EasyChair Preprint

№ 15082

Non-Adaptive Multi-Stage Algorithm for Group Testing with Prior Statistics

Ayelet Cohen-Portnoy and Alejandro Cohen

September 26, 2024

Non-Adaptive Multi-Stage Algorithm for Group Testing with Prior Statistics

Ayelet C. Portnoy and Alejandro Cohen
Faculty of ECE, Technion, Haifa, Israel, Emails: ayeletco@campus.technion.ac.il and alecohen@technion.ac.il

***Abstract*—In this paper, we propose an efficient multi-stage algorithm for non-adaptive Group Testing (GT) with general correlated prior statistics. The proposed solution can be applied to any correlated statistical prior represented in trellis, e.g., finite state machines and Markov processes. We introduce a variation of List Viterbi Algorithm (LVA) to enable accurate recovery using much fewer tests than objectives, which efficiently gains from the correlated prior statistics structure. Our numerical results demonstrate that the proposed Multi-Stage GT (MSGT) algorithm can obtain the optimal Maximum A Posteriori (MAP) performance with feasible complexity in practical regimes, such as with COVID-19 and sparse signal recovery applications, and reduce in the scenarios tested the number of pooled tests by at least 25% compared to existing classical low complexity GT algorithms. Moreover, we analytically characterize the complexity of the proposed MSGT algorithm that guarantees its efficiency.**

I. Introduction

Classical Group Testing (GT) aims to detect a small number of "defective" items within a large population by mixing samples into as few pooled tests as possible. The idea of GT was first introduced during World War II when it was necessary to discover soldiers infected with Syphilis. Dorfman [1] showed that the required number of tests could be reduced if multiple blood samples were grouped into pools. When the samples that participate in the next pool are selected iteratively based on the previous pool test results, the GT algorithm is called adaptive. In contrast, in non-adaptive GT, the whole process is designed in advance. Since it was first suggested, the GT problem has been investigated and generalized to many areas and applications, among them disease detection [2]–[4], cyber security applications [5], pattern matching algorithms [6] and communication [7], [8].

All of these applications imply a strong connection between GT and Compressed Sensing (CS) as two methods for sparse signal recovery that share common applications [9]–[12]. The main difference between the two is that CS aims to recover a real-valued signal [13], while GT recovers a binary signal [14] or discrete-values [4], [15], [16]. Thus, one can consider GT as a Boolean CS [17], [18].

Traditional GT and its performance (i.e., the tradeoff between the number of pool tests and recovery algorithm complexity), focused on the probabilistic model in which the items are identically distributed and independent [14], [19]. Recent research explores cases where prior information about the correlation of objects is available [2], [3], [20]–[22]. The motivation for this approach arises from the fact that correlated prior statistics have the potential to achieve higher recovery rates and reduce the number of required tests. In disease detection, leveraging information about the proximity between individuals, represented by contact tracing information or graphs, can lead to significant savings in pool tests [2], [3], [20], [21]. However, previous GT works presented solutions designed for specific models and applications and may not be extended easily to other models and applications. In numerous signal processing applications, correlation between different frequencies, time signals, or among different sensors can also be utilized to achieve more precise estimations [13], [23], [24]. Hidden Markov Model (HMM) is a common model for many physical signals, such as speech signals [25], human motion [26], and spectrum occupancy in communication systems [27]. Infections can be also modeled as HMMs [28]. For example, [29] presents a GT solution for a specific HMM derived from contact tracing. To the best of our knowledge, no existing solution addresses the GT problem with general Markovian priors and applicable to a wide range of diverse applications.

In this work, we introduce a practical non-adaptive Multi-Stage GT (MSGT) algorithm for correlated items with prior statistics. The proposed multi-stage algorithm employs a new variation of the parallel List Viterbi Algorithm (LVA) [30]–[32] we designed for GT to enable accurate low complexity recovery using fewer tests. The proposed algorithm can be applied for any statistical prior represented in trellis [33], e.g., finite-state machines and Markov processes. Using LVA, MSGT leverages those statistics to estimate the defective set efficiently, even in a regime below the maximum likelihood (ML) upper bound. Furthermore, we show how the algorithm's parameters can be tuned to achieve a maximum probability of success without exceeding the limitation of the available computational capacity. We derive a lower bound that considers the exact priors of the problem and provides analytical results that characterize the MSGT computational complexity efficiency. Our numerical results demonstrate that in practical regimes for COVID-19 [34], [35] and sparse signal recovery in signal processing [9]–[11], [13], the low computational complexity MSGT algorithm proposed herein can reduce in the scenarios tested the number of pool tests by at least 25%.

The rest of this paper is organized as follows. Section II formally describes the GT model with correlated prior statistics. Section III presents the MSGT algorithm and the analytical results. Section IV details the simulation evaluation. Finally, Section V provides concluding remarks and future directions.

II. Problem Formulation

Given a set of individuals $\mathcal{N}$, the objective in GT is to discover a small subset $\mathcal{K}$ of unknown defective items using the minimum number of measurements T. Let $N = |\mathcal{N}|$, $K = |\mathcal{K}|$ denote the total number of items and the number of defective items, respectively, where $K = \mathcal{O}(N^{\theta})$ for some $\theta \in [0, 1)$. The binary vector $\mathbf{U} \in \{0, 1\}^N$ represents the

population, such that $U_i = 1$ indicates that the i-th item is defective. We assume that the set of the individuals is sparse, such that $\theta \leq 1/3$ [14], [17], [19], [36], and that each item can be in one of 2 states: defective and not-defective. For each item, there is some prior probability that it is defective, $\{\pi_i\}_{i=1}^N \in [0,1]^N$, $\pi_i = P(U_i = 1)$. The correlation between the state of the i-th item and the states of the previous τ items is represented by $\mathbf{\Phi}_i \in [0,1]^{2^\tau \times 2}$. $\tau = 1, 2, \ldots$ represents the number of memory steps of the underlying process considered in the recovery stage. When there is no memory in the process, $\tau = 0$ and $\{\pi_i\}_{i=1}^N$ hold all the prior information. $\mathbf{\Phi}_i[l,k] = P(U_i = s_k | (U_{i-\tau}, ..., U_{i-1}) = s_l)$, where s_l is the binary representation of l by a τ-length binary vector and $s_k \in \{0,1\}$ (For example see Fig. 6b).

For the non-adaptive GT, the testing matrix $\mathbf{X} \in \{0,1\}^{T \times N}$ is defined such that each row corresponds to a single pool test, and each column corresponds to a single item. That is, the i-th pool test is represented as a binary row vector: $X_i = [X_i(1), ..., X_i(N)]$, $i \in \{1, ..., T\}$ whose elements are defined: $X_i(j) = 1$ if the item with an index $j \in \{1, ..., N\}$ is included in the i-th pool test, and otherwise $X_i(j) = 0$. Then, the outcome of the i-th pool test is given by

$$Y_i = \bigvee_{j \in \mathcal{K}} X_i(j) = \bigvee_{j=1}^{N} X_i(j) U_j$$

where $\bigvee$ is the Boolean OR function.

