Code Construction and Decoding Algorithms for Semi-Quantitative Group Testing with Nonuniform Thresholds

Amin Emad and Olgica Milenkovic

October 21, 2014

Abstract

We analyze a new group testing system, termed semi-quantitative group testing, which may be viewed as a concatenation of an adder channel and a discrete quantizer. Our focus is on non-uniform quantizers with arbitrary thresholds. For a given choice of parameters for the semi-quantitative group testing model, we define three new families of sequences capturing the constraints on the code design imposed by the choice of the thresholds. The sequences represent extensions and generalizations of $B_h$ and certain types of super-increasing and lexicographically ordered sequences, and they lead to code structures amenable for efficient recursive decoding. We describe the decoding methods and provide an accompanying computational complexity and performance analysis.

I. INTRODUCTION

Group testing is a family of pooling methods designed to efficiently identify relatively small subsets of subjects with some particular characteristic within a large collection of elements [2]. Rather than testing each subject individually, subgroups of subjects are tested simultaneously. The low abundance of the subjects of interest allows for determining their exact identity with a significantly reduced number of tests. Given the ubiquitous nature of the questions it addresses, this classical group testing paradigm has found many applications in communication theory, signal processing, computer science, and computational biology [3]-[14].

This work was presented in part at the IEEE 2014 International Symposium on Information Theory (ISIT’14) [1].

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL. (e-mail: emad2@illinois.edu; milenkovic@illinois.edu).
A number of extensions of classical group testing (CGT) models have also been considered in the literature [2], [15]-[22], [16], including threshold group testing (TGT) [17] and quantitative group testing (QGT) [11], [22]. In the CGT model [15], the result of a test equals 0 if the test does not include subjects of interest (i.e., “defectives”), and 1 otherwise. In the TGT model, if the number of defectives in a test is smaller than a lower threshold, the test outcome equals 0; if the number of defectives is larger than an upper threshold, the test outcome equals 1; and if the number of defectives is between the lower and upper threshold, the test result is arbitrary, either equal to 0 or 1. In QGT, the result of a test equals the exact number of defectives appearing in the test.

Motivated by applications in genotyping [20] and multiple access channel communication (MAC), the authors introduced semi-quantitative group testing (SQGT) as a unifying framework which covers CGT, TGT without gaps, QGT and many other GT models in [19], [20]. In SQGT, the output result of a test is a value from a non-binary alphabet that depends on the number of defectives through a fixed set of thresholds. Simply put, an SQGT model represents a concatenation of a QGT model and a quantizer.

In nonadaptive SQGT, each subject is assigned a unique binary or non-binary vector (codeword) of length equal to the total number of tests. It is customary to arrange the codewords as columns of a matrix, subsequently referred to as the test matrix (codebook). Each coordinate in the codeword assigned to a subject corresponds to a test, and its value reflects the “strength” of the subject in the test [19], [20]. The interpretation of the word “strength” depends on the application: for example, “strength” may correspond to the power level of a MAC user, or, it may correspond to the concentration of the genetic material of an individual. Two important families of SQGT codes, SQ-disjunct and SQ-separable, were introduced and analyzed in our companion papers [19], [20]. In the same work, constructions for uniformly quantized SQGT codes were presented, based on number-theoretic sequence selection methods.

The focus of this work is on non-adaptive SQGT with non-uniformly spaced thresholds, the most general framework in which one can study this testing scheme. For this general family of codes, we first describe new separable code constructions based on three new families of integer sequences. The sequences of interest represent extensions of well studied entities in number theory, such as non-averaging, $B_h$, and superincreasing sequences [23]. The thresholds of the quantizer determine the properties and density of the sequences, and consequently the rate of
the encoding schemes. For each encoding scheme, we also describe computationally efficient decoding algorithms which reduce to simple recursive procedures. The results are organized into four sections, with Section II introducing the model and relevant terminology, and Section III describing the notion of disjunctness and separability in the context of SQGT. Our main results are presented in Sections IV, V, and VI.

In Section IV, we introduce quantized $B_h$ sequences, and describe how to use their elements in conjunction with binary disjunct codes to construct separable SQGT codes. There, we also describe construction methods for quantized $B_h$ sequences as well as decoding methods for the resulting codes. In Section V, we introduce the notion of semi-quantitative lexicographical orders and corresponding sequences, highlight their relationship with quantized $B_h$ sequences and explain how the decoding complexity of codes based on these two families of sequences compare to each other. Section VI introduces yet one more refinement of encoding and decoding methods, and provides an overview of the main results.

II. SEMI-QUANTITATIVE GROUP TESTING: THE MODEL

Throughout the paper, we use bold-face upper-case and bold-face lower-case letters to denote matrices and vectors, respectively. Calligraphic letters are reserved for sets and sequences.

Let $\mathbb{Z}^+$ denote the set of positive integers. For a positive integer $n \in \mathbb{Z}^+$, we write $[n] := \{0, 1, \ldots, n-1\}$ and $\langle n \rangle := \{1, 2, \ldots, n\}$. With slight abuse of notation, we use $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ to denote both a set and/or a sequence consisting of $K$ positive integers. The exact meaning will be apparent from the context, and it will depend on which property of $\mathcal{A}$ is being discussed. Note that for a set of positive integers $\mathcal{A}$, one can view the natural ordering of the elements of $\mathcal{A}$ as the corresponding sequence.

Let $n$, $m$, and $d$ denote the number of test subjects, the number of tests, and the number of defectives, respectively. With each subject, we associate a unique $q$-ary vector, $q \geq 2$, of length $m$, termed a codeword. Due to the one-to-one correspondence between the codewords and test subjects, with some abuse of notation we use $\mathcal{D}$ to denote both the set of defectives and the set of codewords assigned to the defectives. Each coordinate of a codeword corresponds to a test. If $x_i \in [q]^m$ denotes the codeword of the $i^{th}$ subject, then the $k^{th}$ coordinate of $x_i$, denoted by $x_i(k)$, represents the “strength” of the $i^{th}$ subject in the $k^{th}$ test. The set of codewords is represented by the codebook $\mathbf{C} \in [q]^{m \times n}$. 
The result of SQGT tests can be represented as a vector $y \in [Q]^m$, called the vector of test results. Each test outcome depends on the number of defectives $d$ and their strengths through a quantization function $f_\eta(\cdot)$, defined as follows.

**Definition 1.** For a set of thresholds $\eta = [\eta_0 = 0, \eta_1, \ldots, \eta_Q]^T$ and a scalar $\alpha \in \mathbb{Z}^+$, we define the quantization function $f_\eta: \mathbb{Z}^+ \mapsto [Q]$ as

$$f_\eta(\alpha) = r \quad \text{if} \quad \eta_r \leq \alpha < \eta_{r+1},$$

where $r \in [Q]$. In words, the function $f_\eta(\alpha)$ returns the index of the quantization bin that contains its argument.

For a vector of positive integers $\alpha$, $f_\eta(\alpha)$ is a vector with each entry equal to the quantization of the corresponding entry of $\alpha$ according to Def. [1]. For two scalars $\alpha, \alpha' \in \mathbb{Z}^+$, and a set of thresholds $\eta$, we write $\alpha \succ_\eta \alpha'$ to indicate that $f_\eta(\alpha) > f_\eta(\alpha')$. Next, we define the syndrome of a set of codewords using $f_\eta(\cdot)$.

**Definition 2.** Let $\mathcal{X} = \{x_1, x_2, \ldots, x_s\} = \{x_j\}_1^s$ be a set of $s \geq 1$ codewords of length $m$ in a SQGT model with thresholds $\eta = [\eta_0 = 0, \eta_1, \eta_2, \ldots, \eta_Q]^T$. The syndrome of $\mathcal{X}$, denoted by $y_\mathcal{X} \in [Q]^m$, is defined as $y_\mathcal{X} = f_\eta\left(\sum_{j=1}^s x_j\right)$.

By this definition, in the absence of any errors, the vector of test results is equal to the syndrome of defectives, i.e. $y = y_D$. However, when errors occur, some entries of $y$ may differ from $y_D$. In particular, if $e$ tests are erroneous, we assume that $e$ entries of $y_D$ have changed to an arbitrary value in $[Q]$. The relationship between the syndrome of defectives and the strength of the defectives in a test is illustrated in Fig. [1].

Note that SQGT includes different group testing models as special cases. For example if $q = Q = 2$ and $\eta_1 = 1$, the SQGT model reduces to CGT. Furthermore, if $Q - 1 = d(q - 1)$ and $\forall r \in [Q]$, $\eta_r = r$, then SQGT reduces to the quantitative (adder) model (QGT) with a possibly non-binary test matrix.

1Note that this assumption corresponds to the case in which no information is available regarding the pattern of errors (i.e. worst case scenario). However, more informative assumptions regarding the error pattern can be considered to simplify the problem, e.g. errors that change the outcome of a test to the value corresponding to an adjacent bin.
\[ \sum_{j=1}^{d} x_{ij}(k) \]

\[
\begin{array}{cccccc}
0, 1, & \ldots, & \eta_1 - 1, & \eta_1, & \ldots, & \eta_2 - 1, & \ldots, & \eta_{Q-1}, & \ldots, & \eta_Q - 1.
\end{array}
\]

Fig. 1: The outcome of the \( k \)th test (in the absence of error) as a function of \( \sum_{j=1}^{d} x_{ij}(k) \).

III. SUPERIMPOSED CODES FOR SQGT

In [19] and [20], we introduced two families of SQGT codes, termed SQ-disjunct and SQ-separable codes. A \([q; Q; \eta; (l: u); e]\)-SQ-disjunct/separable code is a \( q \)-ary code for a SQGT model with thresholds \( \eta = [0, \eta_1, \eta_2, \ldots, \eta_Q]^T \), capable of uniquely identifying a number of defectives between \( l \) and \( u \), \( l \leq d \leq u \), using a \( Q \)-ary vector of test results that contains up to \( e \) errors. It was also shown in [20] that SQ-disjunct codes are special cases of SQ-separable codes that are endowed with simple decoders of computational complexity \( O(mn) \).

In what follows, we focus on constructions for SQ-separable codes with arbitrarily spaced thresholds and provide efficient decoding algorithms for the proposed codes. More precisely, we introduce three families of integer sequences that lend themselves to SQGT code design, termed “quantized \( B_h \)”,”type-s semi-quantitative lexicographically ordered sequences (SQLOs)” and “type-l semi-quantitative lexicographically ordered sequences (SQLOl)” . While SQLOs and SQLOl sequences are special cases of quantized \( B_h \) sequences, they exhibit a special nested structure that allows for computationally efficient decoding algorithms. Using the new sequences for code construction within the framework of uniform quantization, we also arrive at codes with an effectively smaller alphabet size, or equivalently, codes with higher rates given a fixed alphabet size, compared to that of [20] Construction 3].

The gist of the codebook construction method is matrix concatenation, defined as follows.

**Definition 3 (Horizontal concatenation).** Consider \( K \geq 2 \) matrices \( C_j \in \mathbb{R}^{m \times n} \), \( 1 \leq j \leq K \). The horizontal concatenation of these matrices is a matrix defined by \( C = [C_1, C_2, \ldots, C_K] \), such that for \( j \in [K] \) and \( l \in [n] \), the \( ((j - 1)n + l)^{th} \) column of \( C \) is equal to the \( l^{th} \) column of \( C_j \).

