definition 5. For a given \( \alpha \) let

\[
P_{\psi, \alpha} = \frac{1}{M} \sum_{c \in C} P\{(y|c)|y \in Y_n \psi^{-1}(c)\}
\]

\[
P_{N, \alpha} = \frac{1}{M} \sum_{c \in C} P\{(y|c)|y \in Y_n \Psi_N^{-1}(c)\}
\]

be the decoding error probabilities (densities) of the algorithms \( \psi \) and \( \Psi_N \) obtained over the outputs taken from a subset \( Y_n \). Obviously, the inequality (17) holds if the similar inequality \( P_{N, \alpha} \leq P_{\psi, \alpha}(1 + T/(N - T)) \) does so simultaneously for all \( \alpha \). Let \( Q_\alpha \) be the size of \( Y_n \). Consider the \((q^n \times Q_\alpha)\) matrix \( p(x, y) \) with rows \( x \in X^n \), columns \( y \in Y_n \), and entries \( p(x, y) = P(y|x) \). Due to channel symmetry, all the columns run through the same set of entries \( p = \{p_1, \ldots, p_q\} \). As above, we use a descendant order in \( p \). The elements of any subset \( p_N \) occur at least \( N Q_\alpha \) times in \( p(x, y) \), with equality if \( p_N > p_{N+1} \). Each row includes at least \( f_N = N Q_\alpha /q^n \) entries from \( p(N) \), also due to channel symmetry. Then the set

\[
S_N = \{(x, y)|x \in C, y \in Y_n, p(x, y) \in p(N)\}
\]

includes \( |S_N| \geq M f_N \geq N Q_\alpha /T \) pairs in the code rows \( x \in C \). Also, consider the subset \( S \subseteq S_N \), which includes exactly \( Q_\alpha \) pairs with the largest entries in the \( M \) code rows. The set \( E = \{(\psi(y), y)\} \) of "ML-decoding" pairs includes \( Q_\alpha \) entries in \( M \) code rows. Since \( S \) has the same size and includes the largest entries, the inequality \( P(S) \geq P(E) \) holds.

We first suppose that \( p_N > p_{N+1} \) and prove that then the inequality (12) holds for any \( N \geq T \), as above in Theorem 2. Algorithm \( \Psi_N(y) \) leaves only the subset \( E \setminus S_N \) undecoded. Again, the subsets \( E \setminus S_N \) and \( S_N \setminus S_T \) belong to \( \overline{S} \) and do not intersect. So

\[
P(E \setminus S_N) + P(S_N \setminus S) \leq P(\overline{E}).
\]

Also, \( |E \setminus S_N| \leq Q_\alpha \) and \( |S_N \setminus S| \geq Q_\alpha (N/T - 1) \). Since \( S \subseteq S_N \) includes the largest entries from the subset \( \overline{S} \), we have inequalities (16) and (12), as above in the proof of Theorem 2.

Finally, we consider the general case. For a given \( Y_n \), let \( p_{K+1} = \cdots = p_N = p_{p_r} \) be the maximal subset of ordered equal entries including \( p_{p_r} \). In general, \( K \) and \( R \) can take arbitrary values satisfying the inequality \( K < N < R \). In contrast to the previous case, now the algorithm \( \Psi_N(y) \) can
leave out the “$p_N$ entries” in the code rows, if these entries are ranked as $p_{N+1}, \ldots, p_R$ in the column $y$. Correspondingly, the estimates deteriorate as follows. Similarly to $S_N$, the set

$$S_R = \{(x, y) | x \in C, y \in Y^*_\alpha, p(x, y) \in p(R)\}$$

includes at least $RQ_\alpha/R$ pairs. The remaining subset $S_R \setminus S$ includes at least $Q_\alpha(R/T-1)$ pairs, whose entries are bounded below by $p_N$. Also, $P(S_R \setminus S) \leq P(S) \leq P(\mathcal{E})$. On the other hand, the subset $E \setminus S_N$ includes at most $Q_\alpha$ pairs undecoded by the $\Psi_N(y)$. Even if these pairs fall within $S_R$, they may exhibit only the least possible entries $p_{N+1} = \cdots = p_R$. Therefore

$$P(E \setminus S_N) \leq P(S_R \setminus S)/(R/T-1) \leq P(\mathcal{E})/(R/T).$$

So $P_{\Psi,\alpha} \leq P_{\Psi,\alpha}(1 + T/(R - T))$ for every $\alpha$, and (17) follows.

D. Refinements

The above bounds (12) and (17) can be improved in a few ways. Below we briefly comment on these refinements. First, for some symmetric channels we can match an ordering of the rows with equal entries (as in Theorem 2) and obtain the bound (12). Namely, it can be done for mapping channels, for which the output alphabet can be decomposed into the orbits $Y_\alpha$ of size $q$ generated by any representative. Note that the additive discrete channel is a mapping channel, and so are AWGN channels with BPSK or $g$-PSK (Examples 1 and 2 above), if the zero point is eliminated from the output alphabet $Y$.