Given $\mathbf{X}$ and the outcome vector Y, the recovery success criterion in GT can be measured using various metrics [14]. The main metrics we will use herein are exact recovery and partial recovery. In terms of exact recovery, the goal is to detect the precise subset of defective items $\mathcal{K}$. Therefore, given the estimated defective set $\hat{\mathcal{K}} = \hat{\mathcal{K}}(N, K, \mathbf{X}, Y)$, we define the average error probability by[1]

$$P_e^{exact} \triangleq \frac{1}{\binom{N}{K}} \sum_{\mathcal{K}:|\mathcal{K}|=K} P\left(\hat{\mathcal{K}} \neq \mathcal{K}\right).$$

In partial recovery, we allow a false positive (i.e., $|\hat{\mathcal{K}} \setminus \mathcal{K}|$) and false negative (i.e., $|\mathcal{K} \setminus \hat{\mathcal{K}}|$) detection rate. Thus, the average partial success rate is given by:

$$P_s^{partial} \triangleq \frac{1}{\binom{N}{K}} \sum_{\mathcal{K}:|\mathcal{K}|\neq K} \frac{\left|\hat{\mathcal{K}} \cap \mathcal{K}\right|}{K}.$$

To conclude, for a sparse subset of infected items of size K, form N, the goal in non-adaptive GT with correlated prior data items, is to design a $T \times N$ testing matrix and an efficient and practical recovery algorithm that can exploit correlated priors, such that by observing Y^T we can identify the subset of infected items with high probability and with feasible computational complexity. Thus, given the knowledge of the correlated prior data items and the available computational resources, the test designer could design the testing matrix and a recovery algorithm to maximize the success probability.

[1]For simplicity of notation, P_s and $P_e = 1 - P_s$ refer to success and error probabilities in the exact recovery analysis.

III. Main Results

In this section, we introduce the efficient multi-stage recovery algorithm for any statistical prior represented in a trellis diagram [33], detailed in Algorithm 1. In the first stage, standard low-complexity algorithms [19] reduce the search space independently of prior correlations. This reduction is guaranteed by new analytical results we derive. In the second stage, the algorithm employs a novel adaptation of the List Viterbi Algorithm (LVA) [32], designed for GT to enable accurate low-complexity recovery using fewer tests by exploiting the correlated prior information. Additionally, we derive a bound to ensure the low complexity of the entire algorithm. Section III-A describes the proposed algorithm. Section III-B provides analytical results, followed by a discussion in Section III-C. We refer the reader to [37, Appendix A] for a detailed explanation of all the algorithms used as integral components of Algorithm 1.

A. Pool-Testing Algorithm

1) Testing Matrix and Pooling: The proposed multi-stage recovery algorithm is intended to work with any non-adaptive testing matrix, e.g., as given in [38]. To simplify the technical aspects and focus on the key methods, the testing matrix is generated randomly under a fixed optimal approximation with Bernoulli distribution of $p = \ln(2)/K$ [39], using classical GT methods. The pooling and its outcome are given by the process elaborated in Section II and illustrated in Fig. 1.

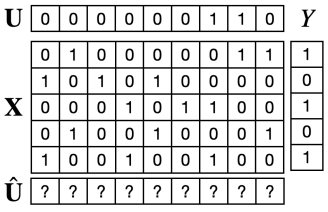


Fig. 1: For an unknown population $\mathbf{U} \in \{0,1\}^9$ with $K = 2$, a random testing matrix is sampled and the test result Y is calculated.

2) Recovery Process: The suggested recovery algorithm operates in two main stages. In the first stage (Stage 1), to reduce the space of search (i.e., the possible defective items), the algorithm efficiently identifies non-defective and definitely defective items without considering the prior correlated information. In the first step of this stage, we use the Definitely Not Defective (DND) algorithm [19], [40]–[42] (line 1, Fig. 2.(a)). DND compares the columns of the testing matrix, $\mathbf{X}$, with the outcome vector, Y^T. If $Y(i) = 0$ for some $i \in \{1, \ldots, T\}$, the algorithm eliminates all items participating in the i-th test from being defective, and outputs them as the set $\mathcal{P}^{(DND)} \subset \mathcal{N}$.

In the second step of Stage 1, we use Definite Defectives (DD) algorithm [19] (line 2, Fig. 2.(b)), which goes over the testing matrix and the test result, looking for positive pool tests that include only one possibly defective item. DD denotes those items as definitely defective items and outputs them as the set $\mathcal{P}^{(DD)}$.

Let $\mathcal{P}^{(S_{1,1})} = \mathcal{N} \setminus \mathcal{P}^{(DND)}$ and $\mathcal{P}^{(S_{1,2})} = \mathcal{N} \setminus \left(\mathcal{P}^{(DND)} \cup \mathcal{P}^{(DD)}\right)$ denote the set of items that their status

Algorithm 1 Multi-Stage Recovery Algorithm

Input: $\mathbf{X}, Y, \{\pi_i, \mathbf{\Phi}_i\}_{i=1}^N, K, L, \tau, \gamma$
Output: $\hat{\mathcal{K}}$

Stage 1: Reduction of space search

1: $\mathcal{P}^{(DND)} \leftarrow \mathbf{DND}(\mathbf{X}, Y)$
2: $\mathcal{P}^{(DD)} \leftarrow \mathbf{DD}\left(\mathbf{X}, Y, \mathcal{P}^{(DND)}\right)$
3: $\mathcal{P}^{(S_{1,2})} \leftarrow \mathcal{N} \setminus \left(\mathcal{P}^{(DND)} \cup \mathcal{P}^{(DD)}\right)$

Stage 2: Recovery exploiting prior info

4: $\{\pi_i, \mathbf{\Phi}_i\}_{i=1}^N \leftarrow \mathbf{updatePriors}\left(\{\pi_i, \mathbf{\Phi}_i\}_{i=1}^N, \mathcal{P}^{(S_{1,2})}, \mathcal{P}^{(DD)}\right)$
5: $\mathbf{Z} \leftarrow \mathbf{LVA}\left(L, \tau, \{\pi_i, \mathbf{\Phi}_i\}_{i=1}^N\right)$
6: $\mathbf{C} \leftarrow \{\}$
7: **for** $l \leftarrow 1$ to L **do**
8: **if** $K \leq \sum_i (\mathbf{Z}_{l,i}) \leq \gamma K$ **then**
9: $\mathbf{V}^{(l)} \leftarrow \{i \mid \mathbf{Z}_{l,i} = 1\}$
10: $\mathbf{C} \leftarrow \mathbf{C} \cup \mathbf{getAllCombinations}\left(\mathbf{V}^{(l)}, K\right)$
11: **end if**
12: **end for**
13: $\hat{\mathcal{K}} \leftarrow \mathbf{MAP}\left(\mathbf{X}, Y, \mathbf{C}, \{\pi_i, \mathbf{\Phi}_i\}_{i=1}^N\right)$

is still unclear after the first step and the second step, respectively. $\mathcal{P}^{(S_{1,2})}$ holds a new space search, and $\mathcal{P}^{(DD)}$ holds the already known defectives. This knowledge is acquired without utilizing any prior data, which we reserve for the second stage.