For the subsequently described code constructions, we use binary disjunct matrices for CGT as building blocks for constructing SQ-separable codes. For completeness, we start by defining
SQ-separable codes [20] and binary disjunct codes for CGT [16], [2].

**Definition 4 (SQ-separable codes).** A code is called a $[q; Q; \eta; (l; u); e]$-SQ-separable code of length $m$ and size $n$ if for any two distinct sets of codewords, $\mathcal{X}$ and $\mathcal{Z}$, satisfying $l \leq |\mathcal{X}|, |\mathcal{Z}| \leq u$, there exists a set of coordinates $\mathcal{R}$, satisfying $|\mathcal{R}| \geq 2e + 1$, such that $\forall k \in \mathcal{R}$, $y_x(k) \neq y_z(k)$.

Intuitively, SQ-separable codes impose the requirement that any collection of not fewer than $l$ and not more than $u$ items have a unique signature after quantization and in the presence of errors that change the test outcomes.

**Definition 5 (Binary $d$-disjunct codes for CGT).** A binary CGT $d$-disjunct code capable of correcting up to $e$ errors is a code of length $m$ and size $n$ with the property that for any codeword $z$ and any subset of $d$ other codewords, $\mathcal{X}$, $z \notin \mathcal{X}$, there exists a set of coordinates $\mathcal{R}$ of size at least $2e + 1$, so that $\forall k \in \mathcal{R}$ and $\forall x \in \mathcal{X}$, $z(k) = 1$ and $x(k) = 0$.

Before describing the main results of this paper, we introduce a simple code construction that provides the intuition behind the derivations of the main results.

**Theorem 1.** Consider a SQGT system with thresholds $\eta = [0, \eta_1, \eta_2, \eta_3, \ldots, \eta_Q]^T$ satisfying $\eta_Q > \eta_1 + \max\{\eta_2, \eta_3 - \eta_1\}$ and $Q \geq 4$. Fix a binary $d$-disjunct code matrix $C_b$ of dimensions $m_b \times n_b$, capable of correcting up to $e$ errors. Form a matrix $C$ of length $m = m_b$ and size $n = 2n_b$ by concatenating $C_1 = \alpha_1 C_b$ and $C_2 = \alpha_2 C_b$ horizontally, where $\alpha_1 = \eta_1$ and $\alpha_2 = \max\{\eta_2, \eta_3 - \eta_1\}$. The constructed code is a $[q; Q; \eta; (1; d); e]$-SQ-separable code with $q = \max\{\eta_2, \eta_3 - \eta_1\} + 1$.

**Proof:** Consider two distinct subsets of codewords, $\mathcal{X}_1$ and $\mathcal{X}_2$, such that $1 \leq |\mathcal{X}_1|, |\mathcal{X}_2| \leq d$. Without loss of generality, assume that $|\mathcal{X}_1| \leq |\mathcal{X}_2|$. Since the two sets are distinct, $\mathcal{X}_2 \setminus \mathcal{X}_1 \neq \emptyset$. Let $z' \in \mathcal{X}_2 \setminus \mathcal{X}_1$. By construction, $z' = \alpha z_b$ for some $\alpha \in \{\alpha_1, \alpha_2\}$ and some binary codeword $z_b$ of $C_b$. Let $z''$ be another codeword of $C$ with the same support as $z'$, obtained by multiplying $z_b$ by $\{\alpha_1, \alpha_2\} \setminus \{\alpha\}$.

If $z'' \notin \mathcal{X}_1$, then by the construction of $C$ and Def. [5] there exists a set of coordinates $\mathcal{R}$ of size at least $2e + 1$, such that $\forall k \in \mathcal{R}$, $z'(k) \geq \alpha_1 = \eta_1$ and $x(k) = 0$, $\forall x \in \mathcal{X}_1$. Since $\forall k \in \mathcal{R}$, $\sum_{x \in \mathcal{X}_2} x(k) \geq z'(k) \geq \alpha_1 \leq \eta_1$, and $\sum_{x \in \mathcal{X}_1} x(k) = 0$, it follows that $y_{x_2}(k) \geq y_{x_1}(k), (k) > y_{x_1}(k)$. 

On the other hand, if \( z'' \in \mathcal{X}_1 \cap \mathcal{X}_2 \), there exists a set of coordinates \( \mathcal{R} \) of size at least \( 2e + 1 \), such that \( \forall k \in \mathcal{R}, \quad z'(k) \in \{\alpha_1, \alpha_2\}, \quad z''(k) \in \{\alpha_1, \alpha_2\}, \) and \( x(k) = 0 \ \forall x \in \mathcal{X}_1 \{z''\} \). Since \( \forall k \in \mathcal{R}, \quad \sum_{x \in \mathcal{X}_1} x(k) \geq z'(k) + z''(k) = \alpha_1 + \alpha_2 = \max\{\eta_1 + \eta_2, \eta_3\} \geq \eta_3 \) and \( \sum_{x \in \mathcal{X}_1} x(k) \leq \alpha_2 < \eta_3 \), and since \( \eta_3 > \eta_1 + \max\{\eta_2, \eta_3 - \eta_1\} \), it follows that

\[
y_{x_2}(k) \geq y_{(z',z'')}(k) > y_{x_1}(k).
\]

If \( z'' \in \mathcal{X}_1 \setminus \mathcal{X}_2 \), we have to separately analyze two cases: if \( z' = \alpha_2 z_b \), then there exists a set of coordinates \( \mathcal{R} \) of size at least \( 2e + 1 \), such that \( \forall k \in \mathcal{R}, \quad z'(k) = \alpha_2, \quad z''(k) = \alpha_1 \), and \( x(k) = 0 \ \forall x \in \mathcal{X}_1 \{z''\} \). Since \( \forall k \in \mathcal{R}, \quad \sum_{x \in \mathcal{X}_2} x(k) \geq z'(k) = \alpha_2 \geq \eta_2 \), and \( \sum_{x \in \mathcal{X}_1} x(k) = \alpha_1 = \eta_1 < \eta_2 \), it follows that

\[
y_{x_2}(k) \geq y_{(z',z'')}(k) > y_{x_1}(k).
\]

However, for the case that \( z'' \in \mathcal{X}_1 \setminus \mathcal{X}_2 \) and \( z' = \alpha_1 z_b \), there exists a set of coordinates \( \mathcal{R} \) of size at least \( 2e + 1 \), such that \( \forall k \in \mathcal{R}, \quad z'(k) = \alpha_1, \quad z''(k) = \alpha_2 \), and \( x(k) = 0 \ \forall x \in \mathcal{X}_2 \{z'\} \). Since \( \forall k \in \mathcal{R}, \quad \sum_{x \in \mathcal{X}_2} x(k) \geq z''(k) \geq \alpha_2 \geq \eta_2 \), and \( \sum_{x \in \mathcal{X}_1} x(k) = \alpha_1 = \eta_1 < \eta_2 \), we conclude that

\[
y_{x_2}(k) < y_{(z',z'')}(k) \leq y_{x_1}(k).
\]

This completes the proof.

In [20, Construction 1], it was shown that multiplying a binary \( d \)-disjunct code of dimension \( m_b \times n_b \) by \( \eta_1 \) results in a SQ-disjunct code of the same dimension. On the other hand, Thm. 1 shows that one may increase the number of test subjects twofold, using only \( m = m_b \) tests. The increase is achieved by using a carefully chosen multiplier for the second block. More precisely, this choice of \( \alpha_2 \) satisfies two properties. First, since \( \alpha_2 \succ \eta_1 \alpha_1 \), none of the two columns of \( C \) have the same syndrome, and therefore can be uniquely distinguished. Second, the fact that \( \alpha_1 + \alpha_2 \succ \eta_2 \alpha_1 \), ensures that if we can identify a column of \( C_b \) that corresponds to at least one defective, denoted by \( x_b \), it is possible to determine if \( \{\alpha_1 x_b\} \), or \( \{\alpha_2 x_b\} \), or \( \{\alpha_1 x_b, \alpha_2 x_b\} \) are the columns of \( C \) that correspond to the defectives. These two properties, combined with the disjunctness property of \( C_b \), ensure that any collection of up to \( d \) items has a unique syndrome after quantization, even in the presence of up to \( e \) errors. This construction can be generalized to include concatenations of more than two matrices using the new families of quantized \( B_h \) sequences and the SQLO\(_s\) and SQLO\(_l\) sequences, described next.
IV. SQ-separable codes using quantized $B_h$ sequences

We start by introducing quantized $B_h$ sequences which generalize the well known $B_h$ sequences from number theory. First, we define the standard $B_h$ sequences [23].

Definition 6 ($B_h$ sequence). A finite sequence of positive integers $A = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ is a $B_h$ sequence if $\forall A_1, A_2 \subseteq A$ such that $A_1 \neq A_2$, $|A_1| = |A_2| = h$, one has $\sum_{\alpha_i \in A_1} \alpha_i \neq \sum_{\alpha_i \in A_2} \alpha_i$.

Similar to the classical $B_h$ sequences which require distinct subset sums of cardinality $h$, in quantized $B_h$ sequences we require that the quantized sums of subsets of cardinality at most $h$ be distinct. These sequences can be used to generalize Thm. [1] to construct SQ-separable codes.

Definition 7 (Quantized $B_h$ sequence). A finite sequence of positive integers $A = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ is called a quantized $B_h$ sequence with respect to $\eta$ if

1) $\alpha_K \eta > \alpha_{K-1} \eta > \cdots > \alpha_1 \eta > 0$ (i.e., all elements of $A$ lie in different quantization bins).

2) $\forall A_1, A_2 \subseteq A$ such that $A_1 \neq A_2$, $|A_1| \leq h$ and $|A_2| \leq h$, one either has $\sum_{\alpha_i \in A_1} \alpha_i \eta > \sum_{\alpha_i \in A_2} \alpha_i \eta$ or $\sum_{\alpha_i \in A_2} \alpha_i > \sum_{\alpha_i \in A_1} \alpha_i$ (the sums of elements of distinct subsets lie in different quantization bins).

Intuitively, we require that all the elements of the sequence are located in different quantization bins, none of them is in the same bin as 0, and in addition, all the sums that are formed by adding elements of subsets of cardinality at most $h$ fall into different bins. Note that when $K = 2$, setting $\alpha_1 = \eta_1$ and $\alpha_2 = \max\{\eta_2, \eta_3 - \eta_1\}$ as was done in Thm. [1] ensures that the condition in the aforementioned definition are met.

Remark 1. Note that the cardinality of a finite quantized $B_h$ sequence may be smaller than the value of $h$. For example, $A = \{\eta_1\}$ is a quantized $B_h$ sequence with respect to $\eta$, for any $h \in \mathbb{Z}^+$. However, one seeks to find the densest such sequence given an upper bound on the values of its largest element.

Quantized $B_h$ sequences can be used to construct SQ-separable codes as shown in the next theorem.