**Lemma 5:** The error probability of $\Psi_N$ decoding can be bounded above by the inequality (12) for any mapping channel and for any $N \geq T$.

**Proof:** Any $y = (y_1, \ldots, y_n)$ belongs to the orbit $Y_\alpha = F(y_1) \times \cdots \times F(y_n)$ of the size $q^n$. Arbitrarily fix an ordering of the elements in every $Y^\alpha$. For example, we can first map any $y \in Y^\alpha$ onto its hard-decision image $a^\alpha \in X^n$ (see Section II) and then rank each $y \in Y^\alpha$ according to the lexicographic order of its image $a^\alpha \in X^n$. Let $y'$ denote the first element in $Y^\alpha$. This element can be obtained from any $y \in Y^\alpha$ by the componentwise mappings, say, $y' = f(g) = (f_1(y_1), \ldots, f_n(y_n))$. As in the proof of Theorem 2 above, consider the $(q^n \times q^n)$ matrix $P(y|x)$ with rows $x \in X^n$ and columns $y \in Y^\alpha$. All the columns run through the same set of $p = \{p_1, \ldots, p_s\}$. First, set up any descendant order of the entries $P(y'|x)$ in the first column. Second, rank any entry $P(y|x)$ in the column $y$ by the rank of the same entry $P(y'|x)$ in the first column. Then each ordered entry $p_k$ appears exactly one time in each row and each column. Now the proof fully follows that of Theorem 2.

**Corollary 1:** For an AWGN channel with a $g$-PSK constellation the error probability of $\Psi_N$ decoding can be bounded above by the inequality (12) for any $N \geq T$.

**Proof:** Eliminate all the points with any zero components from $Y^n$. Then we have the mapping channel. On the other hand, the probability measure of the eliminated subset equals zero. So the output decoding probability is left the same and satisfies (12).

We also note that the bounds (17) can be linked together for different $N \geq T$. For example, the inequality

$$\sum_{i=2}^N (P_i - P_{\psi_i}) \leq P_{\psi_i}$$

holds for any $i = 2, \ldots$. This inequality shows that the extra decoding error probabilities $P_i - P_{\psi_i}$ of the algorithms $\Psi_{2T}, \ldots, \Psi_{iT}$ are totally bounded by $P_{\psi_i}$, and, therefore, decline on average faster than $P_{\psi_i}/(i-1)$. The proof is beyond the scope of this paper.

We conclude this section by the following major problem.

**Open Problem 2:** Generalize the upper bounds on the decoding error probability $P_N$ of suboptimal decoding for non-symmetric channels.

VII. PARTITION LEMMAS

Now we turn to suboptimal decoding for linear codes used over the memoryless symmetric channels. Then the “probability” ordering (9) is equivalent to the ordering

$$w(x(1)) \leq w(x(2)) \leq \cdots \leq w(x(q^n))$$

obtained with respect to the weights $w_j(x)$ (5). So the problem of effective $\Psi_N$ decoding is reduced to designing the list of $N$ lightest input vectors, or equivalently, $N$ shortest paths in the full trellis diagram. Now our main issue is how to project the screening on the length $n$ to a subblock $L(j, s)$ of length $s$. Then the decoding procedure can be completed by re-encoding these subpaths in accordance with Lemma 2. An important question is how many fewer subpaths can be considered in this way. We recall that in Section IV the problem of correcting $D$ errors on the length $n$ was accomplished by correcting at most $Ds/n$ errors on a subblock of the length $s$. In other words, for the Hamming weight the number of error patterns was reduced from $N = q^n/(1-r)$ to the order of $N^s/n$. Below we develop this technique for an arbitrary set of positive weights $w_j(x)$ and prove that only $K \approx N^s/n$ subpaths can be considered on the subblocks $L(j, s)$. The complexity of this design is also proved to have the order $K$. Since the paths are considered in the whole space $X^n$ rather than in the code $C$, this study is purely combinatorial.

A. Orderings on the Subsets

Now let $I = \{0, 1, \cdots, n-1\}$ be the set of all positions and let the positions of a subblock

$$L = \{j, j+1(mod n), \cdots, j+s-1(mod n)\}$$

of any length $s$ be arranged in the increasing order. Define the truncation $x_L = (x_i | i \in L)$ of a vector $x = (x_0, \cdots, x_{n-1}) \in X^n_q$ onto the set $L$ and its weight

$$w(x_L) = \sum_{i \in L} w_i(x_i)$$

(w fix $y$ and omit the subscript $y$ hereafter). Consider the set $X_L = \{x_L\}$ of $q^s$ truncated vectors and put them into the descendant order with respect to their weights $w(x_L)$. Below, for any subset $L$ we use the lexicographic order for
all the vectors of $X_L$ with equal weights, so that vector $x_i' = (x_i^{(i)}, i \in L)$ follows vector $x_\ell = (x_\ell^{(i)}, i \in L)$, if

$$\sum_{i \in L} x_i' q^i \geq \sum_{i \in L} x_i q^i.$$  

In other words, these vectors are ordered as $q$-digital numbers with digits on the subset $L$. Now we have the fixed ordering of subblocks (subpaths) on any $L$ with respect to their weights (lengths):

$$w(x_L(1)) \leq w(x_L(2)) \leq \cdots \leq w(x_L(q^s)).$$  

(19)