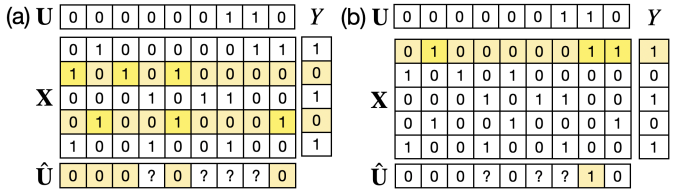


Fig. 2: First stage of MSGT. (a) The first step of Stage 1, the DND algorithm, reveals 5 DND items in $\mathbf{U}$, forming $\mathcal{P}^{(DND)}$. Since items participating in negative tests must be non-defective, we mark all the participants in the two negative test results as non-defective. (b) The second step of Stage 1, the DD algorithm, outputs $\mathcal{P}^{(DD)}$ that includes a single DD item, based on the first test result, as it is the only possibly defective item participating in this test. The two other positive test results do not contribute to our knowledge here because there is more than one possibly defective item participating in them.

In the first step of Stage 2, we translate the data we obtained in DND and DD, $\mathcal{P}^{(S_{1,2})}$ and $\mathcal{P}^{(DD)}$, into the state space in terms of transition probabilities, $\{\mathbf{\Phi}_i\}_{i=1}^N$, and initial probabilities, $\{\pi_i\}_{i=1}^N$, so we can employ all the gathered information in the next steps (line 4, Fig. 3.(a)). In the state space, the population sequence, $\mathbf{U}$, is parallel to time steps considered traditionally in VA for communications problems [32], and there are two possible states per item, the first indicates "non-defective" and the second indicates "defective".

In the second step, the suggested LVA for GT goes over the sequence of items and outputs $\mathbf{Z} \in \{0,1\}^{L \times N}$, a list of the L most likely trajectories in the state space (line 5, Fig. 3.(b)) according to MAP decision based on the given prior information. Each trajectory is a sequence of N states, representing items classified as either defective or non-defective. Thus, $\mathbf{Z}$ provides L estimations of $\mathbf{U}$. In practice, the L estimations may include any number of defective items and require further processing.

In the third step, we extract candidates for the defective set out of the L estimated sequences $\mathbf{Z}$ (lines 7 to 12). For some $l \in \{1, \ldots, L\}$, let $\mathbf{V}^{(l)}$ denote the set of items estimated as defective in $\mathbf{Z}_l$, the l-th row of $\mathbf{Z}$ (line 9). We ignore sequences that contain less than K defective items or more than γK defective items, for some $\gamma \geq 1$, and consider only $\mathbf{Z}_l$ in which $K \leq \left|\mathbf{V}^{(l)}\right| < \gamma K$ as valid sequences. For each one of the valid sequences, we refer to all the combinations of size K in $\mathbf{V}^{(l)}$ as candidates for $\hat{\mathcal{K}}$, and add them to the candidates list $\mathbf{C}$ (line 10).

At this point, we have in $\mathbf{C}$ a list of candidates to be our final estimation $\hat{\mathcal{K}}$, and we can calculate the probability of each one of them using $\{\pi_i, \mathbf{\Phi}_i\}_{i=1}^N$. Then, in the fourth step, the estimated defective set, $\hat{\mathcal{K}}$, is finally chosen using MAP estimator out of the $\mathbf{C}$ (line 13), i.e., $\hat{\mathcal{K}} = \arg\max_{c \in \mathbf{C}} P\left(Y | \mathbf{X}, c\right) P\left(c\right)$.

If there are no valid sequences in $\mathbf{Z}$, we consider trajectories with fewer than K detections for partial recovery. We select the trajectory with the most detections and randomly complete it to form a set of size K for our final estimation $\hat{\mathcal{K}}$.

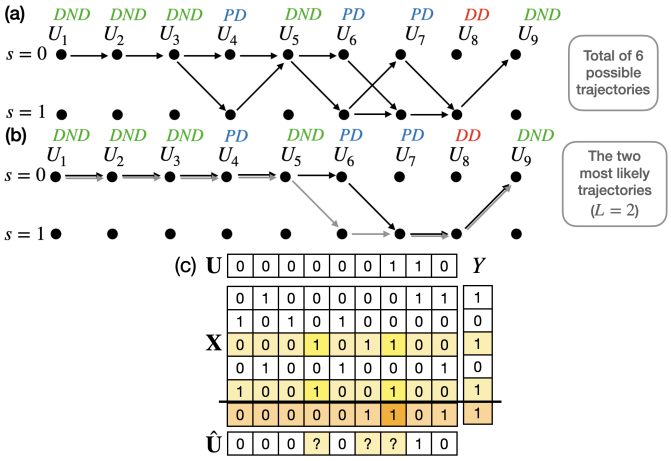
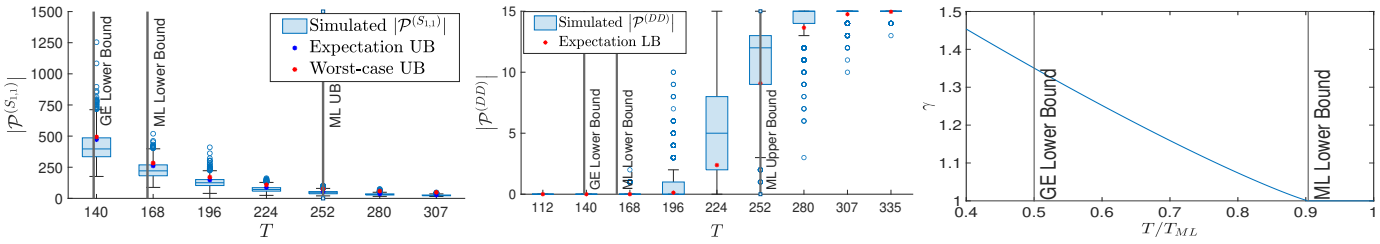


Fig. 3: Stage 2 of MSGT. (a) All the possible transitions in the state space that we consider in the LVA step, following the insights obtained in Stage 1. These transitions aggregate to a total of 6 trajectories. (b) The two most likely trajectories returned by LVA (assuming $L = 2$). Given $K = 2$, the black trajectory corresponds to a valid population vector $\mathbf{U}$ with 2 defective items, while the gray trajectory indicates an invalid population with 3 defective items instead. Consequently, in the subsequent step, MSGT will extract two optional defective sets: $\{U_6, U_8\}$ and $\{U_7, U_8\}$, and will finally choose the most likely one using MAP estimator. (c) Comparison of Stage 2 to ML. With $T = 5$, we use the first 5 rows of the testing matrix, ignoring the last test result. This leaves 3 possibly defective items, forming two potentially defective sets of size $K = 2$. Using ML, one set is chosen randomly, leading to an error probability of 0.5. With $T = 6$, based on the third and sixth test results, there is only one set of size $K = 2$ that matches the outcome Y, resulting in successful decoding with the ML decoder. As shown above, MSGT's Stage 2 can successfully decode $\mathbf{U}$ with just $T = 5$, as using the LVA step it narrows down to only 2 possible trajectories, and then the final estimation is selected based on the given prior information and the insights gained in Stage 1.

