Regular Paper

Efficient (nonrandom) Construction and Decoding for Non-adaptive Group Testing

Thach V. Bui[1,a)] Minoru Kuribayashi[2,b)] Tetsuya Kojima[3,c)] Roghayyeh Haghvirdinezhad[4,d)]
Isao Echizen[5,e)]

Received: June 7, 2018, Accepted: December 4, 2018

Abstract: The task of non-adaptive group testing is to identify up to d defective items from N items, where a test is positive if it contains at least one defective item, and negative otherwise. If there are t tests, they can be represented as a $t \times N$ measurement matrix. We have answered the question of whether there exists a scheme such that a larger measurement matrix, built from a given $t \times N$ measurement matrix, can be used to identify up to d defective items in time $O(t \log_2 N)$. In the meantime, a $t \times N$ nonrandom measurement matrix with $t = O\left(\frac{d^2 \log_2^2 N}{(\log_2(d \log_2 N) - \log_2 \log_2(d \log_2 N))^2}\right)$ can be obtained to identify up to d defective items in time poly(t). This is much better than the best well-known bound, $t = O\left(d^2 \log_2^2 N\right)$. For the special case $d = 2$, there exists an efficient nonrandom construction in which at most two defective items can be identified in time $4 \log_2^2 N$ using $t = 4 \log_2^2 N$ tests. Numerical results show that our proposed scheme is more practical than existing ones, and experimental results confirm our theoretical analysis. In particular, up to $2^7 = 128$ defective items can be identified in less than 16 s even for $N = 2^{100}$.

Keywords: non-adaptive group testing, Nonrandom construction, Efficient decoding, Combinatorics

1. Introduction

Group testing dates back to World War II, when an economist, Robert Dorfman, solved the problem of identifying which draftees had syphilis [10]. It turned out to a problem of finding up to d defective items in a huge number of items N by testing t subsets of N items. The meanings of "items", "defective items", and "tests" depend on the context. Classically, a test is positive if there is at least one defective item, and negative otherwise. Damaschke [9] generalized this problem into threshold group testing in which a test is positive if it contains at least u defective items, negative if it contains at most l defective items, and arbitrary otherwise. If $u = 1$ and $l = 0$, threshold group testing reduces to classical group testing.

In this work, we focus on classical group testing in which a test is positive if there exists at least one defective item, and negative otherwise. There are two main approaches to testing design: adaptive and non-adaptive. In *adaptive group testing*, tests are performed in a sequence of stages, and the designs of later tests depend on the results of earlier tests. With this approach, the number of tests can be theoretically optimized [11]. However, the testing can take a long time if there are many stages. Therefore, *non-adaptive group testing* (NAGT) [12] is preferred: all tests are designed in advance and performed simultaneously. The growing use of NAGT in various fields such as compressed sensing [1], data streaming [8], DNA library screening [19], and neuroscience [3] has made it increasingly attractive recently. The focus here is thus on NAGT.

If t tests are needed to identify up to d defective items among N items, they can be seen as a $t \times N$ measurement matrix. The procedure to get the matrix is called *construction*, the procedure to get the outcome of t tests using the measurement matrix is called *encoding*, and the procedure to get the defective items from t outcomes is called *decoding*. Note that the encoding procedure includes the construction procedure. The objective of NAGT is to design a scheme such that all defective items are "efficiently" identified from the encoding and decoding procedures. Six criteria determine the efficiency of a scheme: measurement matrix construction type, number of tests needed, decoding time, time needed to generate an entry for the measurement matrix, space needed to generate a measurement matrix entry, and probability of successful decoding. The last criterion reduces the number of tests and/or the decoding complexity. With high probability, Cai et al. [5] and Lee et al. [18] achieved a low number of tests and decoding complexity, namely $O(t)$, where $t = O(d \log d \cdot \log N)$ (log is referred to as the logarithm of base 2). However, the construction type is random, and the whole measurement matrix must be stored for implementation, so it is limited to real-time applications. For example, in a data stream [8], routers have limited

1 SOKENDAI (The Graduate University for Advanced Studies), Miura-gun, Kanagawa 240–0193, Japan
2 Graduate School of Natural Science and Technology, Okayama University, Okayama 700–8530, Japan
3 National Institute of Technology, Tokyo College, Hachioji, Tokyo 193–0997, Japan
4 New Jersey Institute of Technology, New Jersey, USA
5 National Institute of Informatics, Chiyoda, Tokyo 101–8430, Japan
a) bvthach@nii.ac.jp
b) kminoru@okayama-u.ac.jp
c) kojt@tokyo-ct.ac.jp
d) rh284@njit.edu
e) iechizen@nii.ac.jp

Table 1 Comparison with existing schemes.

No.	Scheme	Construction type	Number of tests t	Decoding time	Time to generate an entry	Space to generate an entry
⟨1⟩	Indyk et al. [16] (Theorem 3)	Nonrandom	$O(d^2 \log^2 N)$	$O\left(\frac{d^9 (\log N)^{16+1/3}}{(\log(d \log N))^{7+1/3}}\right)$	$O(t)$	$O(t)$
⟨2⟩	Indyk et al. [16] (Theorem 2)	Nonrandom	$O(d^2 \log N)$	$\text{poly}(t) = O\left(d^{11} \log^{17} N\right)$	$\text{poly}(t, N)$	$\text{poly}(t)$
⟨**3**⟩	**Proposed (Theorem 8)**	Nonrandom	$O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log\log(d \log N))^2}\right)$	$O\left(\frac{d^{3.57} \log^{6.26} N}{(\log(d \log N) - \log\log(d \log N))^{6.26}}\right)$ $+O\left(\frac{d^6 \log^4 N}{(\log(d \log N) - \log\log(d \log N))^4}\right)$	$O(t)$	$O(t)$
⟨**4**⟩	**Proposed (Corollary 3)**	Nonrandom	$O\left(\frac{d^2 \log^3 N}{(\log(d \log N) - \log\log(d \log N))^2}\right)$	$O(t)$	$O(t)$	$O(t)$
⟨5⟩	Porat-Rothschild [21] (Theorem 1)	Nonrandom	$O(d^2 \log N)$	$O(tN) = O(d^2 \log N \times N)$	$O(tN)$	$O(tN)$
⟨**6**⟩	**Proposed (Corollary 2)**	Nonrandom	$O(d^2 \log^2 N)$	$O(t) = O(d^2 \log^2 N)$	$O(tN)$	$O(tN)$
⟨7⟩	Indyk et al. [16] (Theorem 3)	Nonrandom $d = 2$	$2 \log N(2 \log N - 1)$	$\frac{2^9 (\log N)^{16+1/3}}{(\log(2 \log N))^{7+1/3}}$	$\log^2 N$	$\log N$
⟨**8**⟩	**Proposed (Theorem 7)**	Nonrandom $d = 2$	$4 \log^2 N$	$4 \log^2 N$	4	$2 \log N$ $+ \log(2 \log N)$
⟨9⟩	Indyk et al. [16] (Theorem 2)	Random	$O(d^2 \log N)$	$\text{poly}(t) = O\left(d^{11} \log^{17} N\right)$	$O(t^2 \log N)$	$O(t \log N)$
⟨**10**⟩	**Proposed (Corollary 1)**	Random	$O(d^2 \log^2 N)$	$O(t) = O(d^2 \log^2 N)$	$O(t^2)$	$O(t \log N)$
⟨**11**⟩	**Proposed (Corollary 4)**	Random	$O(d \log N \cdot \log \frac{d}{\epsilon})$	$O(d \log N \cdot \log \frac{d}{\epsilon})$	$O(tN)$	$O(tN)$

resources and need to be able to access the column in the measurement matrix assigned to an IP address as quickly as possible to perform their functions. The schemes proposed by Cai et al. [5] and Lee et al. [18], therefore, are inadequate for this application.

For exact identification of defective items, there are four main criteria to be considered: measurement matrix construction type, number of tests needed, decoding time, and time needed to generate measurement matrix entry. The measurement matrix is nonrandom if it always satisfies the preconditions after the construction procedure with probability 1. It is random if it satisfies the preconditions after the construction procedure with some probability. A $t \times N$ measurement matrix is more practical if it is nonrandom, t is small, the decoding time is a polynomial of t (poly(t)), and the time to generate its entry is also poly(t). However, there is always a trade-off between these criteria.

Kautz and Singleton [17] proposed a scheme in which each entry in a $t \times N$ measurement matrix can be generated in poly(t), where $t = O(d^2 \log^2 N)$. However, the decoding time is $O(tN)$. Indyk et al. [16] reduced the decoding time to poly(t) while maintaining the order of the number of tests and the time to generate the entries. However, the number of tests in a nonrandom measurement matrix is not optimal.

In term of the pessimum number of tests, Guruswami and Indyk [14] proposed a linear-time decoding scheme in accordance with the number of tests of $O(d^4 \log N)$. To achieve an optimal bound on the number of tests, i.e., $O(d^2 \log N)$, while maintaining a decoding time of poly(t) and keeping the entry computation time within poly(t), Indyk et al. [16] proposed a random construction. Although they tried to derandomize their schemes, it takes $\text{poly}(t, N)$ time to construct such matrices, which is impractical when d and N are sufficiently large.

Cheraghchi [6] achieved similar results. However, his proposed scheme can deal with the presence of noise in the test outcomes. Porat and Rothschild [21] showed that it is possible to construct a nonrandom $t \times N$ measurement matrix in time $O(tN)$ while maintaining the order of the number of tests, i.e., $O(d^2 \log N)$. However, each entry in the resulting matrix is identified after the construction is completed. This is equivalent to each entry being generated in time $O(tN)$. If we reduce the number of tests, the nonrandom construction proposed by Indyk et al. [16] is the most practical.

1.1 Contributions

Overview: There are two main contributions in this work. First, we have answered the question of whether there exists a scheme such that a larger measurement matrix, built from a given $t \times N$ measurement matrix, can be used to identify up to d defective items in time $O(t \log N)$. Second, a $t \times N$ nonrandom measurement matrix with $t = O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log\log(d \log N))^2}\right)$ can be obtained to identify up to d defective items in time poly(t). This is much better than the best well-known bound $t = O\left(d^2 \log^2 N\right)$. There is a special case for $d = 2$ in which there exists a $4 \log^2 N \times N$ nonrandom measurement matrix such that it can be used to identify up to two defective items in time $4 \log^2 N$. Numerical results show that our proposed scheme is the most practical and experimental results confirm our theoretical analysis. For instance, at most $2^7 = 128$ defective items can be identified in less than 16 s even for $N = 2^{100}$.

Comparison: We compare variants of our proposed scheme with existing schemes in **Table 1**. As mentioned above, six criteria determine the efficiency of a scheme: measurement matrix construction type, number of tests needed, decoding time, time needed to generate measurement matrix entry, space needed to generate a measurement matrix entry, and probability of successful decoding. Since the last criterion is only used to reduce the number of tests, it is not shown in the table. If the number of tests and the decoding time are the top priorities, the construction

in ⟨**11**⟩ is the best choice. However, since the probability of successful decoding is at least $1-\epsilon$ for any $\epsilon > 0$, some defective items may not be identified.