Let $N(x_L)$ be the rank of a vector $x_L$ in the ordering (19) and let $X_L(K) = \{x_L(1), \cdots, x_L(K)\}$ be the set of $K$ lightest subvectors on the set $L$. For $K > q^s$ we define $X_L(K)$ as $X_L(q^s)$. Note that, in general, the ordering of subpaths in (19) does not match the ordering (18) of the full paths, or those subpaths, which include more positions. However, we have the following lemma.

Lemma 6: For any vector $x \in X_q^n$ and any pair of subsets $L$ and $J$ such that $L \subset J \subset I$, the inequality

$$N(x_J) \geq N(x_L)N(x_{J \setminus L})$$  

(20)

holds.

Proof: Consider the subvectors on the set $J$ and let $A = \{a_L, a_{J \setminus L}\}$ be the subset of these subvectors such that $N(a_L) \leq N(x_L)$ and $N(a_{J \setminus L}) \leq N(x_{J \setminus L})$ for a given $x$. First, note that $|A| = N(x_L)N(x_{J \setminus L})$. Second, $x_J$ is the last subvector in $A$, since any other subvector in $A$ has either lesser weight, or equal weight and smaller rank due to the lexicographic ordering of all the subvectors with equal weights.

Definition 7: For a given vector $x \in X_q^n$ the subvector $x_L$ and subblock $L$ are called good, if $N(x_L) \leq N(x)(|L|/n)$.  

Let $Z(L) = \{L_1, \cdots, L_f\}$ be any decomposition of a subset $L$ into $f$ disjoint consecutive subsets $L_1, \cdots, L_f$ of the lengths $s_1, \cdots, s_f$.

Lemma 7: Any vector $x \in X_q^n$ contains at least one good subvector $x_{L_i}$ for any decomposition $Z(I) = \{L_1, \cdots, L_f\}$ of the full set $I = \{0, \cdots, n-1\}$.

Proof: Assume that all $f$ subblocks $x_{L_i}$ are not good (i.e., $N(x_{L_i}) > N(x)(s_i/n)$) and use the inequalities

$$N(x) \geq N(x_{L_i}) \geq N(x)(s_i/n)$$  

leading to a contradiction.

B. Design of the Lightest Subvectors of Length $s|n$

Now we assume that $s$ divides $n$, and decompose the set $I$ into $f = n/s$ consecutive subsets $L_1, \cdots, L_f$ of length $s$. Then Lemma 7 leads to the two following corollaries.

Corollary 2: If $s|n$, then any vector $x \in X_q^n$ contains at least one good subvector $x_{L_i}$ of length $s$ for some $1 \leq i \leq n/s$.

Corollary 3: If $s|n$, then any vector (path) $x$ from the list $X^s(N)$ of $N$ lightest vectors (shortest paths) contains at least one subvector $x_{L_i}$ from the list $X_{L_i}(K)$ of $K = \lceil N/s \rceil$ lightest subvectors for some $1 \leq i \leq n/s$.

This corollary shows that for $s = n/f$ we can look for $K$ shortest subpaths on each subblock $L_i$ rather than search $N$ paths on the whole block. Then at least once the projection $x_{L_i}$ of any $x \in X^s(N)$ is found for some $L_i$. This was exactly the case in Section III, where the screening of $N = 2^{n-k} = 2^{16}$ paths was substituted by that of only $2^6$ paths on either half.

The following lemma also proves that $K$ paths can be designed with a complexity of an order $K s$ for any $K$ and $s$. Basically, we use and justify the algorithm described in Section III. Let $\theta(K)$ be any algorithm, sorting $K$ real numbers with a complexity $\eta(K) \leq cK \log_2 K$.

Lemma 8: For any $K, s$, and $j$ the set of $K$ lightest subvectors on the subset $L = L(j, s)$ can be constructed in

$$C(s, K) \leq c\log_2 K \log_2 (s) = O(s K \log_2 K)$$

operations, where $r_q = 1 + 1/2 + \cdots + 1/q$.

