Recovery Algorithms for Pooled RT-qPCR based Covid-19 Screening

Sameera Bharadwaja H. and Chandra R. Murthy

***Abstract*—We consider the problem of sparse signal recovery in a non-adaptive pool-test setting using quantitative measurements from a non-linear model. The quantitative measurements are obtained using the reverse transcription (quantitative) polymerase chain reaction (RT-qPCR) test, which is the standard test used to detect Covid-19. Each quantitative measurement refers to the *cycle threshold*, a proxy for the viral load in the test sample. We propose two novel, robust recovery algorithms based on alternating direction method of multipliers and block coordinate descent to recover the individual sample cycle thresholds and hence determine the sick individuals, given the pooled sample cycle thresholds and the pooling matrix. We numerically evaluate the normalized mean squared error, false positive rate, false negative rate, and the maximum sparsity levels up to which error-free recovery is possible. We also demonstrate the advantage of using quantitative measurements (as opposed to binary outcomes) in non-adaptive pool testing methods in terms of the testing rate using publicly available data on Covid-19 testing. The simulation results show the effectiveness of the proposed algorithms.**

***Index Terms*—RT-qPCR, Covid-19, SARS-CoV-2, Group Testing, Compressed Sensing, Sparse Signal Recovery.**

I. Introduction

IN recent times, COrona (SARS-CoV-2) VIrus Disease (Covid-19), which originated in late 2019 as a local infection, has become a pandemic of unprecedented levels. The disease has spread rapidly and globally from the place of its origin. The mortality rate of Covid-19 is statistically low ($\sim 0.5-4\%$) [1]. However, due to the large caseload in small geographical regions, it has destabilized the healthcare systems in many countries. In a significant number of cases (guesses vary widely $\sim 50-85\%$), the disease manifests itself with mild or no symptoms [2]–[6]. However, such individuals (silent spreaders) are still able to spread the disease [7], [8]. During large public gatherings, the presence of such individuals can and has in the past resulted in massive super-spreader events. Thus, it is a major concern for opening up campuses, office spaces, and other public amenities. Rapid testing, contact tracing, and isolation of the infected has been the only known effective way to control the spread of the disease [9].

In the past couple of years, various vaccines have been out in the market. However, the virus has been mutating rapidly, with each strain having a different combination of lethality and transmissibility (R_0) [10]. As a result, the world has witnessed at least three waves of the pandemic to date, with varying severity levels. Therefore, systematic, cost-effective, reliable, and repeatable testing protocols which can aid effective contact tracing have not lost their importance [11]. Among the various methods that have been developed to detect the Covid-19 virus in an individual [12], one of the most reliable testing methods is the reverse transcription (quantitative) polymerase chain reaction (RT-qPCR), which is described next.

A. RT-qPCR Process

The RT-qPCR process [12], [13] is a type of nucleic acid amplification test. A biological (e.g., naso-oropharyngeal swab) sample is collected from an individual and viral RNA molecules, if any, are extracted via a pre-test preparation process. The RNA molecules are converted to complementary DNA (cDNA) molecules using the RT process. The PCR process, which is a sequence of exponential amplification cycles with heating and cooling phases, is conducted next. Identical copies of the target DNA are obtained in each cycle, roughly doubling the initial population. The rate at which viral loads replicate is often called the PCR (amplification) *efficiency factor,* denoted by q. A *Taqman* probe is added, which contains fluorophores that emit light upon excitation. The intensity of light emitted is proportional to the number of viral DNA strands present in the sample at any given time. The cycle at which the fluorescent light intensity exceeds a preset threshold, denoted by τ, is called the *cycle threshold* (CT) value and is the quantitative output of the RT-qPCR test.

The CT usually takes values between $15-35$ for positive Covid-19 tests [14]. It can also be noted that a higher initial viral load implies that the preset threshold is crossed in the earlier cycles in the PCR process, i.e., CT is low. Also, zero viral loads in the sample are depicted by CT $= \infty$, to say that the preset threshold is not crossed in a finite time.

B. Motivation for the Proposed Work

The major non-renewable components of the RT-qPCR test are the reagents like the Taqman probes, primers, and time-to-test itself. The overall RT-qPCR test procedure typically takes 3 to 8 hours [15]. A standard PCR plate can accommodate roughly 93 or 381 individual samples depending upon the plate layout, after reserving space for positive and negative controls (PC and NCs)[1]. Scaling horizontally by procuring more testing kits is often not feasible due to the high procurement and the operational costs involved. Therefore, vertically scaling, i.e.,

Sameera Bharadwaja H. and Chandra R. Murthy are with the Department of ECE, Indian Institute of Science, Bangalore, India (email: sameerah@iisc.ac.in; cmurthy@iisc.ac.in).