B. Analytical Results

Here we provide analytical results related to the proposed MSGT. The proof of certain theorems is technical and falls



(a) Upper bound for possibly defective items after DND for $N = 10000$, and $K = 15$.

(b) Lower bound for definitely defective items after DD for $N = 10000$, and $K = 15$.

(c) Minimum γ parameter to satisfies (2) for $N = 500$, and $K = 3$.

Fig. 4: Numerical evaluation for theoretical results and bounds. The results in (a), (b), and (c) are over 1000 iterations. For ML Upper Bound (UB), $T_{ML} = (1+\epsilon)K\log_2 N$, for any $\epsilon > 0$ [17]. In particular, $\epsilon = 0.25$ in the results presented herein.

outside the scope of this paper. The complete proofs are deferred to [37].

Let $P_{e,a}^{(DND)}$ and $P_{e,u}^{(DND)}$ denote the error probability of DND in the average case and its deviation from the average ("worst-case") that still allows the success of MSGT on average in feasible computational complexity, respectively. The two following Theorems give the upper bound on the expected number of possible defective items and a "worst-case" upper bound after the first step of Stage 1.

Theorem 1 ([4])**.** *Consider a group test with a Bernoulli testing matrix with $p = \ln 2/K$, and T tests as $K \to \infty$. Let $P_{e,a}^{(DND)} \triangleq N^{-\alpha(1-\ln 2/K)/2}$ for $\alpha \triangleq T/K\log_2 N$. The expected number of possibly defective items is bounded by*

$$\mathbb{E}\left[\left|\mathcal{P}^{(S_{1,1})}\right|\right] \leq K + (N-K)\,P_{e,a}^{(DND)}.$$

Theorem 2. *Consider a group test with a Bernoulli testing matrix and DND decoder. For any $\gamma \geq 1$, as $K \to \infty$, the worst-case error probability of DND is bounded by*

$$P_{e,u}^{(DND)} \leq P_{e,a}^{(DND)} + \left(1 + \frac{\left(\gamma K - \left(K + (N-K)\,P_{e,a}^{(DND)}\right)\right)^2}{(N-K)\,P_{e,a}^{(DND)}\left(1 - P_{e,a}^{(DND)}\right)}\right)^{-1}.$$

Proof Sketch. Let G denote the number of non-defective items that were hidden and not detected in DND (i.e., the number of false positive items). The probability of each definitely defective item being hidden, $P_e^{(DND)}$, depends solely on the design of the testing matrix. We upper bound $P_e^{(DND)}$, by the sum of the average probability of error $P_{e,a}^{(DND)}$ and a concentration term:

$$P_{e,u}^{(DND)} \triangleq P_{e,a}^{(DND)} + P\left(G - \mathbb{E}[G] > g\right),$$

for some $g > 0$. We are interested in $g = \gamma K - \left(K + (N-K)\,P_{e,a}^{(DND)}\right)$ since LVA's performance in Stage 2 may be worse than the brute force performance of MAP, if the number of occlusions in the testing matrix exceeds the threshold γK. In the case of Bernoulli encoder, the $N-K$ random variables, representing occlusions of non-defective items, are independent and $G \sim Bin\left(N-K, P_e^{(DND)}\right)$. Using the binomial distribution and applying the one-sided Chebyshev inequality [43] to the concentration term completes the proof. □

The following theorem provides the expected number of defective items detected by DD in the second step of Stage 1.

Theorem 3. *Consider a group test with a Bernoulli testing matrix with $p = \ln 2/K$, and T tests as $K \to \infty$. The expected number of defective items that DD successfully detects is bounded by*

$$\mathbb{E}\left[\left|\mathcal{P}^{(DD)}\right|\right] \geq K\left(1 - N^{-\frac{\alpha}{2}(1-\ln 2/K)^{N\cdot P_{e,a}^{(DND)}}}\right). \quad (1)$$

Proof Sketch. Since $\mathbb{E}\left[\left|\mathcal{P}^{(DD)}\right|\right] = \left(1 - \mathbb{E}\left[P_e^{(DD)}\right]\right)K$, we focus on bounding $\mathbb{E}\left[P_e^{(DD)}\right]$. The detection of the i-th defective item ($i \in \mathcal{K}$) in the DD algorithm may fail under two conditions: either it does not participate in any test or it participates, but at least one other potentially defective item participates in the same pool test. Let A_i represent the event in which at least one potentially defective item, excluding item i, participates in some test. Consequently, the probability of A_i is given by $P(A_i) = 1-(1-p)^{\left|\mathcal{P}^{(S_{1,1})}\right|-1}$. The probability of not identifying item i as definitively defective in a given test is given by $1 - p + pP(A_i)$. Therefore, the probability of not detecting a defective item in all T tests is given by

$$P_e^{(DD)} = \left[1 - p\,(1-p)^{\left|\mathcal{P}^{(S_{1,1})}\right|-1}\right]^T.$$

Next, we substitute the result shown in Theorem 1 and use the fact that $e^{-x} \geq (1-x/n)^n$ for any integer $n > 0$ and $x > 0$. It follows that $\mathbb{E}\left[P_e^{(DD)}\right] \leq N^{-\frac{\alpha}{2}\left(1-\frac{\ln 2}{K}\right)^{P_{e,a}^{(DND)}N}}$, which concludes the proof. □

The proposition below provides a sufficient condition on γ, such that, MSGT achieves performance that we conjecture outperforms ML. This can be assumed since LVA used for this step is an optimal MAP estimator [32].

Proposition 1. *Let η denote the average success probability of ML decoder. For GT with correlated prior information and $\gamma \geq 1$, MSGT can achieve an average success probability equal or greater than η, if*

$$T \geq \log\eta + \log\binom{N}{K} - \log\binom{N-K}{K(\gamma-1)}. \quad (2)$$

Proof. We follow the lines of the success probability analysis in [36, Theorem 1]. Consider a list decoder that, given the test outcome Y, outputs $\mathcal{L}(Y)$, a list of items estimated as

defective, such that $|\mathcal{L}(Y)| \geq K$. Let $N(Y)$ denote the number of possible defective sets for which, given Y, the list decoding is successful: $N(Y) \triangleq \sum_{\mathcal{K}} \mathbb{1}_{\{\mathcal{L}(Y)\cap\mathcal{K}=K\}}$. Assuming no prior information is given, $N(Y)$ is bounded by the counting bound

$$N(Y) \leq \binom{N-K}{L-K}. \tag{3}$$

The success probability of the decoder can be written as

$$P_s = \frac{1}{\binom{N}{K}} \sum_{Y} N(Y) = \frac{2^T}{\binom{N}{K}} N(Y). \tag{4}$$

By substituting (3) into (4), taking the logarithm and rearranging, we have that

$$T \geq \log P_s + \log \binom{N}{K} - \log \binom{N-K}{L-K}.$$

It is important to note that unlike [36, Theorem 1], we halt the success probability analysis here without deriving a converse, and instead focus on the connection between T and γ. This analysis is valid for any list decoder, particularly for an ML decoder [17], [44], allowing us to set $P_s = \eta$. Since MSGT uses MAP, the non-uniform and correlated prior information allows us to achieve better recovery performance for the same T and L compared to ML. Moreover, since for any $\gamma \geq 1$, γK is the maximum list size of estimated defective items we allow in MSGT. By substituting $L = \gamma K$, we establish a lower limit for γ that guarantees MSGT's success probability remains at least equal to that of the ML estimator. □

One of the key features of the proposed MSGT algorithm is its low and feasible complexity in practical regimes compared to ML or MAP-based GT decoders. Both ML and MAP involve exhaustive searches, resulting in a complexity of $\mathcal{O}(\binom{N}{K} KN \log_2 N)$ operations [17]. The theorem below and the subsequent Remark characterize the computational complexity of MSGT.