Proof: The procedure is trivial for $K \geq q^s$, since at most $q^s$ vectors can be constructed on the subset. The weights of these vectors are calculated and ordered. Now let $K < q^s$, $p = \lceil \log_2 K \rceil$, and $(t)$ be the set of the first $t$ positions in $L(j, s)$. Then we construct the set $X_t(K)$ recurrently for all $t = p, \cdots, s$. The set $X_t(K)$ is already constructed and ordered. Suppose that $X_t(K)$ is designed. Place all $q$ possible symbols in position $t+1$ into the descendant order $x_{t+1} = \{x_{t+1}(1), \cdots, x_{t+1}(q)\}$ with respect to their weights $w_{t+1}(x_{t+1})$. According to Lemma 6, for any subpath $(x_{t+1}, x_{t+1}(i)) \in X_{t+1}(K)$ the inclusion $x_{t+1}(i) \in X_{t+1}([K/i])$ holds, since otherwise the extended path is ranked beyond $K$. Now for each symbol $x_{t+1}(i)$ construct the list $D_i = \{(x_{t+1}(i), x_{t+1}(i+1))\}$ by attaching $x_{t+1}(i)$ to any $x_{t+1}(i) \in X_t([K/i])$. Construct the combined list

$$D = \bigcup_{i=1}^q D_i$$

and order its paths with respect to their weights using sorting procedure $\theta(H)$ with $H = |D|$. Then the first $K$ vectors form the required set $X_{t+1}(K)$. For $t = s$ we obtain the list of $K$ shortest paths on the $L(j, s)$. The size $H \leq K r_q$ of the list $D$ gives the above complexity.

C. Design of the Lightest Subvectors in the General Case

Now we consider arbitrary $s$ and $n$, and start with the following generalization of Definition 7.

Definition 8: The subvector $x_{L_i}$ of a vector $x \in X_q^n$ is called well-decomposable on a subset $L$ if there exists a decomposition $Z(L) = \{L_1, \cdots, L_f\}$ such that all $f$ subvectors $x_{L_i}$ are good:

$$N(x_{L_i}) \leq N(x)(|L_i|/n), \quad i = 1, \cdots, f.$$  

(21)

We also say that this decomposition $Z(L)$ is good for vector $x$ on the subset $L$.

We prove below that any vector $x \in X_q^n$ contains well-decomposable subvector $x_{L_i}$ of arbitrary length $s$, $1 \leq s \leq n$. 

More precisely, let \( s/n = l/g \), where \( l/g \) is an irreducible fraction. Then we construct a set of decompositions of the block \( I \), which totally include about \( n(\lceil \text{log}_2 g \rceil + 1) \) blocks of length \( s \), and produce at least one well-decomposable subvector for any \( x \in X^n_q \). Below we use the following abbreviation in order to avoid bulky formulas. Let \( Z(L) \) be a decomposition of the subblock \( L(j, s) \) which starts in position \( j \) with \( d_i \) consecutive subblocks of length \( s_i \) and is then followed by \( d_{i-1} \) subblocks of length \( s_{i-1} \), \( d_{i-2} \) subblocks of length \( s_{i-2} \), and so on, until \( d_1 \) subblocks of length \( s_1 \) complete the decomposition. Then this decomposition will be denoted as \([j, d_i \times s_i + d_{i-1} \times s_{i-1} + \cdots + d_1 \times s_1]\). Note that the total number of subblocks is \( f = d_1 + \cdots + d_i \). Also where \( s_i < s_{i-1} \) for all \( i = 1, \cdots, m \) and \( s_m = \gcd(s_0, s_1) \). Now for any subset \( L(j, s) \) consider the set of decompositions
\[
Z_1 = [j, s_1],
Z_3 = [j, s_3 + d_2 \times s_2],
Z_5 = [j, s_3 + d_4 \times s_4 + d_2 \times s_2],
\ldots,
Z_{m-1} = [j, s_{m-1} + d_{m-2} \times s_{m-2} + \cdots + d_2 \times s_2],
Z_{m+1} = [j, d_m \times s_m + d_{m-2} \times s_{m-2} + \cdots + d_2 \times s_2]
\] (24)
for \( m \) and the set of decompositions
\[
Z_1 = [j, s_1],
Z_3 = [j, s_3 + d_2 \times s_2],
Z_5 = [j, s_3 + d_4 \times s_4 + d_2 \times s_2],
\ldots,
Z_m = [j, s_m + d_{m-1} \times s_{m-1} + \cdots + d_2 \times s_2]
\] (25)
for odd \( m \). Each \( Z_i \) decomposes the subblock \( L(j, s) \). Now let \( j \) run from 0 to \( n-1 \). Then either set of decompositions (24) and (25) includes \( n[(m+1)/2] \) subblocks \( L(j, s) \). Now we prove that any \( x \) has at least one good decomposition among these in terms of Definition 8.

**Lemma 9:** For any \( x \), \( n \), and \( x \in X^n_q \) there exists at least one good decomposition of length \( s \) in the set (24) for even \( n \) and in the set (25) for odd \( n \).