From here on, we assume that the probability of successful decoding is 1; i.e., all defective items are identified. There are tradeoffs among the first five criteria. When $d = 2$, the number of tests with our proposed scheme (⟨**8**⟩) is slightly larger than that with ⟨7⟩, although our proposed scheme has the best performance for the remaining criteria. When $d > 2$, the comparisons are as follows. First, if the generation of a measurement matrix must be certain, the best choices are ⟨1⟩, ⟨2⟩, ⟨**3**⟩, ⟨**4**⟩, ⟨5⟩, and ⟨**6**⟩. Second, if the number of tests must be as low as possible, the best choices are ⟨2⟩, ⟨5⟩, and ⟨9⟩. Third, if the decoding time is most important, the best choices are three variations of our proposed scheme: ⟨**4**⟩, ⟨**6**⟩, and ⟨**10**⟩. Fourth, if the time needed to generate a measurement matrix entry is most important, the best choices are ⟨1⟩, ⟨3⟩, ⟨**4**⟩, ⟨7⟩, ⟨9⟩ and ⟨**10**⟩. Finally, if the space needed to generate a measurement matrix entry is most important, the best choices are ⟨1⟩, ⟨2⟩, ⟨**3**⟩, ⟨**4**⟩, ⟨7⟩, ⟨9⟩ and ⟨**10**⟩.

For real-time applications, because "defective items" are usually considered to be *abnormal system activities* [8], they should be identified as quickly as possible. It is thus acceptable to use extra tests to speed up their identification. Moreover, the measurement matrix deployed in the system should not be stored in the system because of saving space. Therefore, the construction type should be nonrandom, and the time and space needed to generate an entry should be within poly(t). Thus, the best choice is ⟨**4**⟩ and the second best choice is ⟨**3**⟩.

1.2 Outline

The paper is organized as follows. Section 2 presents some preliminaries on tensor product, disjunct matrices, list-recoverable codes, and a previous scheme. Section 3 describes how to achieve an efficient decoding scheme when a measurement matrix is given. Section 4 presents nonrandom constructions for identifying up to two or more defective items. The numerical and experimental results are presented in Section 5. The final section summarizes the key points and addresses several open problems.

2. Preliminaries

Notation is defined here for consistency. We use capital calligraphic letters for matrices, non-capital letters for scalars, and bold letters for vectors. Matrices and vectors are binary. The frequently used notations are as follows:

- $N; d$: number of items; maximum number of defective items. For simplicity, suppose that N is the power of 2.
- $|\cdot|$: weight; i.e, number of non-zero entries of input vector or cardinality of input set.
- $\otimes, \circledcirc, \circ$: operation for NAGT, tensor product, concatenation code (to be defined later).
- $\mathcal{S}, \mathcal{B}$: $k \times N$ measurement matrices used to identify at most one defective item, where $k = 2\log_2 N$.
- $\mathcal{M} = (m_{ij})$: $t \times N$ d-disjunct matrix, where integer $t \geq 1$ is number of tests.
- $\mathcal{T} = (t_{ij})$: $T \times N$ measurement matrix used to identify at most d defective items, where integer $T \geq 1$ is number of tests.
- $\mathbf{x}; \mathbf{y}$: binary representation of N items; binary representation of test outcomes.
- $\mathcal{S}_j, \mathcal{B}_j, \mathcal{M}_j, \mathcal{M}_{i,*}$: column j of matrix $\mathcal{S}$, column j of matrix $\mathcal{B}$, column j of matrix $\mathcal{M}$, row i of matrix $\mathcal{M}$.
- $\mathbb{D}$: index set of defective items, e.g., $\mathbb{D} = \{2, 6\}$ means items 2 and 6 are defective.
- $\mathrm{diag}(\mathcal{M}_{i,*}) = \mathrm{diag}(m_{i1}, m_{i2}, \ldots, m_{iN})$: diagonal matrix constructed by input vector $\mathcal{M}_{i,*} = (m_{i1}, m_{i2}, \ldots, m_{iN})$.
- $\mathrm{e}, \log, \ln, \exp(\cdot)$: base of natural logarithm, logarithm of base 2, natural logarithm, exponential function.
- $\lceil x \rceil, \lfloor x \rfloor$: ceiling and floor functions of x.

2.1 Tensor Product

Given an $f \times N$ matrix $\mathcal{A}$ and an $s \times N$ matrix $\mathcal{S}$, their tensor product $\circledcirc$ is defined as

$$\mathcal{R} = \mathcal{A} \circledcirc \mathcal{S} = \begin{bmatrix} \mathcal{S} \times \mathrm{diag}(\mathcal{A}_{1,*}) \\ \vdots \\ \mathcal{S} \times \mathrm{diag}(\mathcal{A}_{f,*}) \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} a_{11}\mathcal{S}_1 & \ldots & a_{1N}\mathcal{S}_N \\ \vdots & \ddots & \vdots \\ a_{f1}\mathcal{S}_1 & \ldots & a_{fN}\mathcal{S}_N \end{bmatrix}, \tag{2}$$

where diag(.) is the diagonal matrix constructed by the input vector, $\mathcal{A}_{h,*} = (a_{h1}, \ldots, a_{hN})$ is the hth row of $\mathcal{A}$ for $h = 1, \ldots, f$, and $\mathcal{S}_j$ is the jth column of $\mathcal{S}$ for $j = 1, \ldots, N$. The size of $\mathcal{R}$ is $r \times N$, where $r = fs$. One can imagine that an entry a_{hj} of matrix $\mathcal{A}$ would be replaced by the vector $a_{hj}\mathcal{S}_j$ after the tensor product is used. For instance, suppose that $f = 2, s = 3$, and $N = 4$. Matrices $\mathcal{A}$ and $\mathcal{S}$ are defined as

$$\mathcal{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \quad \mathcal{S} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3}$$

Then $\mathcal{R} = \mathcal{A} \circledcirc \mathcal{S}$ is

$$\mathcal{R} = \mathcal{A} \circledcirc \mathcal{S} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \circledcirc \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} 1 \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & 0 \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & 1 \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & 0 \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ 0 \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & 1 \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & 1 \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & 1 \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6}$$

2.2 Disjunct Matrices

To gain insight into disjunct matrices, we present the concept of an identity matrix inside a set of vectors. This concept is used

to later construct a d-disjunct matrix.

Definition 1. *Any c column vectors with the same size contain a $c \times c$ identity matrix if a $c \times c$ identity matrix could be obtained by placing those columns in an appropriate order.*

Note that there may be more than one identity matrix inside those c vectors. For example, let $\mathbf{b}_1, \mathbf{b}_2$, and $\mathbf{b}_3$ be vectors of size 4×1:

$$\mathbf{b}_1 = \begin{bmatrix}1\\0\\0\\1\end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix}1\\1\\0\\0\end{bmatrix}, \mathbf{b}_3 = \begin{bmatrix}1\\0\\1\\1\end{bmatrix}. \tag{7}$$

Then, $(\mathbf{b}_1, \mathbf{b}_2)$ and $(\mathbf{b}_2, \mathbf{b}_3)$ contain 2×2 identify matrices, whereas $(\mathbf{b}_1, \mathbf{b}_3)$ does not.

$$\begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2\end{bmatrix} = \begin{bmatrix}1&1\\0&1\\0&0\\1&0\end{bmatrix}, \begin{bmatrix}\mathbf{b}_2 & \mathbf{b}_3\end{bmatrix} = \begin{bmatrix}1&1\\1&0\\0&1\\0&1\end{bmatrix} = \begin{bmatrix}1&1\\1&0\\0&1\\0&1\end{bmatrix}.$$

The union of l vectors is defined as follows. Given l binary vectors $\mathbf{y}_w = (y_{1w}, y_{2w}, \ldots, y_{Bw})^T$ for $w = 1, \ldots, l$ and some integer $B \geq 1$, their union is defined as vector $\mathbf{y} = \vee_{i=1}^{l} \mathbf{y}_i = (\vee_{i=1}^{l} y_{1i}, \ldots, \vee_{i=1}^{l} y_{Bi})^T$, where $\vee$ is the OR operator.

Definition 1 is interchangeably defined as follows: the union of at most $c-1$ vectors does not contain the remaining vector. Here we use definition 1, so the definition for a d-disjunct matrix is as follows.

Definition 2. *A binary $t \times N$ matrix is called a d-disjunct matrix iff there exists an $(d+1) \times (d+1)$ identity matrix in a set of $d+1$ columns arbitrarily selected from the matrix.*

For example, a 3×3 identity matrix is a 2-disjunct matrix. The encoding and decoding procedures used to identify up to d defective items using a d-disjunct matrix are as follows. Suppose that $\mathcal{M} = (m_{ij})$ is a $t \times N$ measurement matrix, which is used to identify at most d defective items. Item j is represented by column $\mathcal{M}_j$ for $j = 1, \ldots, N$. Test i is represented by row i in which $m_{ij} = 1$ iff the item j belongs to test i, and $m_{ij} = 0$ otherwise, where $i = 1, \ldots, t$. Usually, $\mathcal{M}$ is a d-disjunct matrix, but this is not a requirement. In Section 3, we will see that $\mathcal{M}$ may not be d-disjunct and still be able to identify up to d defective items.

Let $\mathbf{x} = (x_1, \ldots, x_N)^T$ be a binary representation for N items, in which $x_j = 1$ iff item j is defective for $j = 1, \ldots, N$. The outcome of t tests, denoted as $\mathbf{y} = (y_1, \ldots, y_t)^T \in \{0,1\}^t$, is:

$$\mathbf{y} = \mathcal{M} \otimes \mathbf{x} = \bigvee_{j=1}^{N} x_j \mathcal{M}_j = \bigvee_{j \in \mathbb{D}} \mathcal{M}_j, \tag{8}$$

where $\mathbb{D}$ is the index set of defective items. The construction procedure is used to get $\mathcal{M}$. The encoding procedure (which includes the construction procedure) is used to get $\mathbf{y}$. The decoding procedure is used to recover $\mathbf{x}$ from $\mathbf{y}$ and $\mathcal{M}$.

We next present some recent results for the construction and decoding of disjunct matrices. With naive decoding, all items belonging to tests with negative outcomes are removed; the items remaining are considered to be defective. The decoding complexity of this approach is $O(tN)$. Naive decoding is used only a little here because the decoding time is long. A matrix is said to be nonrandom if its columns are deterministically generated without using randomness. In contrast, a matrix is said to be random if its columns are randomly generated. We thus classify construction types on the basis of the time it takes to generate a matrix entry. A $t \times N$ matrix is said to be weakly explicit if each of its columns is generated in time (and space) $O(tN)$. It is said to be strongly explicit if each of its columns is generated in time (and space) poly(t). We first present a weakly explicit construction of a disjunct matrix.