Theorem 4. *Consider a group test for a population of N items, of which K are defective and a Bernoulli testing matrix. The computational complexity of the MSGT algorithm is bounded by $\mathcal{O}\left(L\gamma^K KN \log_2 N\right)$ operations.*

Proof. We begin by analyzing the complexity of each step of the proposed MSGT solution given in Algorithm 1, and finally, we sum everything up to determine the total complexity.

In Stage 1, the complexity of DND is $\mathcal{O}\left(KN \log_2 N\right)$ as analyzed in [45, Remark 6]. Then, for each positive entry of the test result vector Y^T, the DD algorithm counts the number of possibly defective items that participate in the corresponding pool test. That requires $K\left|\mathcal{P}^{(S_{1,1})}\right| \log_2 N$ computations, and for simplification, we bound it by the DND complexity, i.e., $\mathcal{O}\left(KN \log_2 N\right)$.

In Stage 2, parallel LVA requires L times more computations than the VA [32]. VA calculates all the possible transition probabilities for each step in the sequence. In GT, this sequence is the order's items sequence, where with the suggested algorithm, it is enough to consider only the $\left|\mathcal{P}^{(S_{1,2})}\right|$ items as the sequence steps. The optional states are basically either "non-defective" or "defective", so there are four possible transitions in each step of the trellis proposed herein for GT. Nevertheless, this algorithm can be implemented to leverage additional memory to decide the state of each item based on the preceding τ items. Consequently, LVA takes $2^{2\tau} L \left|\mathcal{P}^{(S_{1,2})}\right|$ computations. For the average case, we use the expectation bounds from Theorems 1 and 3, such that:

$$\begin{aligned}\mathbb{E}\left[\left|\mathcal{P}^{(S_1)}\right|\right] &= \mathbb{E}\left[\left|\mathcal{P}^{(S_{1,1})}\right|\right] - \mathbb{E}\left[\left|\mathcal{P}^{(DD)}\right|\right] \\ &\leq (N-K)\, P_{e,a}^{(DND)} + N^{-\frac{\alpha}{2}(1-\ln 2/K)^{N\cdot P_{e,a}^{(DND)}}} K.\end{aligned}$$

Accordingly, the number of the required computations for LVA step in MSGT is bounded by

$$2^{2\tau} L \left[(N-K)\, P_{e,a}^{(DND)} + N^{-\frac{\alpha}{2}(1-\ln 2/K)^{N\cdot P_{e,a}^{(DND)}}} K\right]. \tag{5}$$

For the worst-case, we assume that the DD step does not affect the possible detected items set, thus $\left|\mathcal{P}^{(S_{1,2})}\right| \leq \left|\mathcal{P}^{(S_{1,1})}\right|$. To further bound the expression, we utilize both the error probability of the worst case as given in Theorem 2, thus $\left|\mathcal{P}^{(DND)}\right| \leq K + (N-K)P_{e,u}^{(DND)}$. By substituting Theorem 2 it follows that the number of possibly defective items in the worst-case is upper bounded by

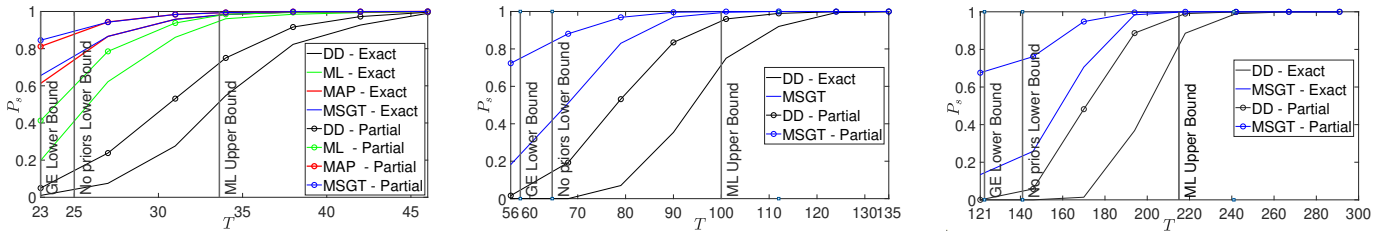
$$K + (N-K)\left(P_{e,a}^{(DND)} + \left(1 + \frac{g^2}{(N-K)\, P_{e,a}^{(DND)}\left(1 - P_{e,a}^{(DND)}\right)}\right)^{-1}\right), \tag{6}$$

for $g = \gamma K - \left(K + (N-K)\, P_{e,a}^{(DND)}\right)$. Now, since the two expressions multiplying $(N-K)$ in (5) and (6) are error probabilities, both of them can be roughly bounded by one. Hence, the number of computations of LVA in the average case and worst-case becomes $\mathcal{O}(LN)$.

In the next step in Stage 2, we filter the LVA results. We sum each sequence $\mathbf{Z_l}$ with a complexity of $\mathcal{O}(N)$. If the result is in the range $[K, \gamma K]$, we extract all the combinations of size $K - \left|\mathcal{P}^{(DD)}\right|$. Thus, this stage is done in at most $\mathcal{O}\left(L\left(N + \binom{\gamma K}{K - \left|\mathcal{P}^{(DD)}\right|}\right)\right)$ computations.

Finally, in the MAP step of Stage 2, the algorithm goes over at most L combinations of size $K - \left|\mathcal{P}^{(DD)}\right|$ out of no more than γK possibly defective items in each sequence. Then, for each combination, the group test is applied. Therefore, the MAP stage requires $\binom{L\gamma K}{K} KN \log_2 N$ computations. Substituting the bounds of the binomial coefficient $\left(\frac{\gamma K}{K}\right)^K \leq \binom{\gamma K}{K} \leq \left(\frac{e\gamma K}{K}\right)^K$, it follows that the complexity of MAP stage in the proposed MSGT is $\mathcal{O}\left(L\gamma^K KN \log_2 N\right)$.

To conclude, the complexity of MSGT is the sum of all the steps, i.e., $\mathcal{O}\big(NK \log_2 N + LN + L\binom{\gamma K}{K - \left|\mathcal{P}^{(DD)}\right|} + L\gamma^K KN \log_2 N\big)$. As N grows, the dominant term is the complexity of the MAP step. Thus, the complexity of the MSGT algorithm is bounded by $\mathcal{O}\left(\gamma^K KN \log_2 N\right)$ operations, which completes the proof. □



(a) $N = 500$ and $K = 3$. (b) $N = 1024$ and $K = 8$. (c) $N = 10000$ and $K = 13$.