This work was funded by the Young Faculty Research Fellowship, Ministry of Electronics and Information Technology, Govt. of India.

[1]PCR Plastics: https://www.thermofisher.com/in/en/home/life-science/cloning/cloning-learning-center/invitrogen-school-of-molecular-biology/pcr-education/pcr-qpcr-plastics/pcr-qpcr-plastics-considerations.html

making the testing process more efficient in effective time-taken per sample and resource usage is crucial in aiding large-scale testing reliably and repetitively (e.g., daily).

Vertical scaling of the test process can be accomplished by pool testing [16] (a.k.a. group testing). In recent decades, group testing has converged to the area of compressed sensing (CS), and is often called boolean compressed sensing [17]. We briefly describe the idea here for the sake of completeness.

The idea of pool testing, first proposed by Dorfman [18] in 1943, is as follows. A subset of samples are mixed together to form a *pooled sample* and tested at once. If the pooled sample tests negative, all the participating samples are deemed *negative* for the disease. A positive pooled test implies that one or more participating individual samples are positive. In this case, the items participating in the pooled test are tested individually. Pool testing has a long history, and has been applied in various settings, including in nucleic acid amplification tests like PCR (see [19] and the references therein) and also for Covid-19 detection [20].

Dorfman-style testing methods are called *adaptive* pool testing [16]. In contrast, in non-adaptive pool testing, all the required tests are performed in a single stage, followed by an application of a suitable decoding algorithm [21]–[23] to recover the individual sample status given the pooled test outcomes and a *pooling matrix*. Each RT-PCR test takes several hours to run, and at the same time, a standard RT-PCR plate can accommodate either 93 or 381 samples depending on the plate layout. An adaptive test, which is a multi-stage procedure, would require 2x or more time than a single-stage non-adaptive group test. Secondly, pooled sample preparation before each stage of multi-stage adaptive testing would expose the technician to bio-hazards for longer duration. Lastly, using finite size, deterministic non-adaptive pooling matrices is practical, especially with the advent of pooling robots [24]. Therefore, non-adaptive pooling methods offer time-advantage, are practitioner-friendly, and are safer for infectious disease testing than the adaptive testing methods. A binary pooling matrix specifies which individuals participate in which test.

The authors in [17] show that the estimation of the n length individual status vector is feasible with an arbitrarily small probability of error using m pooled tests as long as $m \geq O(k \log n)$, where k denotes the number of sick individuals. However, this is an asymptotic bound, i.e., it is valid as $n \to \infty$, with k growing sub-linearly with n [17].

Recently, testing of pooled samples was shown to be feasible for the detection of Covid-19 using RT-qPCR [25]. The authors in [26] optimize pool size and test protocol for Covid-19 detection via RT-qPCR tests and conclude that the limit of detection is $1-3$ RNA copies per μl. An algorithm for classifying each individual as having $\{$*no, low, medium, high*$\}$ infection level was developed in [27]. However, in practice, one may be interested in recovering the actual viral load of the individuals, rather than coarse classification.

The performance of binary pool testing algorithms like combinatorial orthogonal matching pursuit (COMP) and its noisy version, Noisy-COMP (NCOMP) was studied in [28]. In the current work, we use COMP as a pre-processing step to filter out negative tests with CT $= \infty$ and reduce the problem dimension. The authors in [29] use the non-negative least absolute deviation (NNLAD) algorithm for decoding in a non-adaptive pool testing setup. The authors in [30] and [31] consider improving test reliability and throughput via possibly non-binary pooling matrices. Although these papers use sparse recovery techniques to infer the individual viral loads, the system model in these papers is formulated as linear observations corrupted by additive noise, which does not match with well-accepted RT-qPCR models [32].