Theorem 2. Fix a binary $d$-disjunct code matrix $C_b$ of dimensions $m_b \times n_b$, capable of correcting up to $e$ errors. Let $A = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ be a quantized $B_d$ sequence with respect to $\eta$. Form a
matrix $C$ of length $m = m_b$ and size $n = K n_b$ by concatenating $K$ matrices $C_i = \alpha_i C_b$, $1 \leq i \leq K$, horizontally. The constructed code is a $[q; Q; \eta; (1 : d); e]$-SQ-separable code with $q = \alpha_K + 1$.

**Proof:** In order to show that the constructed code is $[q; Q; \eta; (1 : d); e]$-SQ-separable, we consider two distinct sets of codewords $X_1$ and $X_2$ that satisfy $1 \leq |X_1|, |X_2| \leq d$. The idea is to show that the syndrome of these two sets contain at least $2e + 1$ different entries. Without loss of generality, we assume that $X_1 \leq X_2$, since the two sets are distinct, one must have $X_2 \neq X_1$. By construction, $z_r = \alpha_r z_b$ for some binary codeword $z_b$ in $C_b$ and some $\alpha_r \in A$.

For the fixed binary codeword $z_b$, let $Z$ be the set of codewords of $C$ generated by multiplying $z_b$ with the elements of $A$. Let $Z_1 = X_1 \cap Z$ and $Z_2 = X_2 \cap Z$, be the set of codewords with the same support as $z_b$ in $X_1$ and $X_2$, respectively. Also, let $A_{Z_1} \subset A$ and $A_{Z_2} \subset A$ be the set of coefficients used to form the codewords in $Z_1$ and $Z_2$, respectively. Given that $A$ is a quantized $B_d$ sequence, we have to separately consider two different scenarios.

**Case 1:** $\sum_{\alpha_i \in A_{Z_2}} \alpha_i \eta > \sum_{\alpha_i \in A_{Z_1}} \alpha_i$.

By construction of $C$ and Def. 5, there exists a set of coordinates $R_r$ of size at least $2e + 1$, such that $\forall k \in R_r,$

\[
\begin{align*}
  z_r(k) &= \alpha_r, \\
  x(k) &= 0 \quad \forall x \in X_1 \setminus Z_1.
\end{align*}
\]

Consequently, $\forall k \in R_r$ we have the following sequence of inequalities:

\[
\begin{align*}
  y_{X_2}(k) &\geq y_{Z_2}(k) \\
  &> y_{Z_1}(k) \\
  &= y_{X_1}(k)
\end{align*}
\]

where \(1\) follows since $Z_2 \subseteq X_2$, \(2\) follows since $\sum_{\alpha_i \in A_{Z_2}} \alpha_i \eta > \sum_{\alpha_i \in A_{Z_1}} \alpha_i$, and \(3\) follows since $x(k) = 0, \forall x \in X_1 \setminus Z_1$.

**Case 2:** $\sum_{\alpha_i \in A_{Z_1}} \alpha_i \eta > \sum_{\alpha_i \in A_{Z_2}} \alpha_i$.

In this case, we cannot use the set of coordinates $R_r$, since \(2\) no longer holds. On the other hand, this case happens only if $A_{Z_1} \setminus A_{Z_2} \neq \emptyset$. Consequently, one has $Z_1 \setminus Z_2 \neq \emptyset$; let $z_s \in Z_1 \setminus Z_2$, ...
where \( z_s = \alpha_s z_b \) for some \( \alpha_s \in A_{\mathcal{Z}_1} \). Similar to case 1, by considering on \( \mathcal{X}_2 \) instead of \( \mathcal{X}_1 \), there exists a set of coordinates \( R_s \) of size at least \( 2e + 1 \), such that \( \forall k \in R_s \),

\[
\begin{align*}
z_s(k) &= \alpha_s, \\
x(k) &= 0 \quad \forall x \in \mathcal{X}_2 \setminus \mathcal{Z}_2.
\end{align*}
\]

As a result, the following inequalities hold:

\[
\begin{align*}
y_{\mathcal{X}_1}(k) &\geq y_{\mathcal{Z}_1}(k) \\
&> y_{\mathcal{Z}_2}(k) \\
&= y_{\mathcal{X}_2}(k),
\end{align*}
\]

where (4) follows since \( \mathcal{Z}_1 \subseteq \mathcal{X}_1 \), (5) follows since \( \sum_{\alpha_i \in A_{\mathcal{Z}_1}} \alpha_i > \eta \sum_{\alpha_i \in A_{\mathcal{Z}_2}} \alpha_i \), and (6) follows since \( x(k) = 0, \forall x \in \mathcal{X}_2 \setminus \mathcal{Z}_2 \). Note that even though \( |\mathcal{X}_1| \leq |\mathcal{X}_2| \), unlike for Case 1, we have \( y_{\mathcal{X}_1}(k) > y_{\mathcal{X}_2}(k) \) for all \( k \in R_s \).

A. Fundamental limits and constructions of quantized \( B_h \) sequences

Quantized \( B_h \) sequences ensure that a set of integers and their subset sums are placed into different quantization bins. As a result, for a fixed set of \( Q \) thresholds \( \eta \), the existence of quantized \( B_h \) sequences with a predetermined cardinality \( K \) depends on the thresholds. As mentioned in Remark 1, the cardinality of a quantized \( B_h \) sequence may be smaller than \( h \). For example, one can always choose \( A = \{ \eta_1 \} \) as a quantized \( B_h \) sequence with \( K = 1 \). For the case of \( K = 2 \), the sequence \( A = \{ \eta_1, \max\{ \eta_2, \eta_3 - \eta_1 \} \} \) used in Thm. 1 is a quantized \( B_h \) sequence with respect to \( \eta \) as long as \( Q \geq 4 \) and \( \eta_Q > \eta_1 + \max\{ \eta_2, \eta_3 - \eta_1 \} \). These two examples imply that for any set of thresholds, there always exists a quantized \( B_h \) sequence, which in the worst case scenario has cardinality \( K = 1 \).

We discuss next constructions of quantized \( B_h \) sequences with \( K > 2 \). From a practical perspective, and given that in most applications \( q \) cannot be too large, a greedy algorithm for finding a quantized \( B_h \) sequence is the simplest constructive approach. In the greedy approach, one starts with \( \alpha_1 = \eta_1 \); then, given the first \( i \) elements of the sequence, to find \( \alpha_{i+1} \), one increases the value of \( \alpha_i \) until the properties of the quantized \( B_h \) sequence are satisfied.
Consider a SQGT model with thresholds \( \eta = [0, \eta_1, \eta_2, \ldots, \eta_Q]^T \); \( \forall s : 1 \leq s \leq Q \), and let \( g_s = \max_{i:1 \leq i \leq s} \eta_i - \eta_{i-1} \) be the largest gap of the first \( s \) thresholds. Let \( B = \{ \beta_1 < \beta_2 < \ldots \} \) be a sequence for which all the subset sums of at most \( h \) elements are distinct. For a fixed \( s \), \( 2 \leq s \leq Q \), let \( K_s \) be a positive integer small enough to satisfy \( \eta_s > g_s \sum_{i=\max\{1,K_s-h\}}^{K_s} \beta_i \). Then all the sequences of the form \( A_s = \{g_s \beta_1, g_s \beta_2, \ldots, g_s \beta_{K_s}\} \) are quantized \( B_h \) sequences with respect to \( \eta \).

**Proof:** Fix a value of \( s : 1 \leq s \leq Q \); consider any two distinct sets \( A_1, A_2 \subseteq A_s \), \( |A_1| \leq h \) and \( A_2 \leq h \), which are obtained by multiplying the elements of \( B_1 \subseteq B \) and \( B_2 \subseteq B \) with \( g_s \), respectively. Suppose \( f_\eta \left( \sum_{\alpha_i \in A_1} \alpha_i \right) = f_\eta \left( \sum_{\alpha_i \in A_2} \alpha_i \right) \); as a result, there exists \( r, 1 \leq r \leq s \), such that \( \eta_{r-1} \leq \sum_{\alpha_i \in A_1} \alpha_i < \eta_r \) and \( \eta_{r-1} \leq \sum_{\alpha_i \in A_2} \alpha_i < \eta_r \). Consequently,

\[
\left| \sum_{\alpha_i \in A_1} \alpha_i - \sum_{\alpha_i \in A_2} \alpha_i \right| \leq \eta_r - \eta_{r-1} - 1 < g_s. \tag{7}
\]

However, since all the sums of up to \( h \) elements of \( B \) are distinct and \( |B_1| \leq h \) and \( |B_2| \leq h \),

\[
|\sum_{\beta_i \in B_1} \beta_i - \sum_{\beta_i \in B_2} \beta_i| \geq 1.
\]

Consequently,

\[
\left| \sum_{\alpha_i \in A_1} \alpha_i - \sum_{\alpha_i \in A_2} \alpha_i \right| = g_s \left| \sum_{\beta_i \in B_1} \beta_i - \sum_{\beta_i \in B_2} \beta_i \right| \geq g_s, \tag{8}
\]

which contradicts (7).

Given this theorem, one can construct quantized \( B_h \) sequences using the sequences mentioned in the theorem or the more strict subset-sum distinct sequences, for which many constructions are known in the literature \([23], [24], [25]\). One should note that for a fixed value of \( K \), the construction of quantized \( B_h \) sequences described in this theorem may not generate the densest sequence; however, this construction has the important property that it applies to any set of thresholds and only depends on a condition that can be easily verified given the thresholds.

---

\(^2\)A subset-sum distinct sequence is a sequence of positive integers such that the sum of the elements of its subsets are distinct.
**Remark 2.** All the subset-sums consisting of at most $h$ elements of a quantized $B_h$ sequence must fall into different quantization bins; since there are $Q$ such bins, the following bounds on the number of elements of a quantized $B_h$ sequence hold: Let $A$ be a finite quantized $B_h$ sequence with respect to $\eta$ such that $|A| = K$. If $K \leq h$, then $K \leq \log_2 Q$. On the other hand, if $K > h$, then $\sum_{i=0}^{h} \binom{K}{i} \leq Q$.

**Remark 3.** Let $B$ be a subset-sum distinct sequence (i.e. a sequence such that all its subsets sum up to distinct values). Assume that a positive integer $K$ satisfies the condition in Thm. 3; then, this theorem can be used to construct a quantized $B_h$ sequence $A$, $|A| = K$, using $B = \{\beta_1, \beta_2, \ldots, \beta_K\}$. There exist a large body of literature describing constructive bounds on $\beta_K$ [24], [25]. All bounds are of the form $\beta_K \leq c2^K$, where $c < 1$ is a constant that depends on the construction (e.g. $c = 0.22002$ in [25]). Given a bound of this form, one has $\alpha_K < cg2^K$, where $g$ is the largest gap for the first $K$ thresholds.

The aforementioned bound is exponential in $K$, where the base of the exponential equals 2. In Lemma 2 we will prove an upper bound on $\alpha_K$ in which the base of the exponential function is strictly smaller than 2. Although this bound applies to SQLO$_s$ sequences, given that any SQLO$_s$ sequence is also a quantized $B_h$ sequence, it can be considered an upper bound for quantized $B_h$ sequences as well. This implies that the bound in Lemma 2 is asymptotically tighter compared to aforementioned bound.