**Proof:** We start with the trivial decomposition \( Z_1 \) of vector \( x \), as \( j \) runs from 0 to \( n-1 \). If some subblock \( L = L(j, s_1) \) is good for this vector, then we are done. Otherwise, consider the decomposition \( I = [j, s_2 + d_1 \times s_1] \) of the full set \( I \). Then according to Lemma 7, any subblock \( L(j, s_2) \) is good, since all the subblocks of length \( s_1 \) are bad. Now consider the decomposition \( Z_3 \) for all \( j = 0, \cdots, n-1 \). If at least one subblock \( L(j, s_3) \) is good, then \( Z_3 \) is a good decomposition. Otherwise, consider the decomposition \( [j, s_4 + d_3 \times s_3] \) of any subblock \( L(j, s_2) \). Again, these arguments imply that all the subblocks \( L(j, s_4) \) are good, since any subblock \( L(j, s_2) \) is good, whereas all the subblocks \( L(j, s_2) \) are bad. We continue this procedure up to the last step. Any step \( i = 3, 5, \ldots \) can be reached only if all the subblocks \( L(j, s_2), \ldots, L(j, s_{i-1}) \) are good. Then either we get a good subblock \( L(j, s_i) \) and good decomposition \( Z_i \), or otherwise all the subblocks \( L(j, s_{i+1}) \) are good. For odd and even \( m \) only the last step is treated differently. Since all the subblocks \( L(j, s_m) \) are good for even \( m \), we construct the last decomposition \( Z_{m+1} \) as in (24). For odd \( m \) all the subblocks \( L(j, s_m) \) are good. Since \( s_{m-1} = d_m s_m \), this implies that at least one subblock \( L(j, s_m) \) is good on the \( L(0, s_{m-1}) \), and \( Z_m \) is a good decomposition.

Now consider the whole list \( X^N(N) \) of \( N \) shortest paths on the length \( n \). Below we shall again use general notation \( Z = (L_1, L_2, \cdots, L_f) \) for any decomposition \( Z \) in the sets (24) and (25). According to Lemma 9, any path \( x \in X^N(N) \) has a good decomposition \( Z \) on some subblock \( L(j, s) \). This decomposition splits the subvector \( x_L \) into good subpaths
\[
x_{L_1} \in X_{L_1}(N|L_1|/n), \cdots, x_{L_f} \in X_{L_f}(N|L_f|/n).
\]
In other words, the subvector \( x_L \) belongs to the list
\[
X_Z \triangleq X_{L_1}(N|L_1|/n) \times \cdots \times X_{L_f}(N|L_f|/n)
\] (26)
which is constructed by taking $N^{|L_{i}|}/n$ shortest subpaths on each subblock $L_i$, and then by joining these subpaths for all $i = 1, \ldots, f$. Since

$$N^{|L_{1}|}/n \times \cdots \times N^{|L_{f}|}/n = N^{s}/n$$

Lemmas 8 and 9 directly lead to the following corollaries.

**Corollary 4:** For any path $x \in X^s(N)$ there exists a subpath $x_L$ which belongs to the sublist $X_Z$ on some subblock $L$ with a decomposition $Z$ taken from the sets (24) or (25).

**Corollary 5:** For a given decomposition $Z = (L_1, L_2, \ldots, L_f)$ the sublist $X_Z$ can be designed in $O(sK \log_2 K)$ operations, where $K = N^{s}/n$. All the sublists $X_Z$ taken from the sets (24) or (25), can be constructed with complexity $O(n^2 mK \log_2 K)$.

### D. Discussion

Basically, in this section we considered arbitrary functions $N(x_L)$, which satisfy the inequality (20) on the subblocks of length $n$ or less. Now let us take the function $d(x_L) = \log_q N(x_L)$. This function is similar to the Hamming weight with the only essential difference that it satisfies the inequality $d(x) \geq d(x_L) + d(x_{\setminus L})$ instead of the explicit equality for the Hamming weight. Now we recall the discussion in Section III, where only $n$ samples $L(j, s)$ in the sliding window produced a good subvector of weight $D/s/n$. In the general case we can use $n[(m + 1)/2]$ samples in Lemma 9 in order to obtain the similar result. We note without proof that the latter bound can also be improved, so that $n$ samples suffice for this general situation. We also recall the example of Section III, where only two subblocks were used for the code of rate 1/2 instead of $n$ subblocks. This better performance can be generalized for any $s = n|/g$, so that only $g$ subblocks can be used in the decoding. However, these proofs make use of more sophisticated tools and are provided under an assumption that coordinates may be permuted.

### VIII. Algorithm

Below we consider virtually all linear codes $C(q, n, M)$ of code rate $r = \log_q M/n$, which satisfy Lemma 2 for $s = rn + [2 \log_q n]$, and all cyclic codes, which do so for $s = rn$. Let the codewords of $C(q, n, M)$ be transmitted over an arbitrary symmetric memoryless channel and let the decoding $\Psi_N$ be applied. This decoding is executed as follows. First, represent $s/n$ as the continued fraction (22). Let $\{Z\}$ denote the corresponding set of decompositions of the subblocks $L(j, s)$ defined either by (24) or by (25), depending on $m$. Then apply the following algorithm.