Fig. 5: Success probability of MSGT, MAP, ML and DD over 1000 iterations. A comparison to ML and MAP is not presented in (b) and (c), as they are not feasible for populations of those sizes due to the computational complexity burden.

Remark 1. *If we skip the LVA step, MSGT converges to the MAP estimator. Thus, the MAP's complexity is*

$$\mathcal{O}\left(\binom{\left|\mathcal{P}^{(S_{1,2})}\right|}{K-\left|\mathcal{P}^{(DD)}\right|}KN\log_2 N\right),$$

when the DND and DD are executed as prior steps, and otherwise it is $\mathcal{O}\left(\binom{N}{K}KN\log_2 N\right)$.

Note that from Theorem 4 and Remark 1, it follows that the proposed MSGT algorithm performs $\mathcal{O}\left(\frac{1}{L}\left(\frac{N}{\gamma K}\right)^K\right)$, times fewer computational operations compared to MAP.

C. Discussion

To the best of our knowledge, MSGT is the first GT algorithm to effectively leverage Markovian prior statistics. Unlike numerous previous approaches, MSGT utilizes initial probabilities and transition matrices without necessitating specific adjustments for new use cases. The algorithm offers the flexibility to be fine-tuned to optimize its performance in accordance with the available computational resources and the number of tests, T. The simple reduction of the search space in Stage 1 enables MSGT to handle challenging regimes with a small number of tests. Stage 2, particularly the LVA step, contributes to its high success probability. Additionally, using the parallel implementation of LVA, rather than the iterative one, keeps the complexity low [32]. It is important to note that, as explained in [32], achieving optimal results is ensured only with a very large L, inevitably leading to complexity equivalent to MAP's. However, as we empirically demonstrate in the following section, results equivalent to MAP's can be achieved with reasonable complexity. Moreover, it is shown that MSGT addresses practical regimes, e.g., in COVID-19 detection [46] ($T = 48$ for $(N, K) = (384, 4)$), in communications [47] ($(N, K) = (10^5, 6)$), and in GT quantizer [8] ($(N, K) = (1024, 16)$ [48]).

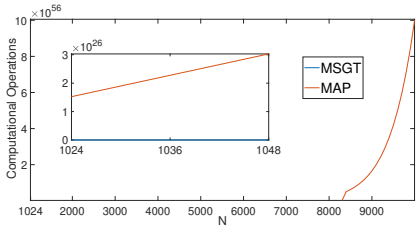
Another aspect of novelty in this work is the integration of Viterbi Algorithm into the GT problem. In the context of Markovian priors, one can think of the population's sequence of items as a sequence of observations stemming from a hidden Markov process within a given Markov model over N steps. In that case, the selection of a Viterbi decoder becomes natural, offering an optimal and efficient decoding solution. However, the most likely sequence of items does not necessarily include K defective items. Particularly, in sparse signal scenarios, which is the focus of GT, the most likely sequence typically involves the minimum number of defectives that explain the observations. As a result, VA may not necessarily detect more defective items than already known and may detect even more than K. To address this, we employ LVA, which produces a list of the L most likely sequences, such that choosing an appropriate value for L guarantees a successful recovery.

Like many previous works, MSGT assumes precise knowledge of K. In practical use cases, this assumption relies on using an accurate estimator for K employing $\mathcal{O}(\log_2(N))$ tests [49], but the estimation might be erroneous. For the sake of practical completeness, it is worth noting that modifications can be made to handle incorrect estimates of K, albeit with increased computational complexity. The authors of [4] suggest altering the ML estimator to consider all possible sets, without restricting the number of defective items, and show that the probability of success is almost unaffected. MSGT relies on knowing K only for the MAP estimation step. Thus, a similar adaptation can be applied. In MSGT, the MAP estimator should consider all possible sets from LVA and cannot stop with the first set that explains the outcome Y. Hence, such modification will increase the complexity compared to [4].

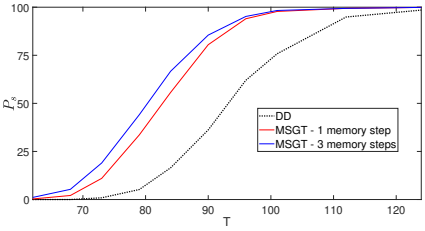
IV. Numerical Evaluation

This section assesses the performance of the proposed MSGT algorithm by numerical study. First, in Subsection IV-1, we provide a numerical evaluation to support our theoretical results and bounds given in Section III. Then, in Subsection IV-2, we contrast the performance of MSGT with the one of DD, ML, and MAP in a practical regime of N and K. To generate the correlated prior information between adjacent items, we use the Gilbert-Elliot (GE) model [50], which is a stationary Markov chain of order 1 with two states: one representing an error phase and the other an error-free phase. The GE model is characterized by initial probabilities assigned to these two states, denoted as $\pi_i \in [0, 1]^2$, as well as transition probabilities between them $\mathbf{\Phi}_i \in [0, 1]^{2\times 2}$. These characteristics align well with the inputs required by the MSGT algorithm. In the practical scenarios tested (e.g., in the regime of COVID-19, when the test machine can simultaneously process a fixed small number of measurements [34], [35], or in sparse signal recovery in signal-processing with fixed vector size of input samples [9]–[11], [13], [51]), we show that the low computational complexity MSGT algorithm can reduce the number of pool tests by at least 25%.

1) Theoretical Analysis: In Fig. 4a we show the concentration of $\left|\mathcal{P}^{(S_{1,1})}\right|$, as obtained from the simulation, along



(U_{i-2}, U_{i-1}) \ U_i	0	1
(0,0)	$P(U_i=0\|U_{i-2}=0,U_{i-1}=0)$	$P(U_i=1\|U_{i-2}=0,U_{i-1}=0)$
(0,1)	$P(U_i=0\|U_{i-2}=0,U_{i-1}=1)$	$P(U_i=1\|U_{i-2}=0,U_{i-1}=1)$
(1,0)	$P(U_i=0\|U_{i-2}=1,U_{i-1}=0)$	$P(U_i=1\|U_{i-2}=1,U_{i-1}=0)$
(1,1)	$P(U_i=0\|U_{i-2}=1,U_{i-1}=1)$	$P(U_i=1\|U_{i-2}=1,U_{i-1}=1)$



(a) Number of computational operations. (b) Correlation matrix $\mathbf{\Phi}_i$ for $\tau = 2$. (c) Long memory priors - exact recovery.