**B. A decoding algorithm for SQGT codes constructed using quantized $B_h$ sequences**

We describe next a decoding algorithm for codes constructed using Theorem 2. Let $D$ denote the set of codewords of $C$ corresponding to the defectives. Also, let $X_D$ be the set of binary codewords each corresponding to the support of at least one codeword in $D$; clearly, $|X_D| \leq |D| \leq d$. As an example, suppose that in a SQGT system $D = \{[2,0,2,2]^T, [6,0,6,6]^T, [2,0,2,0]^T\}$; in this case one has $X_D = \{[1,0,1,1]^T, [1,0,1,0]^T\}$.

The decoding procedure is performed in three steps. The idea is to use the disjunctness property of binary disjunct matrices and the property of quantized $B_h$ sequences to first recover the set $X_D$ in Step 1, and then use this set to recover $D$ in Steps 2 and 3. The steps of the decoding algorithm are listed in Algorithm 1.
Algorithm 1: Dec-QBh

Input: y ∈ [Q]^m, C_b ∈ [2]^{m×n}, η, A, e ≥ 0
Output: ŷ

Step 1: Initialize X ← ∅ and ŷ ← ∅
For i = 1, 2, . . . , n do
    If the number of coordinates j for which the i-th codeword of C_b does not satisfy
    x_i(j) ≤ y(j) is at most equal to e, set X ← X ∪ {x_i}.
End

Step 2:
Form B the ordered list of the distinct sums of elements of subsets of A with cardinality
at most d and their corresponding subsets.

Step 3:
Form u_d such that u_d(j) is the upper threshold of the quantization bin in which y(j) lies.
For i = 1, 2, . . . , |X| do
    Find β_l, the largest element of B such that the number of coordinates j for which
    β_lx_i(j) < u_d(j) is not satisfied is at most e.
    Let A_{i,l} ⊆ A be the set with the sum equal to β_l.
    Set ŷ_i ← {codewords of C of the form z = αx_i, ∀α ∈ A_{i,l}}.
End

Return ŷ = ∪_i ŷ_i

Theorem 4. Algorithm Dec-QBh is capable of identifying up to d defectives in the presence of
at most e errors in the vector of test results y.

Proof: In the first step of the algorithm, and for each codeword of the binary codebook C_b,
we count the number of coordinates for which the test result is smaller than the corresponding
entry of the codeword. In order to show that the set X recovered in Step 1 is equal to X_D, we
first show that X ⊇ X_D. Each codeword in D can be written as z_i = αx_i, 1 ≤ i ≤ |D|, for some
α ∈ A and some binary codeword x_i in X_D. We need to show that if x_i ∈ X_D, then x_i ∈ X, or
equivalently, the number of coordinates j for which

x_i(j) ≤ y(j)  \tag{9}
is not satisfied is at most $e$. All the entries of $y$ which are not erroneous are equal to the corresponding entries of the syndrome of defectives $y_p$. As a result, (9) is trivially satisfied for entries of $x_i$ that are equal to zero, since for these entries $y_p$ is equal to zero and an error can only increase the corresponding coordinate in $y$. On the other hand, since $A$ is a quantized $B_d$ sequence, its smallest element satisfies $\alpha_1 \geq \eta_1$. Consequently, a nonzero entry of $x$, results in a nonzero entry in $y$, which is a nonzero entry in $y$ unless an error occurs; since the nonzero entries of $x_i$ are equal to 1 (the smallest positive integer) and there are at most $e$ errors, condition (9) is satisfied for all except up to $e$ nonzero entries. Consequently, $\mathcal{X} \supseteq \mathcal{X}_D$.

Next, we show that if $x_i \in \mathcal{X}'$, then $x_i \in \mathcal{X}_D$, or equivalently $\mathcal{X} \subseteq \mathcal{X}_D$. Suppose this is not true and let $x \in \mathcal{X}\setminus\mathcal{X}_D$. Since $C_b$ is a binary disjunct matrix and $|\mathcal{X}_D| \leq d$, then there exists a set of coordinates $\mathcal{R}$ such that $|\mathcal{R}| \geq 2e + 1$ and $\forall j \in \mathcal{R}$ one has $x(j) = 1$ while $x_i(j) = 0$, $\forall x_i \in \mathcal{X}_D$. Consequently, $\forall j \in \mathcal{R}$, one has $y_D(j) = 0$, which implies that $y(j) = 0$ unless an error occurred. Since there are at most $e$ errors, $x(j) > y(j)$ for at least $e + 1$ coordinates, which implies that $x \notin \mathcal{X}$. This contradicts the starting assumption. Hence, $\mathcal{X} \subseteq \mathcal{X}_D$.

Now given that Step 1 recovered the set $\mathcal{X} = \mathcal{X}_D$, we only need to show that Step 3 recovers $D$ given $\mathcal{X}_D$. For each $x_i \in \mathcal{X}_D$, let $A_{i,t}$ be the “true” set of coefficients used to generate the codewords in $D$ with the same support as $x_i$. Also, let $\beta_t = \sum_{\alpha \in A_{i,t}} \alpha$ be the sum of these coefficients. Since the error-free entries of $y$ are equal to $y_p$, then for all $1 \leq j \leq m$, one has $\beta_t x_i(j) < u_D(j)$ unless an error occurred in the $j$-th coordinate. Since there are at most $e$ errors, there are at most $e$ coordinates for which this condition is not satisfied. As a result, $\beta_t \geq \beta_i$.

In order to complete the proof, we show that no value of $\beta' \in B$ such that $\beta' > \beta_t$ satisfies the condition in Step 3 and hence conclude that $\beta_t \leq \beta_i$. From the disjunctness property of $C_b$, there exists a set of coordinates $\mathcal{R}_i$ such that $|\mathcal{R}_i| \geq 2e + 1$ and $\forall j \in \mathcal{R}_i$, $x_i(j) = 1$, while all other codewords in $\mathcal{X}_D$ have the value zero at that coordinate. As a result, $\forall j \in \mathcal{R}_i$,

$$\sum_{\alpha \in A_{i,t}} \alpha x_i(j) = \sum_{\alpha \in A_{i,t}} \alpha = \beta_t.$$

Since there are at most $e$ errors in $y$, there exists a set of coordinates $\mathcal{R}_i' \subseteq \mathcal{R}_i$ with $|\mathcal{R}_i'| \geq e + 1$, such that $\forall j \in \mathcal{R}_i'$,

$$u_D(j) > \sum_{z \in D} z(j) = \beta_t.$$
Consider $\beta' \in B$ such that $\beta' > \beta_t$. Since $A$ is a quantized $B_d$ sequence, $\beta' > \eta \beta_t$ implies that $\forall j \in R'_i$, one has $\beta' \geq u_{D}(j) > \beta_t$. Given $|R'_i| \geq e + 1$, the condition in Step 3 is not satisfied for such a choice of $\beta'$ and hence $\beta_t \geq \beta_t$. As a result, Step 3 uniquely recovers $\beta_t = \beta_t$ which corresponds to the set $A_{i,t}$. Consequently, $\hat{D} = D$ in the presence of up to $e$ errors in the vector of test results, as claimed.

**Remark 4.** The computational complexity of Algorithm 1 is equal to $O(\frac{mn}{K} + 2^K(K + md))$. The computational complexity of Step 1 is $O(\frac{mn}{K})$. The second step requires $\sum_{i=1}^{\min(d,K)} (K_i(i-1)) = O(K2^K)$ summations. Finally, the computational complexity of Step 3 is $O(dm2^K)$.

Due to the exponential growth of the computational complexity of the decoding algorithm with $K$, the codes constructed using quantized $B_h$ sequences are most suitable for small values of $K$. On the other hand, for larger values of $K$, we introduce two other families of sequences that lead to codes with significantly smaller decoding complexity.

V. SQ-separable codes using SQLO$_s$ sequences

As discussed earlier, the codes constructed using quantized $B_h$ sequences have a decoding algorithm with computational complexity $O(\frac{mn}{K} + 2^K(K + md))$. Although for small values of $K$ the dominant term is $\frac{mn}{K}$, for large values of $K$ the exponential growth of the complexity with $K$ is problematic. In this section we introduce the notion of SQLO$_s$ sequences and use them to construct SQ-separable codes with a decoding algorithm that has computational complexity linear in $K$.

**Definition 8 (SQLO$_s(\eta, h)$ sequences).** Given a set of thresholds $\eta$, a sequence of positive integers $A = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ is termed a SQLO$_s(\eta, h)$ sequence if

1) $\alpha_K \geq \eta \alpha_{K-1} \geq \eta \ldots \geq \alpha_2 \geq \eta \alpha_1 > \eta > 0$ (i.e., all elements of $A$ lie in different quantization bins).

2) For any two distinct nonempty nested subsets $A_1 \subset A_2 \subseteq A$ such that $|A_1| \leq h$ and $|A_2| \leq h$, one has $\sum_{\alpha_i \in A_2} \alpha_i \geq \eta \sum_{\alpha_i \in A_1} \alpha_i$ (i.e., the sums of elements of nested subsets fall into different quantization bins).

3) For any two distinct nonempty subsets $A_1, A_2 \subseteq A$ that are not nested and $|A_1| \leq h$ and $|A_2| \leq h$, one has $\sum_{\alpha_i \in A_2} \alpha_i \geq \eta \sum_{\alpha_i \in A_1} \alpha_i$ whenever $\exists \alpha \in A_2 \setminus A_1$ such that $\alpha > \eta \alpha'$, $\forall \alpha' \in A_1 \setminus A_2$ (i.e., two subsets that are not nested are ordered based on their largest distinct element).
The properties above induce a partial order on the subsets of a SQLOₜ sequence.

The SQLOₜ properties for \( K = 2 \) and \( h \geq 2 \) simply translates to \( \alpha_2 + \alpha_1 \geq \eta \alpha_2 \geq \eta \alpha_1 \geq \eta 0 \), while for \( K = 3 \) and \( h \geq 3 \) it translates into \( \alpha_3 + \alpha_2 + \alpha_1 \geq \eta \alpha_3 + \alpha_2 \geq \eta \alpha_3 + \alpha_1 \geq \eta \alpha_2 + \alpha_1 \geq \eta \alpha_2 \geq \eta \alpha_1 \geq \eta 0 \). As an example, it can be easily verified that \( \mathcal{A} = \{3, 6, 12\} \) is a SQLOₜ sequence with respect to the thresholds \( \eta = [0, 3, 6, \ldots, 24]^T \), since \( f_\eta(12 + 6 + 3) = 7 > f_\eta(12 + 6) = 6 > f_\eta(12 + 3) = 5 > f_\eta(12) = 4 > f_\eta(6 + 3) = 3 > f_\eta(6) = 2 > f_\eta(3) = 1 > 0 \).

SQLOₜ sequences obey a more stringent set of constraints compared to the quantized \( B_h \) sequences. As a result, one is able to use these constraints to reduce the computational complexity of the decoder. In the next proposition, we show that in fact any SQLOₜ sequence is also a quantized \( B_h \) sequence, but the converse is not necessarily true.

**Proposition 1.** A sequence of \( K \) positive integers \( \mathcal{A} \) is a SQLOₜ(\( \eta, h \)) sequence if and only if both of the following properties are satisfied:

1) \( \mathcal{A} \) is a quantized \( B_h \) sequence.