**Algorithm:**

1. For the given output $y = (y_0, \ldots, y_{n-1})$ define the “output-matched weights” $w_j(a)$ of any input symbol $a \in X$ in position $j = 0, \ldots, n - 1$ as in (4).

2. For any decomposition $Z \in \{Z\}$ construct the corresponding sublist $X_Z = \{x_Z\}$ (26) of subvectors of the length $s$ by linking $N^{|L_{i}|}/n$ shortest subpaths on each subblock $L_i \in Z$ for all $i = 1, \ldots, f$. Encode each subvector $x_Z \in X_Z$ onto the codeword $c(x_Z)$ of length $n$ and leave this codeword if its weight $w(c(x_Z))$ (5) is the least currently found.

3. Repeat step 2) for all decompositions $Z \in \{Z\}$ and leave the codeword $c^*$ with minimum weight $w(c^*)$ as the decoding result.

According to Lemma 9 and Corollary 5, this algorithm necessarily reconstructs all the codewords from the list $X^s(N)$ of $N$ shortest paths (lightest vectors) and therefore provides the $\Psi_N$ decoding. The complexity of designing all the lists $X_Z$ is bounded above by $O(n^2 mK \log_2 K)$, where $K = N^{s}/n$. Also, these lists include at most $n[(m + 1)/2]K$ subvectors. Each encoding requires a polynomial order of $n^3$ operations for linear codes and $n^2$ operations for cyclic codes. So the complexity of the $\Psi_N$ decoding is bounded above as $O(n^3 mK)$ for linear codes and as $O(n^3 mK)$ for cyclic codes. The number of operations with real numbers is $O(n^2 mK \log_2 K)$. Also, for general linear codes $m \leq \log_2 n$, whereas for cyclic codes $m$ is fixed, if code rate $s/n = r$ is fixed. Therefore, we have the following Lemma.

**Lemma 10:** The total complexity of the $\Psi_N$ decoding is bounded above by $O(N^{s}/n^4 \log_2 n)$ for linear codes and by $O(N^{s}/n^3)$ for cyclic codes. The number of operations with real numbers is bounded above by $O(N^{s}/n^3 \log_2 n)$ for linear codes and by $O(N^{s}/n^3)$ for cyclic codes.

**Remark:** As indicated in the above discussion, for cyclic codes of fixed rate $r = l/g$ only $g$ subblocks can be used in the decoding. Therefore, the complexity of their decoding can be upper-bounded as $O(N^{s}/n^2)$.

Now for $q$-ary codes of given rate $r$ and growing length $n$ we choose the parameter $N$, which grows slightly faster than $T = q^{n(1-r)}$ (say, $N \sim q^{n(1-r)} \log_2 n$) and arrive at the following statement.

**Theorem 4:** Virtually all $q$-ary linear codes and all cyclic codes of length $n \to \infty$ and code rate $r$, used over any memoryless symmetric channel, can be decoded with an error probability, which is equivalent to that of ML decoding, and with complexity, which is upper-bounded by the exponential order of $q^{nr(1-r)}$.

### IX. Further Improvements

**Short Codes:**

The above algorithm was also studied for short binary codes of length 32 and 64 by J. Elguren, P. Farrell, and the author. A few nonasymptotic improvements were obtained. First, computer simulation, similar to that mentioned in Section III, showed that for these lengths the number $K$ of the tested paths can be decreased below the derived bound of $2^{nr(1-r)}$. In addition, special algorithms can be designed for the lists of small size $K$, which outperform the algorithm of Lemma 8. Error-trapping technique (see [12]) also turned out to be effective for some codes. The simulation results in [18] showed that these algorithms can surpass those in [10] for lengths 32 and 64.

**Concatenated Codes:**

Below we apply Theorem 4 for binary concatenated codes from [7], which meet the GV bound. Define a set of binary concatenated codes of length $N = nL$ and rate $R = rR$.
in the following way. Let \( \{A\}[2, n, r] \) denote the set of all binary linear inner codes of length \( n \) and code rate \( r \). Let \( B[Q, L, R] \) be the RS code with \( Q = 2^{nr} \) and \( L = Q - 1 \). For any vector \( \beta = (\beta_0, \cdots, \beta_{L-1}) \) with all nonzero components construct the generalized RS code \( B_\beta \) by mapping every codeword \( (b_0, \cdots, b_{L-1}) \in B \) onto the codeword \( (b_0\beta_0, \cdots, b_{L-1}\beta_{L-1}) \). Now let \( \{V_i\}[2, N, R] \) denote the set of binary linear codes, obtained by concatenation of any inner code \( A \in \{A\} \) with any outer code \( B_\beta \). It was proved in [7] (see also [40]) that for \( n \to \infty \) virtually all concatenated codes from the set \( \{V_i\} \) meet the GV bound provided that the inner code rate is lower-bounded by

\[
r_+ = 1 + \log_2(1 - H_2^{-1}(1 - R)).
\]

(27)