Fig. 6: (a) Number of computational operations in MSGT and MAP as given in Theorem 4 and Remark 1, respectively. (b) Example of correlation matrix $\mathbf{\Phi}_i$. (c) Probability of success of MSGT with exact prior statistics of 3-memory-steps Markov process, and with limited prior statistics assuming the Markov process has only one memory step. $N = 1024, K = 8$, 1000 iterations.

with the bound on its expectation and on the worst-case that were calculated in Theorems 1 and 2 respectively. Note that the worst-case scenario regarding MSGT is when LVA filters the correct set of defective items. That may happen if the number of possibly defective items exceeds the threshold γK. Since in MSGT, we only allow this deviation from the average and ignore the case of exceeding this threshold, our upper bound for the worst-case does not cover all potential realizations of $\left|\mathcal{P}^{(S_{1,1})}\right|$. Similarly, Fig. 4b demonstrates the concentration of $\left|\mathcal{P}^{(DD)}\right|$, as acquired through simulation, and the lower bound on its expectation as given in Theorem 3. Fig. 4c illustrates the numerically computed lower bound for γ, derived from the inequality provided in Proposition 1. For this simulation, we fix $N = 500$, $K = 3$, and calculate γ value for a specific range of T relative to the upper bound of ML. As explained in Section III, our conjecture asserts that any value of γ surpassing this lower bound guarantees that MSGT performance will be at least on par with that of ML. Therefore, whenever computational resources allow, it is advisable to choose the value of γ corresponding to the lower bound. This approach was followed in the subsequent simulations, and the practical outcomes presented in Subsection IV-2 provide empirical support for our conjecture. Fig.6a compares the number of potential combinations to be examined in the MAP step, with and without the execution of the LVA step in MSGT. It can be observed that the LVA step performs an extensive filtering process, which allows MSGT to remain feasible even when executing MAP is no longer possible, especially in a regime below ML upper bound.

The converse of the GT problem with general prior statistics was developed by Li et al. [52] and according to which any GT algorithm with a maximum error probability P_e requires a number of tests that satisfies: $T \leq (1 - P_e)\, H(\mathbf{U})$, where $H(\cdot)$ denotes entropy. Using the joint entropy identity we have: $(1 - P_e)\, H(\mathbf{U}) = \sum_{i=1}^{N} H\left(U_i | U_1, \ldots, U_{i-1}\right)$.

The GE model considered in our numerical evaluations is a stationary Markov chain with $\tau = 1$. Thus, $P\left(U_i | U_0, \ldots, U_{i-1}\right) = P\left(U_i | U_{i-1}\right)$, for $i \in \{2, \ldots, N\}$. Substituting those priors, it follows that the converse of our problem is: $T \leq H(U_1) + \sum_{i=2}^{N} H\left(U_i | U_{i-1}\right)$. This bound is illustrated in the practical scenarios tested in Fig. 4 and Fig. 5.

2) Algorithm Evaluation: We demonstrate the performance of MSGT using simulation. The population is sampled from GE model, and the regime is $K = \mathcal{O}\left(N^{\theta}\right)$ with $\theta \leq 1/3$. The GE parameters serve as our prior statistics, but in practice, we ignore samples where the number of defective items does not match K. In addition, although in the complexity analysis, we considered Bernoulli encoder for simplification, here we use a near-constant column weight encoder that optimizes DND's performance [38], with $\frac{\ln 2}{K} T$ tests sampled randomly for each item. The parameter γ was chosen to satisfy (2) and $L = 500$ was chosen empirically. Fig. 5 compares MSGT to MAP, ML, and DD algorithms. We run DND and DD before ML and MAP for reasonable runtime and memory consumption. The population includes $N \in \{500, 1024, 10000\}$ items and $K \in \{3, 8, 13\}$ defective items, respectively, and the empirical success probability is the average over 1000 experiments. Note that for $N = 1024, 10000$ it is no longer possible to compare the performance since ML and MAP become infeasible (Fig. 5b,5c).

Finally, MSGT was also tested with more complex probabilistic models. We sample the population based on a 3-memory-steps Markov process ($\mathbf{\Phi}_i$ is a 8×2 matrix for all i). We execute MSGT using these prior statistics and also execute it with limited prior statistics, assuming that the process has only one memory step ($\mathbf{\Phi}_i$ is 2×2). For example of $\mathbf{\Phi}_i$ representing a 2-memory-steps process, see Fig 6b. The results are shown in Fig. 6c. It is evident that utilizing the prior of long memory improves the success probability by 10% in this scenario. We do note again that for the practical regime tested as in [8], [46]–[48], i.e., $N = 1024$ and $K = 8$, it is no longer possible to compare the performance since, unlike the efficient proposed MSGT algorithm, ML and MAP decoders for GT become infeasible.

V. Discussion and Future Work

To the best of our knowledge, the existing solutions in literature do not offer an efficient and general solution for the regime below T_{ML}. In this work, we are focusing on this regime of the number of tests (i.e., $T < T_{ML}$), where the proposed MSGT can leverage the prior correlated information, within the LVA step, to significantly reduce the number of the potential defective combinations. This approach yields an efficient computational solution that, as demonstrated in our simulation result for practical scenarios, can approach the minimum number of tests as MAP algorithm.

Future work includes the derivation of an upper bound for this problem, which holds significant importance in comprehending the algorithm's potential.

REFERENCES

[1] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[2] R. Goenka, S.-J. Cao, C.-W. Wong, A. Rajwade, and D. Baron, "Contact tracing enhances the efficiency of COVID-19 group testing," in *ICASSP 2021-2021 IEEE ICASSP*. IEEE, 2021, pp. 8168–8172.

[3] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi, "Dynamic group testing to control and monitor disease progression in a population," in *2022 IEEE Int. Sym. on Inf. Theory (ISIT)*. IEEE, 2022, pp. 2255–2260.

[4] A. Cohen, N. Shlezinger, A. Solomon, Y. C. Eldar, and M. Médard, "Multi-level group testing with application to one-shot pooled COVID-19 tests," in *ICASSP 2021-2021 IEEE ICASSP*. IEEE, 2021, pp. 1030–1034. Full Version. [Online]. Available: https://drive.google.com/drive/folders/1k77uIgOFXDwmNPdtd72v2IrEnOBgjgud?usp=drive_link

[5] M. T. Thai, Y. Xuan, I. Shin, and T. Znati, "On detection of malicious users using group testing techniques," in *2008 The 28th Int. Conf. on Distributed Computing Systems*. IEEE, 2008, pp. 206–213.

[6] P. Indyk, "Deterministic superimposed coding with applications to pattern matching," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, 1997, pp. 127–136.

[7] J. Wolf, "Born again group testing: Multiaccess communications," *IEEE Trans. on Inf. Theory*, vol. 31, no. 2, pp. 185–191, 1985.

[8] A. Cohen, N. Shlezinger, Y. C. Eldar, and M. Médard, "Serial quantization for representing sparse signals," in *2019 57th Allerton*. IEEE, 2019, pp. 987–994.

[9] A. C. Gilbert, M. A. Iwen, and M. J. Strauss, "Group testing and sparse signal recovery," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*. IEEE, 2008, pp. 1059–1063.

[10] V. Y. Tan and G. K. Atia, "Strong impossibility results for sparse signal processing," *IEEE Sig. Proc, Letters*, vol. 21, no. 3, pp. 260–264, 2014.

[11] C. Aksoylar, G. K. Atia, and V. Saligrama, "Sparse signal processing with linear and nonlinear observations: A unified shannon-theoretic approach," *IEEE Trans. on Inf. Theory*, vol. 63, no. 2, 2016.