2) \( \forall i : 1 \leq i \leq K \) and \( \forall \mathcal{A}' \subseteq \{\alpha_1, \alpha_2, \ldots, \alpha_{i-1}\} \) such that \( |\mathcal{A}'| \leq h \), one has \( \alpha_i \geq \eta \sum_{j \in \mathcal{A}'} \alpha_j \).

**Proof:** First, we show that if \( \mathcal{A} \) is a SQLOₜ(\( \eta, h \)) sequence, it satisfies properties 1 and 2. It is easy to see that since \( \mathcal{A} \) is a SQLOₜ(\( \eta, h \)) sequence, it satisfies the first property of quantized \( B_h \) sequences, i.e. \( \alpha_K \geq \eta \alpha_{K-1} \geq \eta \cdots \geq \eta \alpha_1 \geq \eta 0 \).

Let \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \) be two arbitrary nonempty distinct subsets of \( \mathcal{A} \) such that \( |\mathcal{A}_1| \leq h \) and \( |\mathcal{A}_2| \leq h \). We need to show that \( \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \geq \eta \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i \) or \( \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i \geq \eta \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \). If these two subsets are nested, i.e. if \( \mathcal{A}_1 \subset \mathcal{A}_2 \) or \( \mathcal{A}_2 \subset \mathcal{A}_1 \), from the second property of a SQLOₜ sequence, it follows that \( \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i \geq \eta \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \) or \( \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \geq \eta \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i \), respectively. Otherwise, the third property of a SQLOₜ(\( \eta, h \)) sequence ensures that \( \mathcal{A} \) is a quantized \( B_h \) sequence. On the other hand, from the third property of a SQLOₜ(\( \eta, h \)) sequence, one can directly conclude that the second property of the proposition holds.

Now we show that if \( \mathcal{A} \) satisfies the two properties stated in the proposition, then it is a SQLOₜ(\( \eta, h \)) sequence. Since \( \mathcal{A} \) is a quantized \( B_h \) sequence, the first property of a SQLOₜ(\( \eta, h \)) sequence is automatically satisfied.

Next, consider two distinct nonempty nested subsets \( \mathcal{A}_1 \subset \mathcal{A}_2 \subseteq \mathcal{A} \) such that \( |\mathcal{A}_1| \leq h \) and \( |\mathcal{A}_2| \leq h \). Since \( \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i \) and \( \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \) fall into different quantization bins, due to the second property of a quantized \( B_h \) sequence, and since \( \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i > \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \), one has \( \sum_{\alpha_i \in \mathcal{A}_2} \alpha_i > \eta \sum_{\alpha_i \in \mathcal{A}_1} \alpha_i \).
Now consider two distinct nonempty subsets $A_1, A_2 \subseteq A$ that are not nested, such that $|A_1| \leq h$ and $|A_2| \leq h$. Assume that $\exists \alpha \in A_2 \setminus A_1$ such that $\alpha >_\eta \alpha', \forall \alpha' \in A_1 \setminus A_2$. In this case, it holds that

$$\sum_{\alpha_i \in A_2} \alpha_i >_\eta \sum_{\alpha_i \in A_1} \alpha_i,$$

where the last inequality follows from the second property of the proposition. This completes the proof of the proposition.

As a result of the first condition in Proposition\(^1\) one can directly use a SQLO\(_s(\eta, h)\) sequence instead of a quantized $B_h$ sequence to construct SQ-separable codes, as formally stated in the next theorem. In addition, the second property in Proposition\(^1\) allows us to reduce the computational complexity of the decoder significantly. This is a consequence of the fact that superincreasing sequences\(^3\) are knapsack-solvable in linear time [26]. In other words, given an integer and a finite superincreasing sequence, it is possible to determine in linear time whether the integer can be expressed as a sum of distinct elements of the sequence, and if so to identify these elements [26].

**Theorem 5.** Fix a binary $d$-disjunct code matrix $C_b$ of dimensions $m_b \times n_b$, capable of correcting up to $e$ errors. Let $A = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ be a SQLO\(_s(\eta, d)\) sequence. Form a matrix $C$ of length $m = m_b$ and size $n = Kn_b$ by concatenating $K$ matrices $C_i = \alpha_i C_b$, $1 \leq i \leq K$, horizontally. The constructed code is a $[q; Q; \eta; (1:d); e]$-SQ-separable code with $q = \alpha_K + 1$.

**Proof:** The proof directly follows from Proposition\(^1\) and Thm.\(^2\). Since any SQLO\(_s(\eta, d)\) sequence is a quantized $B_d$ sequence, Thm.\(^2\) implies that the code $C$ is a $[q; Q; \eta; (1:d); e]$-SQ-separable code with $q = \alpha_K + 1$.\(\blacksquare\)

A. Fundamental limits and constructions of SQLO\(_s\) sequences

We discuss next construction methods and fundamental density limits for SQLO\(_s\) sequences. Given a set of thresholds, a simple greedy algorithm can be used to find a SQLO\(_s\) sequence by checking the properties in Def.\(^4\). For example, assuming that $\eta = [0, 2, 5, 6, 10, 13, 15, 16, 18, 21]^T$ and that $h \geq K = 3$, the greedy algorithm produces $A = \{2, 5, 11\}$. Alternatively, one can use the following theorem to construct SQLO\(_s\) sequences using superincreasing sequences.

\(^3\)A superincreasing sequence is a sequence of positive integers such that each element of the sequence is at least as large as the sum of all the elements preceding it.
**Definition 9.** A sequence of positive integers $B = \{\beta_1, \beta_2, \ldots\}$ is called $h$-superincreasing if $\forall j > 1, \beta_j > \sum_{i=\max\{1,j-h\}}^{j-1} \beta_i$.

**Theorem 6.** Consider a SQGT system with thresholds $\eta = [0, \eta_1, \eta_2, \ldots, \eta_Q]^T; \forall s : 1 \leq s \leq Q$, let $g_s = \max_{1 \leq i \leq s} \eta_i - \eta_{i-1}$ be the largest gap of the first $s$ thresholds. Let $B = \{\beta_1 < \beta_2 < \ldots\}$ be a $h$-superincreasing sequence. For a fixed $s$, $2 \leq s \leq Q$, let $K_s$ be a positive integer small enough to satisfy $\eta_s > g_s \sum_{i=\max\{1,K_s-h\}}^{K_s} \beta_i$. Then all the sequences of the form $A_s = \{g_s \beta_1, g_s \beta_2, \ldots, g_s \beta_{K_s}\}$ are SQLO$_s(\eta, h)$ sequences.

**Proof:** Let $A_s$ be a fixed sequence satisfying the conditions of the theorem. First, we show that $A_s$ is a quantized $B_h$ sequence. Fix a value of $s : 1 \leq s \leq Q$. Consider any two distinct sets $A_1, A_2 \subseteq A_s$, $|A_1| \leq h$ and $|A_2| \leq h$, which are obtained by multiplying the elements of $B_1 \subseteq B$ and $B_2 \subseteq B$ with $g_s$, respectively. Suppose that $f_\eta \left( \sum_{i \in A_1} \alpha_i \right) = f_\eta \left( \sum_{i \in A_2} \alpha_i \right)$; as a result, there exists an integer $r$, $1 \leq r \leq s$, such that $\eta_r - 1 \leq \sum_{i \in A_1} \alpha_i < \eta_r$ and $\eta_{r-1} \leq \sum_{i \in A_2} \alpha_i < \eta_r$. Consequently,

$$\left| \sum_{i \in A_1} \alpha_i - \sum_{i \in A_2} \alpha_i \right| \leq \eta_r - \eta_{r-1} - 1 < g_s. \quad (10)$$

Since $B_1 \neq B_2$, the set $(B_1 \cup B_2) \setminus (B_1 \cap B_2)$ is nonempty. Let $\beta_l$ be the largest element of this set, and without loss of generality assume that $\beta_l \in B_1$. Since $B$ is a $h$-superincreasing sequence and $|B_1| \leq h$ and $|B_2| \leq h$, one has $\beta_l > \sum_{i \in B_2} \beta_i$. This implies that $\sum_{i \in B_1} \beta_i > \sum_{i \in B_2} \beta_i$, or equivalently, that $\left| \sum_{i \in B_1} \beta_i - \sum_{i \in B_2} \beta_i \right| \geq 1$. Consequently,

$$\left| \sum_{i \in A_1} \alpha_i - \sum_{i \in A_2} \alpha_i \right| = g_s \left| \sum_{i \in B_1} \beta_i - \sum_{i \in B_2} \beta_i \right| \geq g_s, \quad (11)$$

which contradicts (10); hence $A_s$ is a quantized $B_h$ sequence.

In order to complete the proof, we need to show that $\forall \alpha_j \in A_s$, $1 \leq j \leq K_s$, and $\forall A_1 \in \{\alpha_1, \alpha_2, \ldots, \alpha_{j-1}\} \subseteq A_s$ such that $|A_1| \leq h$, one has $\alpha_j > \eta \sum_{i \in A_1} \alpha_i$. Suppose this were not true and that $f_\eta(\alpha_j) \leq f_\eta \left( \sum_{i \in A_1} \alpha_i \right) = r, 1 \leq r \leq s$. As a result,

$$\alpha_j - \sum_{i \in A_1} \alpha_i < g_s. \quad (12)$$

Let $\beta_j = \frac{\alpha_j}{g_s}$ and let $B_1 \subseteq B$ be the set which was used to construct $A_1$. Since $B$ is a $h$-
superincreasing sequence and \(|B_1| \leq h\), one has

\[ \beta_j > \sum_{\beta_i \in B_1} \beta_i \Rightarrow \beta_j - \sum_{\beta_i \in B_1} \beta_i \geq 1. \]  

(13)

By multiplying both sides of (13) by \(g_s\), one has \(\alpha_j - \sum_{\alpha_i \in A_1} \alpha_i \geq g_s\) which contradicts (12). As a result, \(A_s\) is a SQLO\(_s\)(\(\eta, h\)) sequence.

Given this result, one can construct SQLO\(_s\)(\(\eta, h\)) sequences using \(h\)-superincreasing sequences. For example, the sequence \(B = \{1, 2, 2^2, 2^3, \ldots\}\) is a superincreasing sequence, hence an \(h\)-superincreasing sequence for any value of \(h\), and can be used to construct SQLO\(_s\)(\(\eta, h\)) sequences. Given this sequence, one obtains a SQLO\(_s\)(\(\eta, h\)) sequence such that \(\alpha_K = O_{g_s}\gamma^K\), where \(g_s\) is the largest gap for the first \(K\) thresholds, and \(\gamma < 2\).

Proof: We construct a sequence \(B\) as follows. First, \(\forall 1 \leq i \leq h\), we set \(\beta_i = 2^{i-1}\). Then, for \(i > h\), we let \(\beta_i = \beta_{i-1} + \beta_{i-2} + \cdots + \beta_{i-h} + 1\). Clearly, this sequence is a \(h\)-superincreasing sequence. The characteristic equation of this recurrence is of the form \(f(x) = x^h - x^{h-1} - \cdots - x - 1 = 0\), which satisfies the condition of Lemma [1]. In addition, the greatest common divisor of the indices of the positive coefficients is 1, since all these coefficients are equal to 1. Consequently, Lemma [1] implies that this equation has a unique real positive root, \(\gamma\), and that the absolute values of all
the other roots are strictly smaller than $\gamma$. Consequently, $\beta_K = O(\gamma^K)$. Simplifying this equation by multiplying both sides by $(x - 1)$, the equation becomes $x^{h+1} - 2x^h + 1 = 0$. Consequently, one has $\alpha_K = O_g(\gamma^K)$, where $\gamma$ is the largest positive real root of $g(x)$.