Theorem 1 (Theorem 1 [21]). *Given $1 \leq d < N$, there exists a nonrandom $t \times N$ d-disjunct matrix that can be constructed in time $O(tN)$, where $t = O(d^2 \log N)$. Moreover, the decoding time is $O(tN)$, and each column is generated in time (and space) $O(tN)$.*

The second construction is strongly explicit.

Theorem 2 (Corollary 5.1 [16]). *Given $1 \leq d < N$, there exists a random $t \times N$ d-disjunct matrix that can be decoded in time poly$(t) = O(d^{11} \log^{17} N)$, where $t = 4800 d^2 \log N = O(d^2 \log N)$. Each column can be generated in time $O(t^2 \log N)$ and space $O(t \log N)$. There also exists a matrix that can be nonrandomly constructed in time poly(t, N) and space poly(t) while the construction time and space for each column of the matrix remain same.*

Finally, the last construction is nonrandom. We analyze this construction in detail for later comparison. Although the precise formulas were not explicitly given in Ref. [16], they can be derived.

Theorem 3 (Corollary C.1 [16]). *Given $1 \leq d < N$, a nonrandom $t \times N$ d-disjunct matrix can be decoded in time $O\left(\frac{d^9 (\log N)^{16+1/3}}{(\log(d \log N))^{7+1/3}}\right) = \text{poly}(t)$, where $t = O(d^2 \log^2 N)$. Moreover, each entry (column) can be generated in time (and space) $O(t)$ ($O(t^{3/2})$). When $d = 2$, the number of tests is $2 \log N \times (2 \log N - 1)$, the decoding time is longer than $\frac{2^9 (\log N)^{16+1/3}}{(\log(2 \log N))^{7+1/3}}$, and each entry is generated in time $\log^2 N$ and space $\log N$.*

2.3 List Recoverable Codes

There may be occasions in the physical world where a person might want to recover a similar codeword from a given codeword. For example, a person searching on a website such as Google might be searching using the word "intercept". However, mistyping results in the input word being "inrercep". The website should suggest a *list* of similar words that are "close" to the input word such as "in**t**erce**pt**" and "in**t**erc**ede**". This observation leads to the concept of *list-recoverable codes*. The basic idea of list-recoverable codes is that, given a list of subsets in which each subset contains at most ℓ symbols in a given alphabet Σ (a finite field), the decoder of the list-recoverable codes produces at most L codewords from the list. Formally, this can be defined as follows.

Definition 3 (Definition 2.2 [13]). *Given integers $1 \leq \ell \leq L$, a code $C \subseteq \Sigma^n$ is said to be (ℓ, L)-list-recoverable if for all sequences of subsets $S_1, S_2, \ldots, S_n$ with each $S_a \subset \Sigma$ satisfying $|S_a| \leq \ell$, there are at most L codewords $\mathbf{c} = (c_1, \ldots, c_n) \in C$ with the property that $c_a \in S_a$ for $a \in \{1, 2, \ldots, n\}$. The value ℓ is referred to as the input list size.*

Note that for any $\ell' \leq \ell$, an (ℓ, L)-list-recoverable code is

also an (ℓ', L)-list-recoverable code. For example, if we set $\Sigma = \{a, b, \ldots, z\}, \ell = 2, n = 9$, and $L = 2$, we have the following input and output:

$$\begin{bmatrix} S_1 = \{e, g\} \\ S_2 = \{r, x\} \\ S_3 = \{o, q\} \\ S_4 = \{t, u\} \\ S_5 = \{e, i\} \\ S_6 = \{s\} \\ S_7 = \{i, q\} \\ S_8 = \{t, u\} \\ S_9 = \{e\} \end{bmatrix} \xrightarrow{\text{decode}} \mathbf{c} = \left\{ \begin{bmatrix} e \\ x \\ q \\ u \\ i \\ s \\ i \\ t \\ e \end{bmatrix}, \begin{bmatrix} g \\ r \\ o \\ t \\ e \\ s \\ q \\ u \\ e \end{bmatrix} \right\}.$$

2.4 Reed-solomon Codes

We first review the concept of $(n, r, D)_q$ code C:

Definition 4. *Let n, r, D, q be positive integers. An $(n, r, D)_q$ code is a subset of Σ^n such that*

(1) Σ is a finite field and is called the alphabet of the code: $|\Sigma| = q$. Here we set $\Sigma = \mathbb{F}_q$.

(2) Each codeword is considered to be a vector of $\mathbb{F}_q^{n\times 1}$.

(3) $D = \min_{\mathbf{x},\mathbf{y}\in C} \Delta(\mathbf{x}, \mathbf{y})$, where $\Delta(\mathbf{x}, \mathbf{y})$ is the number of positions in which the corresponding entries of $\mathbf{x}$ and $\mathbf{y}$ differ.

(4) The cardinality of C, i.e., $|C|$, is at least q^r.

These parameters (n, r, D, q) are the block length, dimension, minimum distance, and alphabet size of C. If the minimum distance is not considered, we refer to C as $(n, r)_q$. Given a full-rank $n \times r$ matrix $\mathcal{G} \in \mathbb{F}_q^{n\times r}$, suppose that, for any $\mathbf{y} \in C$, there exists a message $\mathbf{x} \in \mathbb{F}_q^r$ such that $\mathbf{y} = \mathcal{G}\mathbf{x}$. In this case, C is called a linear code and denoted as $[n, r, D]_q$. Let $\mathcal{M}_C$ denote an $n \times q^r$ matrix in which the columns are the codewords in C.

Reed-Solomon (RS) codes are constructed by applying a polynomial method to a finite field $\mathbb{F}_q$. Here we review a common and widely used Reed-Solomon code, an $[n, r, D]_q$-code C in which $|C| = q^r$ and $D = n - r + 1$. Since D is determined from n and r, we refer to $[n, r, D]_q$-RS code as $[n, r]_q$-RS code. Guruswami [13] (Section 4.4.1) showed that any $[n, r]_q$-RS code is also an $\left(\left\lceil \frac{n}{r} \right\rceil - 1, O\left(\frac{n^4}{r^2}\right)\right)$-list-recoverable code. To efficiently decode RS code, Chowdhury et al. [7] proposed an efficient scheme, which they summarized in Table 1 of their paper with $\omega < 2.38$, as follows:

Theorem 4 (Corollary 18 [7])**.** *Let $1 \le r \le n \le q$ be integers. Then, any $[n, r]_q$-RS code, which is also $\left(\left\lceil \frac{n}{r} \right\rceil - 1, O\left(\frac{n^4}{r^2}\right)\right)$-list-recoverable code, can be decoded in time $O(n^{3.57} r^{2.69})$.*

A codeword of the $[n, r]_q$-RS code can be computed in time $O(r^2 \log\log r) \approx O(r^2)$ and space $O(r \log q / \log^2 r)$ [22].

2.5 Concatenated Codes

Concatenated codes C are constructed by using an $(n_1, k_1)_q$ *outer code* C_{out}, where $q = 2^{k_2}$ (in general, $q = p^{k_2}$ where p is a prime number), and an $(n_2, k_2)_2$ binary *inner code* C_{in}, denoted as $C = C_{\text{out}} \circ C_{\text{in}}$.

Given a message $\mathbf{m} \in \mathbb{F}_q^{k_1}$, let $C_{\text{out}}(\mathbf{m}) = (x_1, \ldots, x_{n_1}) \in \mathbb{F}_q^{n_1}$. Then $C_{\text{out}} \circ C_{\text{in}}(\mathbf{m}) = (C_{\text{in}}(x_1), C_{\text{in}}(x_2), \ldots, C_{\text{in}}(x_{n_1})) \in (\{0,1\}^{n_2})^{n_1}$. Note that C is an $(n_1 n_2, k_1 k_2)_2$ code.

Using a suitable outer code and a suitable inner code, d-disjunct matrices can be generated. For example, let C_{out} and C_{in} be $(3, 1)_8$ and $(3, 3)_2$ codes, where $|C_{\text{out}}| = 12$ and $|C_{\text{in}}| = 8$. The corresponding matrices are $\mathcal{H} = \mathcal{M}_{C_{\text{out}}}$ and $\mathcal{K} = \mathcal{M}_{C_{\text{in}}}$ as follows:

$$\mathcal{H} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 4 & 4 & 4 & 7 & 0 & 0 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 & 4 & 0 & 7 & 0 \\ 1 & 4 & 2 & 4 & 2 & 1 & 2 & 1 & 4 & 0 & 0 & 7 \end{bmatrix},$$

$$\mathcal{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix},$$

If we concatenate each element of $\mathcal{H}$ with its 3-bit binary representation such as matrix $\mathcal{K}$, we get a 2-disjunct matrix:

$$\mathcal{M} = \mathcal{H} \circ \mathcal{K}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

From this discussion, we can draw an important conclusion about decoding schemes using concatenation codes and list-recoverable codes.

Theorem 5 (Simplified version of Theorem 4.1 [16])**.** *Let $d, L \ge 1$ be integers. Let C_{out} be an $(n_1, k_1)_{2^{k_2}}$ code that can be (d, L)-list recovered in time $T_1(n_1, d, L, k_1, k_2)$. Let C_{in} be $(n_2, k_2)_2$ codes such that $\mathcal{M}_{C_{\text{in}}}$ is a d-disjunct matrix that can be decoded in time $T_2(n_2, d, k_2)$. Suppose that matrix $\mathcal{M} = \mathcal{M}_{C_{\text{out}}\circ C_{\text{in}}}$ is d-disjunct. Note that $\mathcal{M}$ is a $t \times N$ matrix where $t = n_1 n_2$ and $N = 2^{k_1 k_2}$. Further, suppose that any arbitrary position in any codeword in C_{out} and C_{in} can be computed in space $S_1(n_1, d, L, k_1, k_2)$ and $S_2(n_2, d, k_2)$, respectively. Then:*

(a) given any outcome produced by at most d positives, the positive positions can be recovered in time $n_1 T_2(n_2, d, k_2) + T_1(n_1, d, L, k_1, k_2) + 2Lt = n_1 T_2(n_2, d, k_2) + T_1(n_1, d, L, k_1, k_2) + O(Lt)$; and

(b) any entry in $\mathcal{M}$ can be computed in $\log t + \log N + S_1(n_1, d, L, k_1, k_2) + S_2(n_2, d, k_2) = O(\log t + \log N) + O(\max\{S_1(n_1, d, L, k_1, k_2), S_2(n_2, d, k_2)\})$ space.

Since the decoding scheme requires knowledge from several fields that are beyond the scope of this work, we do not discuss it here. Readers are encouraged to refer to Ref. [16] for further reading.