A similar statement can also be obtained for multilevel concatenated codes of an arbitrary order \( m \). Namely, consider the set of \( m \) embedded linear codes \( A_i[2, n, r_i] \) of uniformly decreasing rates \( r_i = r(1-i/m) \) for all levels \( i = 0, \cdots, m-1 \). Let \( Q = 2^{nr/m} \), \( L = Q - 1 \), and \( B_i[Q, L, R_i] \) be the set of \( m \) outer generalized RS codes of some rates \( R_i \). Then these sets of inner and outer codes define the set \( \{V_m\}[2, N, R] \) of multilevel concatenations of order \( m \) and code rate

\[
R = (r/m) \sum_{i=0}^{m-1} R_i.
\]

Let

\[
R_i = (r/m) \sum_{j=i}^{m-1} R_j.
\]

It was also proved in [7] that virtually all codes from the set \( \{V_m\} \) meet the GV bound, if the restrictions

\[
(R_i - r_i)/\log_2(2^{1-r_i} - 1) \geq H_2^{-1}(1 - R)
\]

(28)

hold for all \( i = 0, \cdots, m-1 \). For \( m = 1 \) the restriction (27) follows as a special case.

**Theorem 5:** Virtually all binary concatenated codes of code rate \( R \) from the sets \( \{V_i\} \) and \( \{V_m\} \), used over an AWGN channel, can be decoded with an error probability, which is equivalent to that of ML decoding, and with the complexity, whose exponent is upper-bounded by

\[
c_{\text{con}}^{(1)} = R(1 - R/r_+)
\]

(29)

for the concatenations of the first order and by

\[
c_{\text{con}} = -(H_2^{-1}(1 - R)) \log_2(2^{1-R} - 1)
\]

(30)

for the concatenations of the growing order \( m \).

**Proof:** Consider the following algorithm for concatenated codes from the set \( \{V_i\} \). Represent the codewords and the output vector as \( n \times L \) matrices. First, all \( L \) columns of \( y \) are decoded by screening all \( Q \) codewords of the inner code \( A \). We also calculate the weight of each codeword as in (5). The complexity of this procedure is nonexponential in \( N \), being upper-bounded by an order of \( QL n \). Map each codeword of the inner code onto the corresponding symbol of the outer code. Then for the outer code we have the \( Q \)-ary outer channel \( U \) with the given weights of the symbols. Obviously, ML decoding of the binary concatenated code is equivalent to that of the outer code in this \( Q \)-ary channel.

Now note that their exist \( Q \) mappings of the inner Euclidean space \( R^n \) onto itself, which correspond to adding any codeword \( c \) in the Hamming space \( X^n \). (For a given \( e \) the corresponding mapping is obtained by reflection all the coordinate axes, where \( c \) has inputs "one.") These mappings transform linear inner code \( A \) onto itself. Also, the decoding domains of each codeword are transformed onto each other. So, the outer channel \( U \) is also a mapping channel, and suboptimal decoding can be applied. Now, any error in the outer decoding corresponds to the error in cascaded decoding and vice versa. So suboptimal decoding of concatenated code fully corresponds to that of the \( Q \)-ary outer code \( B[Q, L, R] \). For this code, Lemma 10 and Theorem 4 give the complexity of the exponential order \( Q^{L(R(1-R))} \). Substitution \( Q = 2^{nr} \) gives the complexity exponent \( R(1 - R/r) \), which achieves its minimum on (27). Similar arguments can also be applied for multilevel concatenations. Here we omit the technical details.

**Remarks:** We note that this theorem can be readily extended to \( q \)-ary codes used over any additive discrete channel or \( q \)-PSK. It can be also generalized for an arbitrary symmetric channel, but the proof is quite different and makes use of more sophisticated tools. It is also interesting to mention that the bound (30) already appeared in the technical literature. In [3] it was derived for suboptimal MD decoding of multilevel concatenations in the Hamming metric. Here we extend this bound for soft-decision decoding, and also eliminate the factor of almost exponentially growing order that appeared in complexity in [3]. This bound was also derived in [30] for soft-decision decoding of the block codes obtained from convolutional codes (the so-called tail biting codes). The algorithm of [30] allows the error probability of ML decoding to be multiplied by a nonexponential factor. Here Theorem 5 is more restrictive, since it gives the decoding error probability equivalent to that of ML decoding.

**General Linear Codes:**

We note without proof the following theorem that was already stated in [17].

**Theorem 6:** Linear \( q \)-ary codes of length \( n \to \infty \) and code rate \( R, 0 < R < 1 \), can be decoded in any memoryless symmetric channel with a decoding error probability equivalent to that of ML decoding, and decoding complexity of an exponential order \( q^{n c} \), where \( c \) is bounded above by

\[
c_5 = R(1 - R)/(1 + R)
\]

(31)

for virtually all linear codes, and by \( c_5 = (1 - R)/2 \) for all linear codes.

This theorem also leads to an essential reduction of the exponents (29) and (30), and bounds the complexity exponent of concatenated codes by 0.1 for any code rate.