Next, we show that $\gamma < 2$. Evaluating $g(x) = x^{h+1} - 2x^h + 1$ on the real axis reveals that this function has two local optima at $x = 0$ and $x = \frac{2h}{h+1}$, and is monotonically increasing for $x > \frac{2h}{h+1}$. On the other hand, $g(2) > 0$; in addition, for all $h \geq 1$, one has $2 > \frac{2h}{h+1}$; consequently, $\forall x > 2, f(x) > f(2) > 0$. As a result, the largest positive real solution to $g(x) = 0$ is strictly smaller than 2 for any finite value of $h$, i.e. $\gamma < 2$.

\[\blacksquare\]

B. A decoding algorithm for SQGT codes constructed using SQLO$_s$ sequences

We next describe the Dec-SQLO$_s$ algorithm, the decoding procedure for codes based on SQLO$_s$ sequences. This algorithm comprises of two steps. The first step is identical to the first step of Algorithm 1. However, Steps 2 and 3 in Algorithm 1 are replaced by a single step which has a significantly lower computational complexity than steps 2 and 3. The steps of Dec-SQLO$_s$ are listed in Algorithm 2. The first step identifies the set $X_D$. Given this set, Step 2 identifies the set of defectives $D$. In order to show that the second step can identify up to $d$ defectives in the presence of up to $e$ errors, we state the following lemma and proposition which we find useful for our subsequent proofs.

**Lemma 3.** Consider a SQ-separable code constructed using Thm. 5 and let $y$ be the vector of test results with at most $e$ erroneous entries. Fix any binary codeword $x_i \in X_D$, and let $S_i$ be the set of nonzero coordinates of $x_i$. Also, let $S_i' \subseteq S_i$, with $|S_i'| = 2e + 1$, be the set of coordinates such that for any fixed $k \in S_i'$, one has $y(k) \leq y(j), \forall j \in S_i \setminus S_i'$. Then, there exists a set $S_i'' \subseteq S_i$ such that $|S_i''| \geq e + 1$, and $\forall j \in S_i''$, one has $y(j) = y_o(j) = f_n(\sum_{\alpha \in A_{i,t}} \alpha)$; in this equation, $A_{i,t} \subseteq A$ denotes the set of coefficients corresponding to the defective codewords with the same support as $x_i$.

**Proof:** Let $R_i$ be the maximal set of coordinates such that $\forall j \in R_i, x_i(j) = 1$ and $x(j) = 0$ for all $x \in X_D \setminus \{x_i\}$. Since $x_i$ is a codeword of $C_b$ and since $|X_D| \leq d$, the disjunctness property

\[\blacksquare\]

---

\[4\] As an example, assume that $x_i \in X_D$ and let $\{\alpha_{j_1}x_i, \alpha_{j_2}x_i, \alpha_{j_3}x_i\} \in D$ be the only codewords in $C$ with the same support as $x_i$ in $D$. In this case, $A_{i,t} = \{\alpha_{j_1}, \alpha_{j_2}, \alpha_{j_3}\}$.
Algorithm 2: Dec-SQLO

Input: $y \in [Q]^m$, $C_{b} \in [2]^{m \times n}$, $\eta$, $A$, $e \geq 0$
Output: $\hat{D}$

Step 1: Initialize $\mathcal{X} \leftarrow \emptyset$ and $\hat{D} \leftarrow \emptyset$
For $i = 1, 2, \ldots, n$ do
    $x_i$ ← the $i$-th codeword of $C_{b}$
    $N_i$ ← number of coordinates $j$ for which $x_i(j) > y(j)$
    If $N_i \leq e$ then
        Set $\mathcal{X} \leftarrow \mathcal{X} \cup \{x_i\}$
End
Step 2: For $i = 1, 2, \ldots, |\mathcal{X}|$ do
    Set $S_i$ ← \{the set of nonzero coordinates of $x_i$\}
    Set $S'_i$ ← \{subset of $S_i$ with $|S'_i| = 2e + 1$ s.t. $\forall k \in S'_i$ and $\forall j \in S_i \setminus S'_i$, one has $y(k) \leq y(j)$\}
    Initialize the multiset $B' \leftarrow \emptyset$
    For $j = 1, 2, \ldots, |S'_i|$ do
        $\eta_u$ ← the upper threshold of the quantization bin of $y(j)$
        $\eta_l$ ← the lower threshold of the quantization bin of $y(j)$
        $\beta$ ← the integer $\eta_l \leq \beta < \eta_u$ that can be written as the sum of up to $d$ elements of $A$
        (use Proposition $[2]$)
        Update the multiset $B' \leftarrow B' \cup \{\beta\}$
    End
    Set $\hat{\beta}_i \leftarrow$ the element of $B'$ with at least $e + 1$ repetitions
    Set $\hat{A}_{i,t} \leftarrow$ \{the unique subset of $A$ with the sum equal to $\hat{\beta}_i$\}
    Set $\hat{D}_i \leftarrow$ \{codewords of $C$ of the form $z = \alpha x_i$, $\forall \alpha \in \hat{A}_{i,t}$\}
End
Return $\hat{D} = \bigcup_i \hat{D}_i$

implies that such a set exists and that $|R_i| \geq 2e + 1$; clearly, $R_i \subseteq S_i$. Let $A_{i,t}$ be the set of coefficients used to generate the codewords in $\hat{D}$ with the same support as $x_i$. For all $k \in R_i$, one has $\sum_{z \in \hat{D}} z(k) = $ $\sum_{\alpha \in A_{i,t}} \alpha$, and $\forall j \in S_i \setminus R_i$, one has $\sum_{z \in \hat{D}} z(j) > $ $\sum_{\alpha \in A_{i,t}} \alpha$. Note that the strict inequality follows since $R_i$ is a maximal set. Since all the sums of up to $d$ elements of $A$
fall into different quantization bins, for any \( k \in \mathcal{R}_i \) and for any \( j \in \mathcal{S}_i \setminus \mathcal{R}_i \), one has

\[
 f_\eta \left( \sum_{z \in D} z(k) \right) < f_\eta \left( \sum_{z \in D} z(j) \right).
\]

As a result, if there were no errors in \( y \), one would have \( S'_i \subseteq \mathcal{R}_i \). Each erroneous entry of \( y \) removes at most one coordinate of \( \mathcal{R}_i \) from \( S'_i \). Since there are at most \( e \) errors and \( |S'_i| = 2e + 1 \), there exists a set of coordinates \( S''_i \subseteq S'_i \cap \mathcal{R}_i \) with cardinality at least \( e + 1 \) for which the corresponding entries of \( y \) are error-free. As a result, \( \forall j \in S''_i \) one has \( y(j) = y_D(j) = f_\eta \left( \sum_{\alpha \in \mathcal{A}_{i,t}} \alpha \right) \).

**Proposition 2.** Given a SQLO\(_s\)(\( \eta, d \)) sequence \( \mathcal{A} \) and a fixed integer \( \beta \), one can identify whether \( \beta \) can be written as a sum of up to \( d \) elements of \( \mathcal{A} \) with an algorithm of computational complexity \( O(K) \), where \( K = |\mathcal{A}| \). Given that the answer to this question is positive, one can identify the elements of \( \mathcal{A} \) which sum up to \( \beta \) with computational complexity \( O(K) \).

**Proof:** This problem is known as the knapsack-solvability problem [26]. From the second property of Proposition 1, \( \forall i : 1 \leq i \leq K \) and \( \forall \mathcal{A}_1 \subseteq \{ \alpha_1, \alpha_2, \ldots, \alpha_{i-1} \} \) such that \( |\mathcal{A}_1| \leq d \), one has \( \alpha_i \gtrsim \eta \sum_{\alpha_j \in \mathcal{A}_1} \alpha_j \), which also implies that \( \alpha_i \gtrsim \sum_{\alpha_j \in \mathcal{A}_1} \alpha_j \).

To find the answer to the query with linear computational complexity, we perform a standard knapsack recursion [26]. First, we initialize the procedure by setting \( \beta' \leftarrow \beta \) and \( \hat{\mathcal{A}}_\beta \leftarrow \emptyset \). Then, in the \( i \)-th iteration, we compare the value of \( \beta' \) with the \( i \)-th largest element of \( \mathcal{A} \), \( \alpha_{K-i+1} \). If \( \beta' \gtrsim \alpha_{K-i+1} \), then we update \( \beta' \leftarrow \beta' - \alpha_{K-i+1} \) and \( \hat{\mathcal{A}}_\beta \leftarrow \hat{\mathcal{A}}_\beta \cup \{ \alpha_{K-i+1} \} \); otherwise, we go to the next iteration. The procedure stops with a negative answer to the first query if \( |\hat{\mathcal{A}}_\beta| > d \) or if \( \beta' > 0 \) and no element in \( \mathcal{A} \) is left that is smaller than or equal to \( \beta' \). Otherwise, the procedure stops when \( \beta' = 0 \) with a positive answer to the first query, and \( \hat{\mathcal{A}}_\beta \) corresponds to the elements of \( \mathcal{A} \) that sum up to \( \beta \). Note that this procedure is based on the superincreasing property of a SQLO\(_s\) sequence, which implies that the largest element of \( \mathcal{A} \) that is smaller than \( \beta' \) must be present in the sum.

The previous proposition and lemma provide the core of the second step of Algorithm 2. The idea is that for each \( x_i \in \mathcal{X}_D \), we use Lemma 3 to find \( S'_i \). The majority of elements \( y(j), j \in S'_i \), correctly correspond to the bin in which \( \beta_i = \sum_{\alpha \in \mathcal{A}_{i,t}} \alpha \) is located. Each correctly identified bin
contains a finite number of integers, one of which is the true value of $\beta_t$. As a result, by testing each such integer, we can determine whether it can be written as the sum of up to $d$ elements of $\mathcal{A}$ or not using the algorithm in Proposition 2. The integer for which the answer to this query is positive is equal to $\beta_t$, which can then be used to identify the elements of $\mathcal{A}_{i,t}$.

**Theorem 7.** The Dec-SQLO$_s$ algorithm is capable of identifying up to $d$ defectives in the presence of at most $e$ errors in the syndrome of defectives.

**Proof:** Since the first step of this algorithm is identical to the first step of the Dec-QBh algorithm, it follows that $X = X_D$. Therefore, we only need to show that Step 2 recovers $D$ given $X_D$.