2.6 Review of Bui et al.'s Scheme

A scheme proposed by Bui et al. [2] plays an important role for constructions in later sections. It is used to identify at most one defective item while never producing a false positive. The technical details are as follows.

Encoding procedure: Lee et al. [18] proposed a $k \times N$ measurement matrix $\mathcal{S}$ that uses $\log N$-bit representation of an integer, to

detect at most one defective item:

$$\mathcal{S} := \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 \dots \mathbf{b}_N \\ \overline{\mathbf{b}}_1 & \overline{\mathbf{b}}_2 \dots \overline{\mathbf{b}}_N \end{bmatrix} = \begin{bmatrix} \mathcal{S}_1 \dots \mathcal{S}_N \end{bmatrix}, \tag{9}$$

where $k = 2\log N$, $\mathbf{b}_j$ is the $\log N$-bit binary representation of integer $j-1$, $\overline{\mathbf{b}}_j$ is $\mathbf{b}_j$'s complement, and $\mathcal{S}_j := \begin{bmatrix} \mathbf{b}_j \\ \overline{\mathbf{b}}_j \end{bmatrix}$ for $j = 1, 2, \dots, N$. The weight of every column in $\mathcal{S}$ is $k/2 = \log N$.

Given an input vector $\mathbf{g} = (g_1, \dots, g_N) \in \{0,1\}^N$, measurement matrix $\mathcal{S}$ is generalized:

$$\mathcal{B} := \mathcal{S} \times \mathrm{diag}(\mathbf{g}) = \begin{bmatrix} g_1\mathcal{S}_1 & \dots & g_N\mathcal{S}_N \end{bmatrix}, \tag{10}$$

where $\mathrm{diag}(\mathbf{g}) = \mathrm{diag}(g_1, \dots, g_N)$ is the diagonal matrix constructed by input vector $\mathbf{g}$, and $\mathcal{B}_j = g_j\mathcal{S}_j$ for $j = 1, \dots, N$. It is obvious that $\mathcal{B} = \mathcal{S}$ when $\mathbf{g}$ is a vector of all ones; i.e., $\mathbf{g} = \mathbf{1} = (1, 1, \dots, 1) \in \{1\}^N$. Moreover, the column weight of $\mathcal{B}$ is either $k/2 = \log N$ or 0.

For example, consider the case $N = 8, k = 2\log N = 6$, and $\mathbf{g} = (1,0,1,0,1,1,1,1)$. Measurement matrices S and $\mathcal{B}$ are

$$\mathcal{S} = \begin{bmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \\ 1&1&1&1&0&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \end{bmatrix}, \tag{11}$$

$$\begin{aligned} \mathcal{B} &= \mathcal{S} \times \mathrm{diag}(\mathbf{g}) = \mathcal{S} \times \mathrm{diag}(1,0,1,0,1,1,1,1) \\ &= [1\times\mathcal{S}_1 \ 0\times\mathcal{S}_2 \ 1\times\mathcal{S}_3 \ 0\times\mathcal{S}_4 \\ &\quad 1\times\mathcal{S}_5 \ 1\times\mathcal{S}_6 \ 1\times\mathcal{S}_7 \ 1\times\mathcal{S}_8] \\ &= \begin{bmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&0&0&0&1&1 \\ 0&0&0&0&0&1&0&1 \\ 1&0&1&0&0&0&0&0 \\ 1&0&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \end{bmatrix}. \end{aligned} \tag{12}$$

Then, given a representation vector of N items $\mathbf{x} = (x_1, \dots, x_N)^T \in \{0,1\}^N$, the outcome vector is

$$\mathbf{y}' = \mathcal{B} \otimes \mathbf{x} = \bigvee_{j=1}^{N} x_j\mathcal{B}_j \tag{13}$$

$$= \bigvee_{j=1}^{N} x_j g_j \mathcal{S}_j = \bigvee_{\substack{j=1 \\ x_j g_j = 1}}^{N} \mathcal{S}_j. \tag{14}$$

Note that, even if there is only one entry $x_{j_0} = 1$ in $\mathbf{x}$, index j_0 cannot be recovered if $g_{j_0} = 0$.

Decoding procedure: From Eq. (14), the outcome $\mathbf{y}'$ is the union of at most $|\mathbf{x}|$ columns in $\mathcal{S}$. Because the weight of each column in $\mathcal{S}$ is $\log N$, if the weight of $\mathbf{y}'$ is $\log N$, the index of one non-zero entry in $\mathbf{x}$ is recovered by checking the first half of $\mathbf{y}'$. On the other hand, if $\mathbf{y}'$ is the union of at least two columns in $\mathcal{S}$ or zero vector, the weight of $\mathbf{y}'$ is not equal to $\log N$. This case is considered here as a defective item is not identified. Therefore, given a $k\times 1$ input vector, we can either identify one defective item or no defective item in time $k = 2\log N = O(\log N)$. Moreover, the decoding procedure does not produce a false positive.

For example, given $\mathbf{x}_1 = (0,1,0,0,0,0,0,0)^T, \mathbf{x}_2 = (0,1,1,0,0,0,0,0)^T$, and $\mathbf{x}_3 = (0,1,1,1,0,0,0,0)^T$, their corresponding outcomes using the measurement matrix $\mathcal{B}$ in Eq. (12) are $\mathbf{y}'_1 = (0,0,0,0,0,0)^T, \mathbf{y}'_2 = (0,1,0,1,0,1)^T$, and $\mathbf{y}'_3 = (0,1,0,1,0,1)^T$. Since $|\mathbf{y}'_1| = 0$, there is no defective item identified. Since $|\mathbf{y}'_2| = |\mathbf{y}'_3| = 3 = \log N$, the only defective item identified from the first half of $\mathbf{y}'_2$ or $\mathbf{y}'_3$, i.e., $(0,1,0)$ is 3. Note that, even if $|\mathbf{x}_1| \neq |\mathbf{x}_2|$, the same defective item is identified.

3. Efficient Decoding Scheme Using a given Measurement Matrix

In this section, we present a simple but powerful tool for identifying defective items using a given measurement matrix. We thereby answer the question of whether there exists a scheme such that a larger $T \times N$ measurement matrix built from a given $t \times N$ measurement matrix, can be used to identify up to d defective items in time $\mathrm{poly}(t) = t \times \log N = T$. It can be summarized as follows:

Theorem 6. *For any $\epsilon \geq 0$, suppose each set of d columns in a given $t \times N$ matrix $\mathcal{M}$ contains a $d \times d$ identity matrix with probability at least $1-\epsilon$. Then there exists a $T \times N$ matrix $\mathcal{T}$ constructed from $\mathcal{M}$ that can be used to identify at most d defective items in time $T = t \times 2\log N$ with probability at least $1-\epsilon$. Further, suppose that any entry of $\mathcal{M}$ can be computed in time β and space γ, so every entry of $\mathcal{T}$ can be computed in time $O(\beta \log N)$ and space $O(\log T + \log N) + O(\gamma \log N)$.*

Proof. Suppose $\mathcal{M} = (m_{ij}) \in \{0,1\}^{t\times N}$. Then the $T \times N$ measurement matrix $\mathcal{T}$ is generated by using the tensor product of $\mathcal{M}$ and $\mathcal{S}$ in Eq. (9):

$$\begin{aligned} \mathcal{T} = \mathcal{M} \circledcirc \mathcal{S} &= \begin{bmatrix} \mathcal{S} \times \mathrm{diag}(\mathcal{M}_{1,*}) \\ \vdots \\ \mathcal{S} \times \mathrm{diag}(\mathcal{M}_{t,*}) \end{bmatrix} = \begin{bmatrix} \mathcal{B}^1 \\ \vdots \\ \mathcal{B}^t \end{bmatrix} \\ &= \begin{bmatrix} m_{11}\mathcal{S}_1 & \dots & m_{1N}\mathcal{S}_N \\ \vdots & \ddots & \vdots \\ m_{t1}\mathcal{S}_1 & \dots & m_{tN}\mathcal{S}_N \end{bmatrix}, \end{aligned} \tag{15}$$

where $T = t \times k = t \times 2\log N$ and $\mathcal{B}^i = \mathcal{S} \times \mathrm{diag}(\mathcal{M}_{i,*})$ for $i = 1, \dots, t$. Note that $\mathcal{B}^i$ is an instantiation of $\mathcal{B}$ when $\mathbf{g}$ is set to $\mathcal{M}_{i,*}$ in Eq. (10). Then, for any $N \times 1$ representation vector $\mathbf{x} = (x_1, \dots, x_N) \in \{0,1\}^N$, the outcome vector is

$$\mathbf{y}^\star = \mathcal{T} \otimes \mathbf{x} = \begin{bmatrix} \mathcal{B}^1 \otimes \mathbf{x} \\ \vdots \\ \mathcal{B}^t \otimes \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{y}'_1 \\ \vdots \\ \mathbf{y}'_t \end{bmatrix}, \tag{16}$$

where $\mathbf{y}'_i = \mathcal{B}^i \otimes \mathbf{x}$ for $i = 1, \dots, t$; $\mathbf{y}'_i$ is obtained by replacing $\mathcal{B}$ by $\mathcal{B}_i$ in Eq. (13).

By using the decoding procedure in Section 2.6, the decoding procedure is simply to scan all $\mathbf{y}'_i$ for $i = 1, \dots, t$. If $|\mathbf{y}'_i| = \log N$, we take the first half of $\mathbf{y}'_i$ to calculate the defective item. Thus, the decoding complexity is $T = t \times 2\log N = O(T)$.

Our task now is to prove that the decoding procedure above can identify all defective items with probability at least $1-\epsilon$. Let

$\mathbb{D} = \{j_1, \ldots, j_{|\mathbb{D}|}\}$ be the defective set, where $|\mathbb{D}| = g \leq d$. We will prove that there exists $\mathbf{y}'_{i_1}, \ldots, \mathbf{y}'_{i_g}$ such that j_a can be recovered from $\mathbf{y}'_{i_a}$ for $a = 1, \ldots, g$. Because any set of d columns in $\mathcal{M}$ contains a $d \times d$ identity matrix with probability at least $1 - \epsilon$, any set of $g \leq d$ columns $j_1, \ldots, j_g$ in $\mathcal{M}$ also contains a $g \times g$ identity matrix with probability at least $1 - \epsilon$. Let $i_1, \ldots, i_g$ be the row indexes of $\mathcal{M}$ such that $m_{i_a j_a} = 1$ and $m_{i_a j_b} = 0$, where $a, b \in \{1, 2, \ldots, g\}$ and $a \neq b$. Then the probability that rows $i_1, \ldots, i_g$ coexist is at least $1 - \epsilon$.