[12] A. Naderi and Y. Plan, "Sparsity-free compressed sensing with applications to generative priors," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 3, pp. 493–501, 2022.

[13] Y. C. Eldar and G. Kutyniok, "Theory and applications, compressed sensing," 2012.

[14] M. Aldridge, O. Johnson, J. Scarlett *et al.*, "Group testing: an information theory perspective," *Foundations and Trends® in Communications and Information Theory*, vol. 15, no. 3-4, pp. 196–392, 2019.

[15] A. Cohen, A. Cohen, and O. Gurewitz, "Efficient data collection over multiple access wireless sensors network," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 491–504, 2020.

[16] A. Cohen, N. Shlezinger, S. Salamatian, Y. C. Eldar, and M. Médard, "Serial quantization for sparse time sequences," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3299–3314, 2021.

[17] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Trans. on Inf. Theory*, vol. 58, no. 3, 2012.

[18] D. Sejdinovic and O. Johnson, "Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction," in *2010 48th Allerton*. IEEE, 2010, pp. 998–1003.

[19] M. Aldridge, L. Baldassini, and O. Johnson, "Group testing algorithms: Bounds and simulations," *IEEE Trans. on Inf. Theory*, vol. 60, no. 6, pp. 3671–3687, 2014.

[20] P. Nikolopoulos, S. R. Srinivasavaradhan, C. Fragouli, and S. N. Diggavi, "Group testing for community infections," *IEEE BITS the Information Theory Magazine*, vol. 1, no. 1, pp. 57–68, 2021.

[21] I. Lau, J. Scarlett, and Y. Sun, "Model-based and graph-based priors for group testing," *arXiv preprint arXiv:2205.11838*, 2022.

[22] B. Arasli and S. Ulukus, "Group testing with a graph infection spread model," *Information*, vol. 14, no. 1, p. 48, 2023.

[23] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011.

[24] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "Compressive sensing: Performance comparison of sparse recovery algorithms," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, 2017, pp. 1–7.

[25] M. Gales, S. Young *et al.*, "The application of hidden markov models in speech recognition," *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[26] Y. Su, L. Lloyd, X. Chen, and J. G. Chase, "Latency mitigation using applied HMMs for mixed reality-enhanced intuitive teleoperation in intelligent robotic welding," *The International Journal of Advanced Manufacturing Technology*, vol. 126, no. 5, pp. 2233–2248, 2023.

[27] C. Ghosh, C. Cordeiro, D. P. Agrawal, and M. B. Rao, "Markov chain existence and hidden markov models in spectrum sensing," in *2009 IEEE Int. Conf. on Pervasive Comp. and Comm.* IEEE, 2009, pp. 1–6.

[28] K. M. Ceres, Y. H. Schukken, and Y. T. Gröhn, "Characterizing infectious disease progression through discrete states using hidden markov models," *Plos one*, vol. 15, no. 11, p. e0242683, 2020.

[29] S.-J. Cao, R. Goenka, C.-W. Wong, A. Rajwade, and D. Baron, "Group testing with side information via generalized approximate message passing," *IEEE Trans. on Sig. Proc.*, vol. 71, pp. 2366–2375, 2023.

[30] J. Andrew, "Viterbi. 1967. error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. on Inf. Theory*.

[31] H.-L. Lou, "Implementing the viterbi algorithm," *IEEE Signal processing magazine*, vol. 12, no. 5, pp. 42–52, 1995.

[32] N. Seshadri and C.-E. Sundberg, "List viterbi decoding algorithms with applications," *IEEE Trans. on Comm.*, vol. 42, pp. 313–323, 1994.

[33] T. K. Moon, *Error correction coding: mathematical methods and algorithms*. John Wiley & Sons, 2020.

[34] C. Lucia, P.-B. Federico, and G. C. Alejandra, "An ultrasensitive, rapid, and portable coronavirus SARS-CoV-2 sequence detection method based on CRISPR-Cas12," *BioRxiv*, pp. 2020–02, 2020.

[35] Ben-Ami *et al.*, "Large-scale implementation of pooled RNA extraction and RT-PCR for SARS-CoV-2 detection," *Clinical Microbiology and Infection*, vol. 26, no. 9, pp. 1248–1253, 2020.

[36] J. Scarlett and V. Cevher, "How little does non-exact recovery help in group testing?" in *2017 IEEE ICASSP*. IEEE, 2017, pp. 6090–6094.

[37] A. C. Portnoy and A. Cohen, "Multi-stage algorithm for group testing with prior statistics," *arXiv preprint arXiv:2402.10018*, 2024.

[38] O. Johnson, M. Aldridge, and J. Scarlett, "Performance of group testing algorithms with near-constant tests per item," *IEEE Trans. on Inf. Theory*, vol. 65, no. 2, pp. 707–723, 2018.

[39] M. Aldridge, L. Baldassini, and K. Gunderson, "Almost separable matrices," *Jour. of Comb. Opt.*, vol. 33, pp. 215–236, 2017.

[40] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. on Inf. Theory*, vol. 10, no. 4, pp. 363–377, 1964.

[41] H.-B. Chen and F. K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *Jour. of Comb. Opt.*, vol. 15, pp. 49–59, 2008.

[42] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama, "Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms," in *2011 49th Allerton*. IEEE, 2011, pp. 1832–1839.

[43] J. Kingman, "An introduction to probability theory and its applications," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 430–430, 1972.

[44] R. G. Gallager, *Information theory and reliable communication*. Springer, 1968, vol. 588.

[45] A. Cohen, A. Cohen, and O. Gurewitz, "Secure group testing," *IEEE Trans. on Inf. Forensics and Security*, vol. 16, pp. 4003–4018, 2020.

[46] N. Shental, S. Levy, S. Skorniakov, V. Wuvshet, Y. Shemer-Avni, A. Porgador, and T. Hertz, "Efficient high throughput SARS-CoV-2 testing to detect asymptomatic carriers," *medRxiv*, 2020.

[47] J. Luo and D. Guo, "Neighbor discovery in wireless Ad Hoc networks based on group testing," in *2008 46th Allerton*. IEEE, 2008, pp. 791–797.

[48] L. Jacques, K. Degraux, and C. De Vleeschouwer, "Quantized iterative hard thresholding: Bridging 1-bit and high-resolution quantized compressed sensing," *arXiv preprint arXiv:1305.1786*, 2013.

[49] P. Damaschke and A. S. Muhammad, "Competitive group testing and learning hidden vertex covers with minimum adaptivity," *Discrete Mathematics, Algorithms and Applications*, vol. 2, no. 03, pp. 291–311, 2010.

[50] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell system technical journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[51] Z. Li, W. Xu, X. Zhang, and J. Lin, "A survey on one-bit compressed sensing: Theory and applications," *Frontiers of Computer Science*, vol. 12, pp. 217–230, 2018.

[52] T. Li, C. L. Chan, W. Huang, T. Kaced, and S. Jaggi, "Group testing with prior statistics," in *2014 IEEE Int. Sym. on Inf. Theory*. IEEE, 2014, pp. 2346–2350.