Fix a binary vector $x_i \in X_D$. Fix a coordinate $j \in S'_i$, and let $\eta_l$ and $\eta_u$ be the lower and upper thresholds of the quantization bin corresponding to $y(j)$. Since all the sums of up to $d$ elements of $\mathcal{A}$ fall into different bins, there exists exactly one subset sum $\beta$ in $[\eta_l, \eta_u)$ that corresponds to the sum of up to $d$ elements of $\mathcal{A}$. As a result, one can test all the $(\eta_u - \eta_l)$ integers in this bin using Proposition 2 to find the unique value of $\beta$ that can be written as sum of up to $d$ elements of $\mathcal{A}$. On the other hand, as was shown in Lemma 3, there exists a set $S''_i \subseteq S'_i$ such that $|S''_i| \geq e + 1$, and consequently $\forall j \in S''_i$ one has $y(j) = f_\eta(\sum_{\alpha \in A_{i,t}} \alpha) = f_\eta(\beta_t)$. As a result, the element $\beta$ in the multiset $B'$ with multiplicity at least $e + 1$ corresponds to $\beta_t$, or in other words $\hat{\beta}_t = \beta_t$. This implies that $\hat{\mathcal{A}}_{i,t} = \mathcal{A}_{i,t}$, and consequently, $\hat{D} = D$. 

**Remark 5.** The computational complexity of the Dec-SQLO$_s$ algorithm is equal to $O(\frac{mn}{K} + dm \log m + \deg_{\text{max}} K)$, where $g_{\text{max}} = \max_{i=1,2,...,Q} (\eta_i - \eta_{i-1})$ is the largest gap between the consecutive thresholds. The computational complexity of Step 1 is $O(\frac{mn}{K})$. On the other hand, sorting the elements of $S_i$ to find $S'_i$ requires $O(dm \log m)$ computations. One can identify the elements of $\mathcal{A}$ that sum up to a fixed integer in linear time., i.e. using $O(K)$ computational steps. As a result, the algorithm for finding $\beta$ in each iteration has complexity $O(\epsilon \deg_{\text{max}} K)$. Hence, finding $\hat{\mathcal{A}}_{i,t}$ requires $O(\deg_{\text{max}} K)$ computational steps.

**VI. SQ-separable codes using SQLO$_t$ sequences**

The SQLO$_s$ sequences introduced in the previous section resolves the problem of exponential growth of decoding computational complexity with respect to $K$. However, due to the superincreasing property of these sequences (the second property in Prop. 1) the multipliers $\alpha_K$ tend
to grow rapidly as a function of \( K \). In order to overcome this issue while preserving efficient decoding, we introduce a new family of integer sequences, termed SQLO\(_l\) sequences.

**Definition 10 (SQLO\(_l\)(\( \eta, h \)) sequences).** Given a set of thresholds \( \eta \), a sequence of positive integers \( A = \{ \alpha_1, \alpha_2, \ldots, \alpha_K \} \) is a SQLO\(_l\)(\( \eta, h \)) sequence if

1) \( \alpha_K > \eta \alpha_{K-1} > \eta \ldots \alpha_2 > \eta \alpha_1 > \eta 0 \) (i.e., all elements of \( A \) lie in different quantization bins).

2) For any two subsets \( A_1 \subseteq A \) and \( A_2 \subseteq A \) such that \( |A_1| < |A_2| \leq h \), one has \( \sum_{\alpha_i \in A_2} \alpha_i > \eta \sum_{\alpha_i \in A_1} \alpha_i \) (i.e., subsets of different cardinality are ordered based on the number of their members).

3) For any two distinct subsets \( A_1 = \{ \alpha'_1, \alpha'_2, \ldots, \alpha'_s \} \) and \( A_2 = \{ \alpha''_1, \alpha''_2, \ldots, \alpha''_t \} \) with elements listed in an increasing order such that \( |A_1| = |A_2| = s \leq h \), one has \( \sum_{\alpha''_i \in A_2} \alpha''_i > \eta \sum_{\alpha'_i \in A_1} \alpha'_i \) if there exists \( r : 1 \leq r \leq s \) such that \( \forall i : 1 \leq i < r, \alpha'_i = \alpha''_i \) and \( \alpha''_r > \eta \alpha'_r \) (i.e., two subsets with the same cardinality are lexicographically ordered).

As an example, consider the set of thresholds \( \eta = [0, 2, 5, 6, 10, 11, 15, 18]^T \) and let \( h = 2 \) and \( K = 3 \). The sequence \( A_1 = \{2, 5, 10\} \) is a SQLO\(_s\)(\( \eta, 2 \)) sequence that has the smallest value for \( \alpha_3 \), i.e. \( \alpha_3 = 10 \). On the other hand, the sequence \( A_2 = \{4, 5, 6\} \) is a SQLO\(_l\)(\( \eta, 2 \)) sequence that has the smallest positive value for \( \alpha_3 \), i.e. \( \alpha_3 = 6 \). This simple example illustrates how SQLO\(_l\) properties may lead to denser sequences compared to SQLO\(_s\) sequences.

The SQLO\(_l\) properties impose a partial order on the subsets of the sequence. For example, if \( K = 3 \) and \( h \geq K \), these properties translate into \( \alpha_3 + \alpha_2 + \alpha_1 > \eta \alpha_3 + \alpha_2 > \eta \alpha_3 + \alpha_1 > \eta \alpha_2 + \alpha_1 > \eta \alpha_3 > \eta \alpha_2 > \eta \alpha_1 > \eta 0 \). Similarly to the case of SQLO\(_s\) sequences, it is not difficult to see that any SQLO\(_l\)(\( \eta, h \)) sequence is also a quantized \( B_h \) sequence; however, the converse is not necessarily true. As a result, the following theorem holds.

**Theorem 8.** Fix a binary \( d \)-disjunct code matrix \( C_b \) of dimensions \( m_b \times n_b \), capable of correcting up to \( e \) errors. Let \( A = \{ \alpha_1, \alpha_2, \ldots, \alpha_K \} \) be a SQLO\(_l\)(\( \eta, d \)) sequence. Form a matrix \( C \) of length \( m = m_b \) and size \( n = K n_b \) by concatenating \( K \) matrices \( C_i = \alpha_i C_b \), \( 1 \leq i \leq K \) horizontally. The constructed code is a \( [q; Q; \eta; (1:d); e]-SQ\)-separable code with \( q = \alpha_K + 1 \).

**Proof:** Since any SQLO\(_l\)(\( \eta, d \)) sequence is a quantized \( B_d \) sequence, the proof follows directly from Thm. 2.
A. Fundamental limits and construction of SQLO\(l\) sequences

In [28], two types of lexicographically ordered sequences were defined that are closely related to the SQLO\(l\) sequences. For simplicity, we call these sequences “lex\((h)\)” and “strong-lex\((h)\)” and we provide their definition for completeness.

**Definition 11.** A sequence of positive integers \(B = \{\beta_1, \beta_2, \ldots\}\) is a lex\((h)\) sequence, if for any two distinct subsets \(B_1 = \{\beta'_1, \beta'_2, \ldots, \beta'_h\}\) and \(B_2 = \{\beta''_1, \beta''_2, \ldots, \beta''_h\}\) with elements listed in an increasing order, one has \(\sum_{\beta''_i \in B_2} \beta''_i > \sum_{\beta'_i \in B_1} \beta'_i\) if there exists an integer \(r, 1 \leq r \leq h\), such that \(\forall i, 1 \leq i < r, \beta'_i = \beta''_i\) and \(\beta''_r > \beta'_r\).

**Definition 12.** A sequence of positive integers \(B = \{\beta_1, \beta_2, \ldots\}\) is a strong-lex\((h)\) sequence, if it is a lex\((s)\) sequence \(\forall s \leq h\); in addition, for any two subsets \(B_1 \subseteq B\) and \(B_2 \subseteq B\) such that \(|B_1| < |B_2| \leq h\), one has \(\sum_{\beta_i \in B_2} \beta_i > \sum_{\beta_i \in B_1} \beta_i\).

The strong-lex\((h)\) sequences can be used to construct SQLO\(l\) sequences as shown in the next proposition.

**Proposition 3.** Consider a SQGT model with thresholds \(\eta = [0, \eta_1, \eta_2, \ldots, \eta_Q]^T\); \(\forall s : 1 \leq s \leq Q\), let \(g_s = \max_{i: 1 \leq i \leq s} \eta_i - \eta_{i-1}\) be the largest gap of the first \(s\) thresholds. Let \(B = \{\beta_1 < \beta_2 < \ldots\}\) be a strong-lex\((h)\) sequence. For a fixed \(s, 2 \leq s \leq Q\), let \(K_s\) be a positive integer small enough to satisfy \(\eta_s > g_s \sum_{i=\max\{1, K_s-h\}}^{K_s} \beta_i\). Then all the sequences of the form \(A_s = \{g_s \beta_1, g_s \beta_2, \ldots, g_s \beta_{K_s}\}\) are SQLO\(l\)(\(\eta, h\)) sequences.

**Proof:** The proof of this proposition follows along the same lines as the proof of Thms. 3 and 6 and is hence omitted.

As we demonstrated through a simple example earlier, the SQLO\(l\) properties may result in denser sequences compared to SQLO\(s\) properties. However, a SQLO\(l\)(\(\eta, h\)) sequence constructed from strong-lex\((h)\) sequences according to Proposition 3 does not improve the bound \(\alpha_K = O_g(\gamma^K)\) derived in Lemma 2. This can be shown as follows. We define an optimal lex\((h)\) sequence as a lex\((h)\) sequence \(B = \{\beta_1, \beta_2, \ldots, \beta_K\}\) with the smallest possible value of \(\beta_K\). In [28, Thm. 1], it was proven that the largest element of an optimal lex\((h)\) sequence satisfies
\[ \beta_K = O(\gamma^K), \] where \( \gamma \) is the largest root of \( x^{h+1} - 2x^h + 1 = 0 \). Since any strong-lex(h) sequence needs to also satisfy the lex(h) property, one can conclude that a SQLO\(_t\)\((\eta, h)\) sequence constructed from strong-lex(h) sequences according to Proposition 3 cannot improve the bound \( \alpha_K = O_\eta(\gamma^K) \).

B. Decoding algorithm for SQGT codes constructed using SQLO\(_t\) sequences

Next, we describe the Dec-SQLO\(_t\) algorithm, the decoding procedure for codes based on SQLO\(_t\) sequences. This algorithm resembles the Dec-SQLO\(_s\) algorithm, and similar intuition also applies as follows. In the first step, one identifies \( \mathcal{X} = \mathcal{X}_D \), the set of binary codewords corresponding to the support of the codewords in \( D \). To complete the decoding, one needs to identify the set of elements \( A_{i,t} \subseteq A \) which are used to form the codewords in \( D \) from the binary codeword \( x_i \), \( \forall x_i \in \mathcal{X}_D \).

Since in the proof of Lemma 3 we only used the quantized \( B_d \) property, this lemma also holds for codes constructed using Thm. 3. As a result, using the notation defined in Lemma 3 for any binary codeword \( x_i \in \mathcal{X}_D \) there exists at least \( e+1 \) elements of \( S_i', \) denoted by \( S_i'' \), such that \( \forall j \in S_i'' \) one has \( y(j) = f_\eta(\sum_{\alpha \in A_{i,t}} \alpha) \). This implies that the majority of elements \( y(j), j \in S_i' \), correctly correspond to the bin in which \( \beta_t = \sum_{\alpha \in A_{i,t}} \alpha \) is located. Each correctly identified bin contains a limited number of integers, one of which is the true value of \( \beta_t \). As a result, by testing each such integer, we can determine whether it can be written as the sum of up to \( d \) elements of \( A \) or not. The integer for which the answer to this query is positive is equal to \( \beta_t \), which can then be used to identify the elements of \( A_{i,t} \).