For any outcome $\mathbf{y}'_{i_a}$, where $a = 1, \ldots, g$, by using Eq. (14), we have

$$\mathbf{y}'_{i_a} = \mathcal{B}^{i_a} \otimes \mathbf{x} = \bigvee_{\substack{j=1 \\ x_j m_{i_a j}=1}}^{N} \mathcal{S}_j = \bigvee_{\substack{j \in \mathbb{D} \\ x_j m_{i_a j}=1}} \mathcal{S}_j = \mathcal{S}_{j_a}, \tag{17}$$

because there are only g non-zero entries $x_{j_1}, \ldots, x_{j_g}$ in $\mathbf{x}$. Thus, all defective items $j_1, \ldots, j_g$ can be identified by checking the first half of each corresponding $\mathbf{y}'_{i_1}, \ldots, \mathbf{y}'_{i_g}$. Since the probability that rows $i_1, \ldots, i_g$ coexist is at least $1 - \epsilon$, the probability that defective items $j_1, \ldots, j_g$ are identified is also at least $1 - \epsilon$.

We next estimate the computational complexity of computing an entry in $\mathcal{T}$. An entry in row $1 \leq i \leq T$ and column $1 \leq j \leq N$ needs $\log T + \log N$ bits (space) to be indexed. It belongs to vector $m_{i_0 j}\mathcal{S}_j$, where $i_0 = i/(2 \log N)$ if $i \mod (2 \log N) \equiv 0$ and $i_0 = \lfloor i/(2 \log N) \rfloor$ if $i \mod (2 \log N) \not\equiv 0$. Since each entry in $\mathcal{M}$ needs γ space to compute, every entry in $\mathcal{T}$ can be computed in space $O(\log T + \log N) + O(\gamma \log N)$ after mapping it to the corresponding column of $\mathcal{S}$. The time to generate an entry for $\mathcal{T}$ is straightforwardly obtained as $\beta \log N = O(\beta \log N)$. □

Part of Theorem 6 is implicit in other papers (e.g., Ref. [2], [4], [5], [18]). However, the authors of those papers only considered cases specific to their problems. They mainly focused on how to generate matrix $\mathcal{M}$ by using complicated techniques and a non-constructive method, i.e., random construction (e.g., Ref. [5], [18]). As a result, their decoding schemes are randomized. Moreover, they did not consider the cost of computing an entry in $\mathcal{M}$. In two of the papers [2], [4], the decoding time was not scaled to $t \times \log N$ for deterministic decoding, i.e., $\epsilon = 0$. Our contribution is to *generalize* their ideas into the framework of non-adaptive group testing. We next instantiate Theorem 6 in the broad range of measurement matrix construction.

3.1 Case of $\epsilon = 0$

We consider the case in which $\epsilon = 0$; i.e., a given matrix $\mathcal{M}$ is always $(d-1)$-disjunct. There are three metrics for evaluating an instantiation: number of tests, construction type, and time to generate an entry for $\mathcal{T}$. We first present an instantiation of a strongly explicit construction. Let $\mathcal{M}$ be a measurement matrix generated from Theorem 2. Then $t = O(d^2 \log N)$, $\beta = O(t^2 \log N)$, and $\gamma = O(t \log N)$. Thus, we obtain efficient NAGT where the number of tests and the decoding time are $O(d^2 \log^2 N)$.

Corollary 1. *Let $1 \leq d \leq N$ be integers. There exists a random $T \times N$ measurement matrix $\mathcal{T}$ with $T = O(d^2 \log^2 N)$ such that at most d defective items can be identified in time $O(T)$. Moreover, each entry in $\mathcal{T}$ can be computed in time $O(T^2)$ and space $O(T \log N)$.*

It is also possible to construct $\mathcal{T}$ deterministically. However, it would take poly(t, N) time and poly(t) space, which are too long and too much for practical applications. Therefore, we should increase the time needed to generate an entry for $\mathcal{T}$ in order to achieve nonrandom construction with the same number of tests $T = O(d^2 \log^2 N)$ and a short construction time. The following theorem is based on the weakly explicit construction of a given measurement matrix as in Theorem 1; i.e., $t = O(d^2 \log N)$, $\beta = O(tN)$, and $\gamma = O(tN)$.

Corollary 2. *Let $1 \leq d \leq N$ be integers. There exists a nonrandom $T \times N$ measurement matrix $\mathcal{T}$ with $T = O(d^2 \log^2 N)$ that can be used to identify at most d defective items in time $O(T)$. Moreover, each entry in $\mathcal{T}$ can be computed in time (and space) $O(TN)$.*

Although the number of tests is low and the construction type is nonrandom, the time to generate an entry for $\mathcal{T}$ is long. If we increase the number of tests, one can achieve both nonrandom construction and low generating time for an entry as follows:

Corollary 3. *Let $1 \leq d \leq N$ be integers. There exists a nonrandom $T \times N$ measurement matrix $\mathcal{T}$ with $T = O\left(\frac{d^2 \log^3 N}{(\log(d \log N) - \log \log(d \log N))^2}\right)$ that can be used to identify at most d defective items in time $O(T)$. Moreover, each entry in $\mathcal{T}$ can be computed in time (and space) $O(T)$.*

The above corollary is obtained by choosing a measurement matrix as a d-disjunct matrix in Theorem 8 (Section 4): $t = O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log \log(d \log N))^2}\right)$, $\beta = O(t)$, and $\gamma = O(t)$.

3.2 Case of $\epsilon > 0$

To reduce the number of tests and the decoding complexity, the construction process of the given measurement matrix must be randomized. We construct the matrix as follows. A given $t \times N$ matrix $\mathcal{M} = (m_{ij})$ is generated randomly, where $\Pr(m_{ij} = 1) = \frac{1}{d}$ and $\Pr(m_{ij} = 0) = 1 - \frac{1}{d}$ for $i = 1, \ldots, t$ and $j = 1, \ldots, N$. The value of t is set to $\mathrm{e}d \ln \frac{d}{\epsilon}$. Then, for each set of d columns in $\mathcal{M}$, the probability that a set does not contain a $d \times d$ identity matrix is at most

$$\binom{d}{1}\left(1 - \frac{1}{d}\left(1 - \frac{1}{d}\right)^{d-1}\right)^t \tag{18}$$

$$\leq d \cdot \exp\left(-\frac{1}{d-1}\left(1 - \frac{1}{d}\right)^d t\right) \tag{19}$$

$$\leq d \cdot \exp\left(-\frac{t}{d-1} \cdot \mathrm{e}^{-1}\left(1 - \frac{1}{d}\right)\right) \tag{20}$$

$$\leq d \cdot \exp\left(-\frac{t}{\mathrm{e}d}\right) = d \cdot \exp\left(-\ln \frac{d}{\epsilon}\right) \tag{21}$$

$$\leq \epsilon. \tag{22}$$

Expression (19) is obtained because $(1 + x)^y \leq \exp(xy)$ for all $|x| \leq 1$ and $y \geq 1$. Expression (20) is obtained because $\left(1 + \frac{x}{n}\right)^n \geq \mathrm{e}^x\left(1 - \frac{x^2}{n}\right)$ for $n > 1$ and $|x| < n$. Therefore, there exists a $t \times N$ matrix $\mathcal{M}$ with $t = O\left(d \log \frac{d}{\epsilon}\right)$ such that each set of d columns contains a $d \times d$ identity matrix with probability at least $1 - \epsilon$, for any $\epsilon > 0$. Since $\beta = \gamma = O(tN)$, W can derive the following corollary.

Corollary 4. *Given integers $1 \leq d \leq N$ and a scalar $\epsilon > 0$, there exists a random $T \times N$ measurement matrix $\mathcal{T}$ with $T =$*

$O\left(d \log N \cdot \log \frac{d}{\epsilon}\right)$ *that can be used to identify at most* d *defective items in time* $O(T)$ *with probability at least* $1-\epsilon$. *Furthermore, each entry in* $\mathcal{T}$ *can be computed in time (and space)* $O(TN)$.

While the result in Corollary 4 is similar to previously reported ones [5], [18], construction of matrix $\mathcal{M}$ is much simpler. It is possible to achieve the number of tests $t = O\left(d \log \frac{d}{\epsilon} \cdot \log N\right)$ when *each* set of d columns in $\mathcal{M}$ contains a $d \times d$ identity matrix with probability at least $1-\epsilon$ for any $\epsilon > 0$. However, it is impossible to achieve this number for *every* set of d columns that contains a $d \times d$ identity matrix with probability at least $1-\epsilon$. In this case, by using the same procedure used for generating random matrix $\mathcal{M}$ and by resolving $\binom{N}{d}\binom{d}{1}\left(1-\frac{1}{d}\left(1-\frac{1}{d}\right)^{d-1}\right)^t \le \epsilon$, the number of tests needed is determined to be $t = O\left(d^2 \log N + d \log \frac{1}{\epsilon}\right)$. Since this number is greater than that when $\epsilon = 0$ ($O(d^2 \log N)$), it is not beneficial to consider the case that every set of d columns that contains a $d \times d$ identity matrix with probability at least $1-\epsilon$.

4. Nonrandom Disjunct Matrices

It is extremely important to have nonrandom constructions for measurement matrices in real-time applications. Therefore, we now focus on nonrandom constructions. We have shown that the well-known barrier on the number of tests $O(d^2 \log^2 N)$ for constructing a d-disjunct matrix can be overcome.

4.1 Case of $d = 2$

When $d = 2$, the measurement matrix is $\mathcal{T} = \mathcal{S} \circledcirc \mathcal{S}$, where $\mathcal{S}$ is given by Eq. (9). Note that the size of $\mathcal{S}$ is $k \times N$, where $k = 2 \log N$, and $\mathcal{T}$ is not a 2-disjunct matrix. We start by proving that any two columns in $\mathcal{S}$ contain a 2×2 identity matrix. Indeed, suppose $\mathbf{b}_w = (b_{1w}, \ldots, b_{(k/2)w})^T$, which is a $\log N$-bit binary representation of $0 \le w-1 \le N-1$. For any two vectors $\mathbf{b}_{w_1}$ and $\mathbf{b}_{w_2}$, there exists a position i_0 such that $b_{i_0 w_1} = 0$ and $b_{i_0 w_2} = 1$, or $b_{i_0 w_1} = 1$ and $b_{i_0 w_2} = 0$ for any $1 \le w_1 \ne w_2 \le N$. Then their corresponding complementary vectors $\overline{\mathbf{b}}_{w_1} = (\overline{b}_{1w_1}, \ldots, \overline{b}_{(k/2)w_1})^T$ and $\overline{\mathbf{b}}_{w_2} = (\overline{b}_{1w_2}, \ldots, \overline{b}_{(k/2)w_2})^T$ satisfy: $\overline{b}_{i_0 w_1} = 0$ and $\overline{b}_{i_0 w_2} = 1$ when $b_{i_0 w_1} = 0$ and $b_{i_0 w_2} = 1$, or $\overline{b}_{i_0 w_1} = 1$ and $\overline{b}_{i_0 w_2} = 0$ when $b_{i_0 w_1} = 1$ and $b_{i_0 w_2} = 0$. Thus, any two columns w_1 and w_2 in $\mathcal{S}$ always contain a 2×2 identity matrix. From Theorem 6 (set $\mathcal{M} = \mathcal{S}$), we obtain the following theorem.