**X. CONCLUDING REMARKS**

In this paper, complexity and performance of suboptimal decoding is studied for codes used over symmetric channels. In our opinion, the most important point is that the tight
bounds on error probability of suboptimal decoding can be applied to any code used over any symmetric channel. Another result is that an arbitrarily small increase in decoding error probability relative to full ML decoding is accompanied for general linear codes of rate $r$ by a substantial reduction in complexity exponent, which is upper-bounded by $n r(1 - r)$ versus the conventional bound of $n \min (r, 1 - r)$. Also, the new complexity exponent can fall off below the known lower bounds on trellis complexity. Therefore, for a wide range of code rates, the obtained reduction in decoding complexity cannot be achieved by conventional trellis decoding. The above technique of bounding decoding error probability can be used in designing various decoding algorithms. It can also be applied for numerical simulation of ML performance through the reduced search of about $q^{n r(1 - r)}$ shortest subpaths.

At first sight, this reduction in decoding complexity mostly results from the fact that we can consider the lists of only $l q^{n(1 - r)}$ shortest paths, whenever the decoding error probability $P_\psi$ of full ML decoding may be changed for the probability $P_\psi(1 + 1/l)$. However, in our opinion, the problem is deeper than it seems to be. In this regard, we recall our discussion of near-MD algorithms in Section I-B. Namely, the algorithm $\Psi_T$ with the list of size $T = [q^{n(1 - r)}]$ provides the decoding error probability bounded above by $2 P_\psi$ for any code $C(q, n, q^{n r})$. On the other hand, the above mentioned Blinovskii's result [6] on the covering radius $\rho = D + O(\ln n)$ leads to the conclusion that the algorithm $\Psi_N$ provides full MD decoding for virtually all long linear codes, where

$$N = \sum_{i=0}^{\rho} (q - 1)^i \binom{n}{i}$$

has the same exponential order as $T$. In other words, the MD decoding $\Psi_N$ being suboptimal for any code, is yet optimal for virtually all linear codes. We believe that the same fact holds in a more general situation for ML decoding in an arbitrary mapping channel, so that suboptimal decoding with a list of the exponential order $q^{n(1 - r)}$ is essentially optimal for virtually all linear codes. Namely, for a given mapping channel $(X^n, Y^n, P(y|x))$ we say that a linear code $C(q, n, r)$ is N-good if for every $y \in Y^n$ the corresponding list $X^N(q, n)$ of the $N$ most probable (closest) vectors includes at least one codeword. In other words, $C(q, n, r)$ is N-good if

$$\min_{y \in Y^n} |X^N(q, n) \cap C(q, n, r)| \geq 1.$$

Then we raise the following conjecture.

**Conjecture:** For any mapping channel $(X^n, Y^n, P(y|x))$ there exists a number $N$ of the exponential order $q^{n(1 - r)} = 1 - r$ such, that virtually all linear $(q, n, r)$ codes are N-good, if $n \to \infty$.

The next question arising with respect to this conjecture, is how the designed trellis-like algorithm if explicit indeed, can surpass the lower bounds on ML trellis complexity. We note in this regard that conventional trellis decoding imposes strong restriction on a decoding process, which is only developed in one direction from the initial vertex 0 to the final vertex $n - 1$. Such a restriction turns out to be useful for some codes with regular properties and highly decomposable structure, such as convolutional codes, or their block terminations, or concatenated codes and their subclasses (say, RM codes). These regular properties yield a substantial number of paths being merged in any vertex, and result in the corresponding complexity reductions. However, for general linear codes this merging effect is less significant. By contrast, the designed trellis-like algorithm outperforms conventional trellis decoding by splitting the full trellis of length $n$ into a few subtrellises. It can be interesting to exploit this merging effect on the decomposed subtrellises to reduce complexity further. However, note that we used rather sparse subtrellises, which do not allow for merging as often as full trellises.

We also recall our Open Problem 1. Basically, only the bounds $C_4$ and $C_5$ are to be studied in this regard. We believe that these bounds also hold for near-ML decoding of binary codes. Yet, it seems rather unlikely that they can be applied for arbitrary $q$, since the latter fact can be proved to result in nonexponential complexity for concatenated codes. Finally, note that the simulation results of Section III show that for $(q, n, r)$ codes the number of tested paths can fall below the derived bound of $q^{n r(1 - r)}$. So it is interesting to derive more precise bounds on this number, especially for the AWGN channel with a given signal-to-noise ratio. More generally, the following problem is of profound interest.

**Open Problem 3:**

What is an extra complexity cost of the soft-decision suboptimal decoding in an AWGN channel as compared to its hard-decision counterpart (that is, the minimum distance decoding in a binary symmetric channel)?

**Acknowledgment**

The author wishes to thank G. Kabatyaniskii for helpful discussion, J. Eiguren and P. Farrell for developing algorithms for short codes, and the anonymous referees for their remarks that improved the readability of the paper.

**References**