As a result, given a SQLO\(_t\) sequence \( A \) and an integer \( \beta \), the main issue is to efficiently determine whether \( \beta \) can be written as the sum of up to \( d \) elements of \( A \); and if so, what those elements are. In Lemma 4, an algorithm with computational complexity of \( O(K) \) is described that can perform this task.

**Lemma 4.** Given a SQLO\(_t\)(\( \eta, d \)) sequence \( A \) and a fixed integer \( \beta \), it is possible to identify whether \( \beta \) can be written as a sum of up to \( d \) elements of \( A \) with complexity \( O(K) \), where

\[ \gamma \text{ is defined as the largest root of } x^{h+1} - 2x^h + 1 = 0. \]

\[ \text{However, it is evident from the proof of the theorem that } \gamma \text{ is in fact the largest root of } x^{h+1} - 2x^h + 1 = 0. \]
Given that the answer to this question is positive, one can identify these elements of $A$ with complexity $O(K)$.

Proof: Suppose $\beta$ can be written as the sum of $s \leq d$ elements of $A$, and let $A_t \subseteq A$ be the subset such that $\sum_{\alpha \in A_t} \alpha = \beta$. The value of $s = |A_t|$ can be easily determined as follows. First, we form the set $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$, where $\gamma_i = \sum_{j=1}^{i} \alpha_i$, $1 \leq i \leq K$. As a consequence of the second property in Def. [10] for any $1 \leq i \leq K$, $\gamma_i$ is larger than all $j$-subsets of $A$ for $j < i$. On the other hand, due to the third property in Def. [10] $\gamma_i$ is smaller than all $j$-subsets of $A$ for $j \geq i$. Consequently, one can determine $s$ using $s = \min\{i : \beta < \gamma_i\} - 1$.

Given the value of $s$, we can determine the elements of $A_t$ successively using $K$ iterations. First, we initialize the procedure by setting $s' \leftarrow s$, $\beta' \leftarrow \beta$, and $A' \leftarrow \emptyset$. In the $i$-th iteration, $1 \leq i \leq K$, we determine whether $\alpha_i \in A_t$ or not. At the beginning of the $i$-th iteration, $A'$ equals $A_t \cap \{\alpha_1, \alpha_2, \ldots, \alpha_{i-1}\}$, and $s'$ is equal to the number of remaining unidentified elements of $A_t$, i.e. $s' = |A_t \setminus A'|$. In addition, $\beta'$ is equal to the sum of $s'$ elements in $A_t \setminus A'$. To determine whether $\alpha_i$ is in $A_t$, we use the following rule: if $\beta' < \alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_{i+s'}$, then $\alpha_i \in A_t$; the reason is that the sum of any $s'$ elements of $\{\alpha_i, \alpha_{i+1}, \ldots, \alpha_K\}$ that does not include $\alpha_i$ is at least as large as $\alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_{i+s'}$. Therefore, if $\beta' < \alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_{i+s'}$, then $\alpha_i$ must be in $A_t$. Given that this condition is satisfied, we update $A' \leftarrow A' \cup \{\alpha_i\}$, $\beta' \leftarrow \beta' - \alpha_i$, and $s' \leftarrow s' - 1$. Otherwise, we go to the next iteration. The algorithm stops after $K$ iterations. At the end, if $s' = 0$ and $|A'| \leq d$, the answer to the first query is positive and $A' = A_t$. Otherwise the answer to the query is negative.

The decoding algorithm for codes constructed using Thm. [8] is described in Algorithm 3. Note that the main difference between this algorithm and Algorithm 2 is that in Step 2, we use Lemma [4] instead of Proposition [2].

ACKNOWLEDGMENT

This work was supported in part by NSF grants CCF 0809895, CCF 1218764 and the Emerging Frontiers for Science of Information Center, CCF 0939370. The authors would like to thank Farzad Farnoud (Hassanzadeh), Han Mao Kiah, and Gregory Puleo for useful discussions.
Algorithm 3: Dec-SQLO

Input: $y \in [Q]^m$, $C_b \in [2]^{m \times N}$, $\eta$, $A$, $e \geq 0$
Output: $\hat{D}$

Step 1: Initialize $X \leftarrow \emptyset$ and $\hat{D} \leftarrow \emptyset$
For $i = 1, 2, \ldots, \frac{n}{K}$ do
  $x_i \leftarrow$ the $i$-th codeword of $C_b$
  $N_i \leftarrow$ number of coordinates $j$ for which $x_i(j) > y(j)$
  If $N_i \leq e$ then
    Set $X \leftarrow X \cup \{x_i\}$
End

Step 2: For $i = 1, 2, \ldots, |X|$ do
  Set $S_i \leftarrow \{\text{the set of nonzero coordinates of } x_i\}$
  Set $S'_i \leftarrow \{\text{subset of } S_i \text{ with } |S'_i| = 2e + 1 \text{ s.t. } \forall k \in S'_i \text{ and } \forall j \in S_i \setminus S'_i, \text{ one has } y(k) \leq y(j)\}$
  Initialize the multiset $B' \leftarrow \emptyset$
  For $j = 1, 2, \ldots, |S'_i|$ do
    $\eta_u \leftarrow$ the upper threshold of the quantization bin for $y(j)$
    $\eta_l \leftarrow$ the lower threshold of the quantization bin for $y(j)$
    $\beta \leftarrow$ the unique integer $\eta_l \leq \beta < \eta_u$ that can be written as the sum of up to $d$ elements of $A$ (use Lemma 4)
    Update the multiset $B' \leftarrow B' \cup \{\beta\}$
End
  Set $\hat{\beta}_i \leftarrow$ the element of $B'$ with at least $e + 1$ repetitions
  Set $\hat{A}_{i,t} \leftarrow \{\text{the unique subset of } A \text{ with the sum equal to } \hat{\beta}_i\}$
  Set $\hat{D}_i \leftarrow \{\text{codewords of } C \text{ of the form } z = \alpha x_i, \forall \alpha \in \hat{A}_{i,t}\}$
End

Return $\hat{D} = \bigcup_i \hat{D}_i$

REFERENCES

[1] A. Emad and O. Milenkovic, “Group testing for non-uniformly quantized adder channels,” Proc. IEEE Int. Symp. Inf. Theory (ISIT’14), July 2014, pp. 2351-2355.
[2] D. -Z. Du and F. K. Hwang, Pooling designs and nonadaptive group testing, NJ: World Scientific, 2006.
[3] J. K. Wolf, “Born again group testing: multi-access communications,” IEEE Trans. Inf. Theory, vol. 31, pp. 185-191.
[4] P. -O. Anderson, “Superimposed codes for the Euclidean channel,” Ph.D. thesis no. 342, Linköping Univ., Linköping, Sweden, 1994.
[5] P. Z. Fan, M. Darnell, and B. Honary, “Superimposed codes for the multiaccess binary adder channel,” IEEE Trans. Inf. Theory, vol. 41, no. 4, pp. 1178-1182, 1995.
[6] E. Amiri, and G. Tardos, “High rate fingerprinting codes and fingerprinting capacity,” in Proc. 20th ACM-SIAM Sympos. Discrete Algorithms, New York, Jan. 2009, pp. 336-345.
[7] A. Barg, G. R. Blakley, and G. A. Kabatiansky, “Digital fingerprinting codes: problem statements, constructions, identification of traitors,” IEEE Trans. Inf. Theory, vol. 49, pp. 852-865, 2003.
[8] C. J. Colbourn, G. Kéri, R. P. Rivas Soriano, and J. -C. Schlage-Puchta, “Covering and radius-covering arrays: constructions and classification,” Discrete Appl. Math., vol. 158, pp. 1158-1180, 2010.
[9] A. G. D’yachkov, and V. V. Rykov, “A coding model for a multiple-access adder channel,” Probl. Perdachi Inform. vol. 17, pp. 26-32, 1981 (in Russian).
[10] I. Bar-David, E. Plotnik, and R. Rom, “Forward collision resolution – a technique for random multiple-access to the adder channel,” IEEE Trans. Inf. Theory, vol. IT-39, pp. 1671-1675, 1993.
[11] D. -Z. Du and F. K. Hwang, Combinatorial group testing and its applications, NJ: World Scientific, 2000.
[12] Y. Erlich, A. Gordon, M. Brand, G. Hannon, and P. P. Mitra, “Compressed genotyping,” IEEE Trans. Inf. Theory, vol. 56, pp. 706-723, 2010.
[13] N. Shental, A. Amnon, and O. Zuk, “Identification of rare alleles and their carriers using compressed sequencing,” Nucleic Acids Research, vol. 38, no. 19, pp. 1-22, 2010.
[14] W. Dai, M. Sheikh, O. Milenkovic, and R. Baraniuk, “Compressive sensing DNA microarrays,” EURASIP J. Bioinform. Syst. Biol., 2009:162824, 2009.
[15] R. Dorfman, “The detection of defective members of large populations,” Ann. Math. Stat., vol. 14, pp. 436-440, 1943.
[16] W. Kautz and R. Singleton, “Nonrandom binary superimposed codes,” IEEE Trans. Inf. Theory, vol. 10, no. 4, pp. 363-377, Oct. 1964.
[17] P. Damaschke, “Threshold group testing,” General Theory of Information Transfer and Combinatorics in: LNCS, vol. 4123, pp. 707-718, 2006.
[18] A. Emad and O. Milenkovic, “Symmetric Group Testing and Superimposed Codes,” Proc. Inf. Theory Workshop (ITW’11), Oct. 2011, pp. 20-24.
[19] A. Emad and O. Milenkovic, “Semi-quantitative Group Testing,” Proc. IEEE Int. Symp. Inf. Theory (ISIT’12), July 2012, pp. 1847-1851.
[20] A. Emad and O. Milenkovic, “Semiquantitative Group Testing,” to appear in IEEE Trans. Inf. Theory, 2014. doi: 10.1109/TIT.2014.2327630
[21] A. G. D’yachkov, V. V. Rykov, “Generalized superimposed codes and their application to random multiple access”, in Proc. 6th Int. Symp. Inf. Theory, part 1, Taschkent, 1984.
[22] A. G. D’yachkov, “Lectures on designing screening experiments,” Lecture Note Series 10, Feb. 2004.
[23] H. Halberstam and K. F. Roth. Sequences, New York: Springer, 1983.
[24] W. F. Lunnon, “Integer sets with distinct subset sums,” Math. Compute., vol. 50, pp. 297-320, 1988.
[25] T. Bohman, “A construction for sets of integers with distinct subset sums,” Electron. J. Combin., vol. 5, R3, 1998.
[26] J. Pieprzyk, T. Hardjono, and J. Seberry, Fundamentals of Computer Security, NY: Springer, 2003.
[27] V. V. Prasolov, “Polynomials,” Algorithms and Computation in Mathematics, ser. 11, Springer, 2010.
[28] M. D. Atkinson, A. Negro, and N. Santoro, “Sums of lexicographically ordered sets,” Discrete Mathematics, vol. 80, pp. 115-122, 1990.