Theorem 7. *Let* $2 \le N$ *be an integer. A* $4 \log^2 N \times N$ *nonrandom measurement matrix* $\mathcal{T}$ *can be used to identify at most two defective items in time* $4 \log^2 N$. *Moreover, each entry in* $\mathcal{T}$ *can be computed in space* $2 \log N + \log(2 \log N)$ *with four operations.*

Proof. It takes $\gamma = 2 \log N + \log(2 \log N)$ bits to index an entry in row i and column j. Only two shift operations and a mod operation are needed to exactly locate the position of the entry in column $\mathcal{S}_j$. Therefore, at most four operations ($\beta = 4$) and $2 \log N + \log(2 \log N)$ bits are needed to locate an entry in matrix $\mathcal{T}$. The decoding time is straightforwardly obtained from Theorem 6 ($t = k = 2 \log N$). □

4.2 General Case

Indyk et al. [16] used Theorem 5 and Parvaresh-Vardy (PV) codes [20] to come up with Theorem 3. Since they wanted to convert RS code into list-recoverable code, they instantiated PV code into RS code. However, because PV code is powerful in terms of solving general problems, its decoding complexity is high. Therefore, the decoding complexity in Theorem 3 is relatively high. Here, by converting RS code into list-recoverable code using Theorem 4, we carefully use Theorem 5 to construct and decode disjunct matrices. As a result, the number of tests and the decoding time for a nonrandom disjunct matrix are significantly reduced.

Let $W(x)$ be a Lambert W function in which $W(x)\mathrm{e}^{W(x)} = x$ for any $x \ge -\frac{1}{\mathrm{e}}$. When $x > 0$, $W(x)$ is an increasing function. One useful bound [15] for a Lambert W function is $\ln x - \ln\ln x \le W(x) \le \ln x - \frac{1}{2}\ln\ln x$ for any $x \ge \mathrm{e}$. Theorem 5 is used to achieve the following theorem with careful setting of C_{out} and C_{in}

Theorem 8. *Let* $1 \le d \le N$ *be integers. Then there exists a nonrandom* d*-disjunct matrix* $\mathcal{M}$ *with* $t = O\left(\frac{d^2 \ln^2 N}{(W(d \ln N))^2}\right) = O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log\log(d \log N))^2}\right)$. *Each entry (column) in* $\mathcal{M}$ *can be computed in time (and space)* $O(t)$ ($O(t^{3/2})$). *Moreover,* $\mathcal{M}$ *can be used to identify up to* d' *defective items, where* $d' \ge \left\lfloor \frac{d}{2} \right\rfloor + 1$, *in time*

$$O\left(\frac{d^{3.57} \log^{6.26} N}{(\log(d \log N) - \log\log(d \log N))^{6.26}}\right) + O\left(\frac{d^6 \log^4 N}{(\log(d \log N) - \log\log(d \log N))^4}\right).$$

When d *is the power of 2,* $d' = d - 1$.

Proof. **Construction:** We use the classical method proposed by Kautz and Singleton [17] to construct a d-disjunct matrix. Let η be an integer satisfying $2^\eta < 2\mathrm{e}^{W(\frac{1}{2} d \ln N)} < 2^{\eta+1}$. Choose C_{out} as an $[n = q-1, r]_q$-RS code, where

$$q = \begin{cases} 2\mathrm{e}^{W(\frac{1}{2} d \ln N)} = \frac{d \ln N}{W(\frac{1}{2} d \ln N)} & \text{if } 2\mathrm{e}^{W(\frac{1}{2} d \ln N)} \text{ is the power of 2.} \\ 2^{\eta+1}, & \text{otherwise.} \end{cases} \tag{23}$$

Set $r = \left\lceil \frac{q-2}{d} \right\rceil$, and let C_{in} be a $q \times q$ identity matrix. The complexity of q is $\Theta\left(\mathrm{e}^{W(d \ln N)}\right) = \Theta\left(\frac{d \ln N}{W(d \ln N)}\right)$ in both cases because

$$2\mathrm{e}^{W(\frac{1}{2} d \ln N)} = \frac{d \ln N}{W\left(\frac{1}{2} d \ln N\right)} \le q < 2 \cdot 2\mathrm{e}^{W(\frac{1}{2} d \ln N)} = \frac{2d \ln N}{W\left(\frac{1}{2} d \ln N\right)}.$$

Let $C = C_{\text{out}} \circ C_{\text{in}}$. We are going to prove that $\mathcal{M} = \mathcal{M}_C$ is d-disjunct for such q and r. It is well known [17] that if $d \le \frac{q-1-1}{r-1}$, $\mathcal{M}$ is d-disjunct with $t = q(q-1)$ tests. Indeed, we have

$$\frac{q-1-1}{r-1} = \frac{q-2}{\lceil \frac{q-2}{d} \rceil - 1} \ge \frac{q-2}{\frac{q-2}{d} + 1 - 1} = d. \tag{24}$$

Since $q = O\left(\frac{d \ln N}{W(d \ln N)}\right)$, the number of tests in $\mathcal{M}$ is

$$t = q(q-1) = O\left(\frac{d^2 \ln^2 N}{(W(d \ln N))^2}\right)$$

$$= O\left(\frac{d^2 \ln^2 N}{(\ln(d \ln N) - \ln\ln(d \ln N))^2}\right)$$
$$= O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log\log(d \log N))^2}\right),$$

because $\ln x - \ln\ln x \le W(x) \le \ln x - \frac{1}{2}\ln\ln x$ for any $x \ge \mathrm{e}$. Since C_{out} is an $[n, r]_q$-RS code, each of its codewords can be computed [22] in time

$$O(r^2) = O\left(\left(\frac{\ln N}{\ln(d \ln N) - \ln\ln(d \ln N)}\right)^2\right)$$
$$= O\left(\frac{t}{d^2}\right) = O(t),$$

and space

$$S_1 = O(r \log q / \log^2 r)$$
$$= O(q \log q) = O(d \ln N) = O(t). \quad (25)$$

Our task is now to prove that the number of columns in $\mathcal{M}_C$, i.e., q^r, is at least N. The range of $\frac{d \ln N}{W(\frac{1}{2} d \ln N)} \le q < \frac{2d \ln N}{W(\frac{1}{2} d \ln N)}$ is:

$$d + 2 < \frac{d \ln N}{\ln\left(\frac{1}{2} d \ln N\right) - \frac{1}{2}\ln\ln\left(\frac{1}{2} d \ln N\right)} \le q \quad (26)$$

$$q \le \frac{2d \ln N}{\ln\left(\frac{1}{2} d \ln N\right) - \ln\ln\left(\frac{1}{2} d \ln N\right)} < 2d \ln N. \quad (27)$$

These inequalities were obtained because $\ln x - \ln\ln x \le W(x) \le \ln x - \frac{1}{2}\ln\ln x$ for any $x \ge \mathrm{e}$. Then we have:

$$q^{(q-2)/d} = \left(\frac{q^q}{q^2}\right)^{1/d}$$
$$\ge \left(\frac{1}{q^2} \times \left(2\mathrm{e}^{W(\frac{1}{2} d \ln N)}\right)^q\right)^{1/d}$$
$$\ge \left(\frac{2^q}{q^2} \times \left(\mathrm{e}^{W(\frac{1}{2} d \ln N)}\right)^q\right)^{1/d}$$
$$\ge \left(\frac{2^q}{q^2} \times \left(\mathrm{e}^{W(\frac{1}{2} d \ln N) \times 2\mathrm{e}^{W(\frac{1}{2} d \ln N)}}\right)\right)^{1/d}$$
$$\ge \left(\frac{2^q}{q^2} \times \mathrm{e}^{2 \times \frac{1}{2} d \ln N}\right)^{1/d} \quad (28)$$
$$\ge N \times \left(\frac{2^q}{q^2}\right)^{1/d} > N. \quad (29)$$

Equation (28) is achieved because $W(x)\mathrm{e}^{W(x)} = x$. Equation (29) is obtained because $\left(\frac{2^q}{q^2}\right)^{1/d} \ge 1$ for any $q \ge 5$. Since $\frac{q-2}{d} \le r = \lceil \frac{q-2}{d} \rceil < \frac{q-2}{d} + 1$, the number of codewords in C_{out} is:

$$N < q^{(q-2)/d} \le q^r < q^{(q-2)/d+1} = q \times q^{(q-2)/d} \quad (30)$$
$$< \frac{d \ln N}{W\left(\frac{1}{2} d \ln N\right)} \left(\frac{2^q}{q^2}\right)^{1/d} \times N. \quad (31)$$

Equation (30) indicates that the number of columns in $\mathcal{M}_C$ is more than N. To obtain a $t \times N$ matrix, one simply removes $q^r - N$ columns from $\mathcal{M}_C$. The maximum number of columns that can be removed is $O(d \ln N \times N^2)$ because of Eq. (31).

Decoding: Consider the ratio $\frac{q-1}{r}$ implied by list size $d' = \left\lceil \frac{q-1}{r} \right\rceil - 1 = \left\lceil \frac{q-1}{\lceil (q-2)/d \rceil} \right\rceil - 1$ of $[q-1, r]_q$-RS code. Parameter d' is also the maximum number of defective items that $\mathcal{M}$ can be used to identify because of Theorem 5. We thus have

$$d' = \left\lceil \frac{q-1}{\lceil (q-2)/d \rceil} \right\rceil - 1 \ge d\left(1 - \frac{d-1}{q+d-2}\right) > \frac{d}{2},$$

because $q + d - 2 \ge 2d > 2(d-1)$. Since d' is an integer, $d' \ge \left\lfloor \frac{d}{2} \right\rfloor + 1$.

Next we prove that $d' = d - 1$ when d is the power of 2, e.g., $d = 2^x$ for some positive integer x. Since q is also the power of 2 as shown by Eq. (23), suppose that $q = 2^y$ for some positive integer y. Because $q > d$ in Eq. (26), $2^y > 2^x$. Then $r = \lceil \frac{q-1}{d} \rceil = 2^{y-x}$. Therefore, $d' = \left\lceil \frac{q-1}{r} \right\rceil - 1 = 2^x - 1 = d - 1$.

The decoding complexity of our proposed scheme is analyzed here. We have:

- Code C_{out} is an $(d' = \left\lceil \frac{q-1}{\lceil (q-2)/d \rceil} \right\rceil - 1, L = O\left(\frac{n^4}{r^2}\right) = O(q^2 d^2))$-list recoverable code as in Theorem 4. It can be decoded in time

  $$T_1 = O(n^{3.57} r^{2.69})$$
  $$= O\left(\frac{d^{3.57} \log^{6.26} N}{(\log(d \log N) - \log\log(d \log N))^{6.26}}\right).$$

  Moreover, any codeword in C_{out} can be computed in time $O(r^2) = O\left(\frac{t}{d^2}\right)$ and space $S_1 = O(t)$ as in Eq. (25).
- C_{in} is a $q \times q$ identity matrix. Then $\mathcal{M}_{C_{\text{in}}}$ is a q-disjunct matrix. Since $d' \le d < q$, $\mathcal{M}_{C_{\text{in}}}$ is also a d'-disjunct matrix. It can be decoded in time $T_2 = d'q$ and each codeword can be computed in space $S_2 = \log q$.

From Theorem 5, given any outcome produced by at most d' defective items, those items can be identified in time

$$T_s = nT_2 + T_1 + O(Lt)$$
$$= nd'q + O\left(\frac{d^{3.57} \log^{6.26} N}{(\log(d \log N) - \log\log(d \log N))^{6.26}}\right)$$
$$+ O\left(\frac{d^6 \log^4 N}{(\log(d \log N) - \log\log(d \log N))^4}\right)$$
$$= O\left(\frac{d^{3.57} \log^{6.26} N}{(\log(d \log N) - \log\log(d \log N))^{6.26}}\right)$$
$$+ O\left(\frac{d^6 \log^4 N}{(\log(d \log N) - \log\log(d \log N))^4}\right). \quad (32)$$

Moreover, each entry (column) in $\mathcal{M}$ can be computed in time $O(t)$ ($O(tq) = O(t^{3/2})$) and space $O(\log t + \log N) + O(\max\{S_1, S_2\}) = O(d \log N) = O(t)$ ($O(tq) = O(t^{3/2})$). □

If we substitute d by $2^{\lfloor \log_2 d \rfloor + 1}$ in the theorem above, the measurement matrix is $2^{\lfloor \log_2 d \rfloor + 1}$-disjunct. Therefore, it can be used to identify at most $d' = 2^{\lfloor \log_2 d \rfloor + 1} - 1 \ge d$ defective items. The number of tests and the decoding complexity in the theorem remain unchanged because $d < 2^{\lfloor \log_2 d \rfloor + 1} \le 2d$.

5. Evaluation

We evaluated variations of our proposed scheme by simulation using $d = 2, 2^3, 2^7, 2^{10}, 2^{12}$ and $N = 2^{20}, 2^{40}, 2^{60}, 2^{80}, 2^{100}$ in Matlab R2015a on an HP Compaq Pro 8300SF desktop PC with a 3.4-GHz Intel Core i7-3770 processor and 16-GB memory.

Table 2 Parameter settings for $[q-1, r]_q$-RS code and resulting $q(q-1) \times N$ d-disjunct matrix: number of items N, maximum number of defective items d, alphabet size q as in Eq. (23), number of tests $t = q(q-1)$, dimension $r = \lceil \frac{q-2}{d} \rceil$. Parameter $d' = \left\lceil \frac{q-1}{\lceil (q-2)/d \rceil} \right\rceil - 1$ is the maximum number of defective items that the $t \times N$ resulting matrix can be used to identify. Parameter $N' = q^r$ is maximum number of items such that resulting $q(q-1) \times N'$ matrix generated from this RS code is still d-disjunct. Parameters $t_2 = 4800d^2 \log N$ and $t_1 = d \log N(d \log N - 1)$ are number of tests from Theorems 2 and 3.

d	N	q	$t = q(q-1)$	r	d'	N'	$t_1 = d \log N(d \log N - 1)$	$t_2 = 4800d^2 \log N$
$2^3 = 8$	2^{20}	$2^6 = 64$	4,032	8	$d-1$	2^{48}	25,440	6,144,000
	2^{40}	$2^7 = 128$	16,256	16	$d-1$	2^{102}	102,080	12,288,000
	2^{60}	$2^7 = 128$	16,256	16	$d-1$	2^{102}	229,920	18,432,000
	2^{80}	$2^7 = 128$	16,256	16	$d-1$	2^{102}	408,960	24,576,000
	2^{100}	$2^8 = 256$	65,280	32	$d-1$	2^{256}	639,200	30,720,000
$2^7 = 128$	2^{20}	$2^9 = 512$	261,632	4	$d-1$	2^{36}	6,551,040	1,572,864,000
	2^{40}	$2^{10} = 1{,}024$	1,047,552	8	$d-1$	2^{80}	26,209,280	3,145,728,000
	2^{60}	$2^{10} = 1{,}024$	1,047,552	8	$d-1$	2^{80}	58,974,720	4,718,592,000
	2^{80}	$2^{11} = 2{,}048$	4,192,256	16	$d-1$	2^{176}	104,847,360	6,291,456,000
	2^{100}	$2^{11} = 2{,}048$	4,192,256	16	$d-1$	2^{176}	163,827,200	7,864,320,000
$2^{10} = 1{,}024$	2^{20}	$2^{11} = 2{,}048$	4,192,256	2	$d-1$	2^{22}	419,409,920	100,663,296,000
	2^{40}	$2^{12} = 4{,}096$	16,773,120	4	$d-1$	2^{48}	1,677,680,640	201,326,592,000
	2^{60}	$2^{13} = 8{,}192$	67,100,672	8	$d-1$	2^{104}	3,774,812,160	301,989,888,000
	2^{80}	$2^{13} = 8{,}192$	67,100,672	8	$d-1$	2^{104}	6,710,804,480	402,653,184,000
	2^{100}	$2^{14} = 16{,}384$	268,419,072	16	$d-1$	2^{224}	10,485,657,600	503,316,480,000
$2^{12} = 4{,}096$	2^{20}	$2^{13} = 8{,}192$	67,100,672	2	$d-1$	2^{26}	6,710,804,480	1,610,612,736,000
	2^{40}	$2^{14} = 16{,}384$	268,419,072	4	$d-1$	2^{56}	26,843,381,760	3,221,225,472,000
	2^{60}	$2^{15} = 32{,}768$	1,072,398,336	8	$d-1$	2^{120}	60,397,731,840	4,831,838,208,000
	2^{80}	$2^{15} = 32{,}768$	1,072,398,336	8	$d-1$	2^{120}	107,373,854,720	6,442,450,944,000
	2^{100}	$2^{15} = 32{,}768$	1,072,398,336	8	$d-1$	2^{120}	167,771,750,400	8,053,063,680,000

5.1 Numerical Settings for N, d, and q

We focused on nonrandom construction of a $t \times N$ d-disjunct matrix $\mathcal{M}$ for which the time to generate an entry is poly(t). Given integers d and N, an $[n = q-1, r]_q$ code C_{out} and a $q \times q$ identity matrix C_{in} were set up to create $\mathcal{M} = \mathcal{M}_{C_{\text{out}} \circ C_{\text{in}}}$. The precise formulas for q, r, t are $q = 2e^{W(\frac{1}{2} d \ln N)}$ or $q = 2^{\eta+1}$ as in Eq. (23), $r = \lceil \frac{q-2}{d} \rceil$, and $t = q(q-1)$. Note that the integer q is the power of 2. Moreover, $N' = q^r$ is the maximum number of items such that the resulting $t \times N'$ matrix generated from this RS code was still d-disjunct. Parameter $d' = \left\lceil \frac{q-1}{r} \right\rceil - 1 = \left\lceil \frac{q-1}{\lceil (q-2)/d \rceil} \right\rceil - 1$ is the maximum number of defective items that matrix $\mathcal{M}$ could be used to identify. The parameters $t_2 = 4800d^2 \log N$ and $t_1 = d \log N(d \log N - 1)$ are the number of tests from Theorems 2 and 3. The numerical results are shown in **Table 2**.

Since the number of tests from Theorem 2 is $O(d^2 \log N)$, it *should* be smaller than the number of tests in Theorem 3, which is $t = O(d^2 \log^2 N)$, and Theorem 8, which is $t = O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log\log(d \log N))^2}\right)$. However, the numerical results in Table 2 show the opposite. Even when $d = 2^{12} \approx 0.4\%$ of N, the number of tests from Theorem 2 was the largest. Moreover, there was no efficient construction scheme associated with it. The main reason is that the multiplicity of $O(d^2 \log N)$ is 4,800, which is quite large. **Figure 1** shows the ratio between the number of tests from Theorem 2 and the number from Theorem 8 (our proposed scheme) and between the number from Theorem 3 and the number from Theorem 8 (our proposed scheme). The number of tests with our proposed scheme was clearly smaller than with the existing schemes, even when $N = 2^{100}$. This indicates that the matrices generated from Theorem 2 and Theorem 3 are *good in theoretical analysis* but *bad in practice*.

In contrast, a nonrandom d-disjunct matrix is easily generated from Theorem 8. It also can be used to identify at most $d-1$ defective items. If we want to identify up to d defective items, we must generate a nonrandom $(d+1)$-disjunct matrix in which the number of tests is still smaller than t_1 and t_2. Since the number of tests from Theorem 8 is the lowest, its decoding time is the shortest. In short, for implementation, we recommend using the nonrandom construction in Theorem 8.

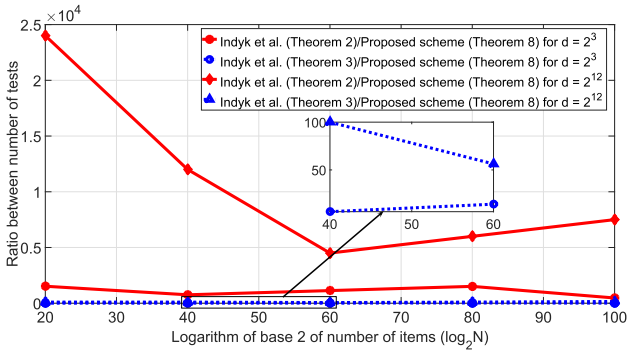

Fig. 1 Ratio of number of tests from Theorem 2 and number from Theorem 3 to number with proposed scheme (Theorem 8) for $d = 2^3, 2^{12}$ and $N = 2^{20}, 2^{40}, 2^{60}, 2^{80}, 2^{100}$. Ratio was always larger than 1; i.e., the number of tests in the proposed scheme is smaller than the compared one.

5.2 Experimental Results

Since the time to generate a measurement matrix entry would be too long if it were $O(tN)$, we focus on implementing the methods for which the time to generate a measurement matrix entry is poly(t), i.e., $\langle \mathbf{3} \rangle, \langle \mathbf{4} \rangle, \langle \mathbf{8} \rangle, \langle 9 \rangle, \langle \mathbf{10} \rangle$ in Table 1. However, to incorporate a measurement matrix into applications, random constructions are not preferable. Therefore, we focus on nonrandom constructions. Since we are unable to program decoding of list-recoverable codes because it requires knowledge of algebra, finite field, linear algebra, and probability. We therefore tested our pro-

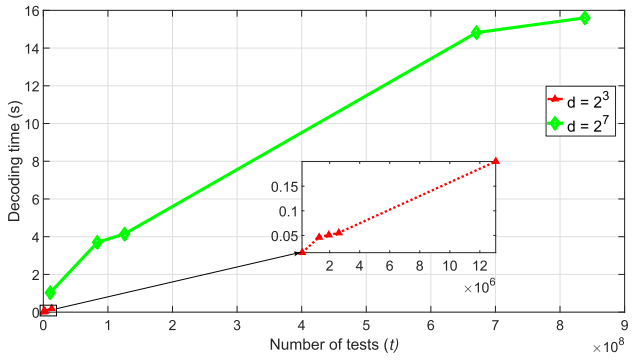

Fig. 2 Decoding time for $d = 2^3$ and $d = 2^7$ from Theorem 7. Number of items N was $\{2^{20}, 2^{40}, 2^{60}, 2^{80}, \text{or } 2^{100}\}$.

posed scheme by implementing ⟨**4**⟩ (Theorem 7) and ⟨**8**⟩ (Corollary 3). This is reasonable because, as analyzed in Section 5.1, the number of tests in Theorem 8 is the best for implementing nonrandom constructions. Since Corollary 3 is derived from Theorem 8, its decoding time should be the best for implementation.

We ran experiments for $d = 2$ from Theorem 7 and $d = 2^3, 2^7$ from Corollary 3. We did not run any for $d = 2^{10}, 2^{12}$ because there was not enough memory in our set up (more than 100 GB of RAM is needed). The decoding time was calculated in seconds and averaged over 100 runs. When $d = 2$, the decoding time was less than 1 ms. As shown in **Fig. 2**, the decoding time was linearly related to the number of tests, which confirms our theoretical analysis. Moreover, defective items were identified extremely quickly (less than 16 s) even when $N = 2^{100}$. The accuracy was always 1; i.e., all defective items were identified.

6. Conclusion

We have presented a scheme that enables a larger measurement matrix built from a given $t \times N$ measurement matrix to be decoded in time $O(t \log N)$ and a construction of a nonrandom d-disjunct matrix with $t = O\left(\frac{d^2 \log^2 N}{(\log(d \log N) - \log \log(d \log N))^2}\right)$ tests. This number of tests indicates that the upper bound for nonrandom construction is no longer $O(d^2 \log^2 N)$. Although the number of tests with our proposed schemes is not optimal in term of theoretical analysis, it is good enough for implementation. In particular, the decoding time is less than 16 seconds even when $d = 2^7 = 128$ and $N = 2^{100}$. Moreover, in nonrandom constructions, there is no need to store a measurement matrix because each column in the matrix can be generated efficiently.

Open problem: Our finding that N becomes much smaller than N' as q increases (Table 2) is quite interesting. Our hypothesis is that the number of tests needed may be smaller than $2e^{W(\frac{1}{2}d \ln N)}\left(2e^{W(\frac{1}{2}d \ln N)} - 1\right)$. If this is indeed true, it paves the way toward achieving a very efficient construction and a shorter decoding time without using randomness. An interesting question is to answer the question that whether there exists a $t \times N$ d-disjunct matrix with $t \leq 2e^{W(\frac{1}{2}d \ln N)}\left(2e^{W(\frac{1}{2}d \ln N)} - 1\right)$ that can be constructed in time $O(tN)$ with each entry generated in time (and space) poly(t) and with a decoding time of $O(t^2)$.

Acknowledgments The first author would like to thank Dr. Mahdi Cheraghchi, Imperial College London, UK for his fruitful discussions.

References

[1] Atia, G.K. and Saligrama, V.: Boolean compressed sensing and noisy group testing, *IEEE Trans. Information Theory*, Vol.58, No.3, pp.1880–1901 (2012).
[2] Bui, T.V., Kojima, T., Kuribayashi, M. and Echizen, I.: Efficient Decoding Schemes for Noisy Non-Adaptive Group Testing when Noise Depends on Number of Items in Test, arXiv preprint arXiv:1803.06105 (2018).
[3] Bui, T.V., Kuribayashi, M., Cheraghchi, M. and Echizen, I.: A framework for generalized group testing with inhibitors and its potential application in neuroscience, arXiv preprint arXiv:1810.01086 (2018).
[4] Bui, T.V., Kuribayashil, M., Cheraghchi, M. and Echizen, I.: Efficiently decodable non-adaptive threshold group testing, *2018 IEEE International Symposium on Information Theory* (*ISIT*), pp.2584–2588, IEEE (2018).
[5] Cai, S., Jahangoshahi, M., Bakshi, M. and Jaggi, S.: GROTESQUE: Noisy group testing (quick and efficient), *2013 51st Annual Allerton Conference on Communication, Control, and Computing* (*Allerton*), pp.1234–1241, IEEE (2013).
[6] Cheraghchi, M.: Noise-resilient group testing: Limitations and constructions, *Discrete Applied Mathematics*, Vol.161, No.1-2, pp.81–95 (2013).
[7] Chowdhury, M.F., Jeannerod, C.-P., Neiger, V., Schost, E. and Villard, G.: Faster algorithms for multivariate interpolation with multiplicities and simultaneous polynomial approximations, *IEEE Trans. Information Theory*, Vol.61, No.5, pp.2370–2387 (2015).
[8] Cormode, G. and Muthukrishnan, S.: What's hot and what's not: Tracking most frequent items dynamically, *ACM Trans. Database Systems* (*TODS*), Vol.30, No.1, pp.249–278 (2005).
[9] Damaschke, P.: Threshold group testing, *General Theory of Information Transfer and Combinatorics*, pp.707–718, Springer (2006).
[10] Dorfman, R.: The detection of defective members of large populations, *The Annals of Mathematical Statistics*, Vol.14, No.4, pp.436–440 (1943).
[11] Du, D., Hwang, F.K. and Hwang, F.: *Combinatorial group testing and its applications*, Vol.12, World Scientific (2000).
[12] D'yachkov, A.G. and Rykov, V.V.: Bounds on the length of disjunctive codes, *Problemy Peredachi Informatsii*, Vol.18, No.3, pp.7–13 (1982).
[13] Guruswami, V. et al.: Algorithmic results in list decoding, *Foundations and Trends® in Theoretical Computer Science*, Vol.2, No.2, pp.107–195 (2007).
[14] Guruswami, V. and Indyk, P.: Linear-time list decoding in error-free settings, *International Colloquium on Automata, Languages, and Programming*, pp.695–707, Springer (2004).
[15] Hoorfar, A. and Hassani, M.: Inequalities on the Lambert W function and hyperpower function, *J. Inequal. Pure and Appl. Math*, Vol.9, No.2, pp.5–9 (2008).
[16] Indyk, P., Ngo, H.Q. and Rudra, A.: Efficiently decodable non-adaptive group testing, *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.1126–1142, SIAM (2010).
[17] Kautz, W. and Singleton, R.: Nonrandom binary superimposed codes, *IEEE Trans. Information Theory*, Vol.10, No.4, pp.363–377 (1964).
[18] Lee, K., Pedarsani, R. and Ramchandran, K.: Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes, *2016 IEEE International Symposium on Information Theory* (*ISIT*), pp.2873–2877, IEEE (2016).
[19] Ngo, H.Q. and Du, D.-Z.: A survey on combinatorial group testing algorithms with applications to DNA library screening, *Discrete Mathematical Problems with Medical Applications*, Vol.55, pp.171–182 (2000).
[20] Parvaresh, F. and Vardy, A.: Correcting errors beyond the Guruswami-Sudan radius in polynomial time, *46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005*, pp.285–294, IEEE (2005).
[21] Porat, E. and Rothschild, A.: Explicit non-adaptive combinatorial group testing schemes, *International Colloquium on Automata, Languages, and Programming*, pp.748–759, Springer (2008).
[22] Von Zur Gathen, J. and Nöcker, M.: Exponentiation in finite fields: Theory and practice, *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pp.88–113, Springer (1997).

Thach V. Bui is a Ph.D. candidate at SOKENDAI (The Graduate University for Advanced Studies), Hayama, Kanagawa, Japan. He received a B.S. degree of the honor program in Information Technology from the University of Science, Vietnam in 2012. His research interests include group testing, information theory, and cryptography. He is a student member of IEEE and ACM.

Minoru Kuribayashi received B.E., M.E., and D.E. degrees from Kobe University, Kobe, Japan, in 1999, 2001, and 2004. From 2002 to 2007, he was a Research Associate in the Department of Electrical and Electronic Engineering, Kobe University. In 2007, he was appointed as an Assistant Professor at the Division of Electrical and Electronic Engineering, Kobe University. Since 2015, he has been an Associate Professor in the Graduate School of Natural Science and Technology, Okayama University. His research interests include digital watermarking, information security, cryptography, and coding theory. He received the Young Professionals Award from IEEE Kansai Section in 2014. He is a senior member of IEEE.

Tetsuya Kojima received his B.E., M.E., and D.E. degrees in information engineering from Hokkaido University, Sapporo, Japan, in 1992, 1994 and 1997, respectively. From 1997 to 2001, he was with the Graduate School of Information Systems, the University of Electro-Communications, Tokyo, Japan as a research associate. In 2001, he joined the Department of Computer Science, National Institute of Technology, Tokyo College, Tokyo, Japan, as research associate, and is currently a professor. His research interests include the wireless communication systems, information hiding and applications of information theory. He has served as an organizing committee member in various conferences and workshops, such as General Co-Chair of the Eighth International Workshop on Signal Design and Its Applications in Communications (IWSDA'17). He is a member of IEEE.

Roghayyeh Haghvirdinezhad received B.Sc. and M.S. degrees from the Amirkabir University of Technology, Iran, and the New Jersey Institute of Technology, USA, respectively. Her research interests include information theory and computational complexity.

Isao Echizen received B.S., M.S., and D.E. degrees from the Tokyo Institute of Technology, Japan, in 1995, 1997, and 2003, respectively. He joined Hitachi, Ltd. in 1997, and until 2007 was a research engineer in the company's systems development laboratory. He is currently a deputy director general of the National Institute of Informatics (NII), and a professor and a director of the Information and Society Research Division, the NII, and a professor in the Department of Informatics, the School of Multidisciplinary Sciences, The Graduate University For Advanced Studies (SOKENDAI). He is also a visiting professor at the Tsuda University and was a visiting professor at the University of Freiburg in 2010 and at the University of Halle-Wittenberg in 2011. He has been engaged in research on information security and content security and privacy. He received the Best Paper Award from the IPSJ in 2005 and 2014, the Fujio Frontier Award and the Image Electronics Technology Award in 2010, the One of the Best Papers Award from the Information Security and Privacy Conference and the IPSJ Nagao Special Researcher Award in 2011, the Docomo Mobile Science Award in 2014, the Information Security Cultural Award in 2016, and the Best Paper Award at the IEEE WIFS 2017. He is a member of the Information Forensics and Security Technical Committee and IEEE Signal Processing Society.