In [33], a weighted least-squares (LS) approach is used to solve the non-linear CS problem $\mathbf{y} = f(\mathbf{A}\mathbf{x})$, where $\mathbf{A}$ is the test matrix, $f(\cdot)$ is the non-linearity due to the amplification and interpolation operation relating the CT values, $\mathbf{y}$, to the sample viral loads, $\mathbf{x}$. For lab-experiments, they model the sensing matrix as $\mathbf{A} = \mathbf{P} \odot \mathbf{W}$, where $\mathbf{P}$ is a boolean-valued participation matrix and $\mathbf{W}$ is a positive real-valued sample allocation matrix. The (i,j)th element of the allocation matrix determines the fraction of the jth sample participating in the ith test. We note that the sensing matrix is no longer binary-valued; instead, it has either 0 or positive real entries.

The authors in [24] convert the observed pooled-sample CT values into corresponding viral loads. Such conversion requires the knowledge of the PCR efficiency factor ($q = 0.95$ is used). The value of q depends on various factors like probe-primers used, other PCR specifications (exact operating temperature of the test process), etc. We show that the mismatch between the true value of q and its value assumed by the decoding algorithm can result in severe performance degradation. Therefore, it is essential to develop algorithms that can jointly learn q and estimate the individual CTs from the pooled CT values.

In addition to the binary and CS-based quantitative methods for group testing, there are approaches which do not fall strictly into either category, and therefore requires a separate mention. For instance, the authors in [34] propose to use tropical arithmetic and formulate an adaptive group testing protocols based on a *delay and match* principle. The delay-and-match principle uses a protocol where they add samples into the pool during the testing process, i.e., say after, Δ cycles, and use this information while decoding.

In the current work, we focus on estimating the viral loads and determining healthy/sick status of individuals using the pooled RT-qPCR outcomes and the deterministic, single-stage pooling matrix. Our main contributions are as follows:

1) We propose two novel algorithms, namely, a gradient descent (GD)-CT method and an iterative mirrored hard thresholding (IMHT)-CT method, which can recover the individual CT vector given the pooled CT vector and the pooling matrix. The challenge lies in addressing the non-linear nature of the mathematical model of the RT-qPCR process. Due to this, the recovery problem departs from the standard sparse signal recovery problem [35], [36].
2) In practice, the PCR efficiency factor, q, is unknown. To address the issue of performance degradation due to the mismatch between the efficiency factor assumed by the algorithm and the true efficiency factor, we propose two novel algorithms based on alternating direction method of multipliers (ADMM) and block coordinate descent

(BCD), which can jointly recover the individual sample CT vector and estimate the unknown efficiency factor. The sub-problem of estimating the CTs is performed using GD-CT or IMHT-CT. The sub-problem involving q is solved using the projected gradient descent (PGD) method. The proposed recovery algorithms are robust to noise and varying machine parameters.

3) The advantage of using quantitative measurements in non-adaptive pool testing in terms of the testing rate and hence the cost is presented using publicly available data on the number of tests conducted. The results illustrate that using quantitative measurements results in significant cumulative cost savings.
4) We compile the best rates achievable for a given prevalence rate using deterministic testing matrices like Kirkman[2] [37] and Euler[3] [38] designs. The prevalence rate estimate (e.g., the previous day's value) can be used to decide the optimum (testing) rate design for each day.
5) We empirically evaluate the performance of the proposed algorithms and show that they outperform related algorithms in the literature under practical settings (e.g., unknown machine-specific parameters, CT measurement noise, etc.) in terms of the normalized mean-squared error (NMSE) and the sparsity level up to which the algorithms guarantee zero recovery errors.

One of the main takeaways from our work is that the use of quantitative measurements in a non-adaptive pool test setting results in significant cumulative cost savings. Also, the optimal achievable testing rate vs. prevalence rate compiled using various deterministic pooling matrix designs and recovery algorithms in the non-linear RT-qPCR model is crucial for reducing pooled testing ideas to practice. Finally, the proposed algorithms recover the un-normalized CT values without the knowledge of the machine-specific parameters. This makes the output of our algorithms similar to that of individual testing.

Notation: The real number space is denoted by $\mathbb{R}$. Sets are denoted by calligraphic letters, e.g., $\mathcal{X}$; with $|\mathcal{X}|$ denoting its cardinality. Matrices and vectors are represented using boldface upper and lowercase letters, respectively. The symbol $\setminus$ denotes the set difference operation. The ℓ_1 and ℓ_2 norms of a vector, $\mathbf{x}$, are denoted by $\|\mathbf{x}\|_1$ and $\|\mathbf{x}\|_2$, respectively [39]. $\|\mathbf{x}\|_0$ denotes the ℓ_0 norm of $\mathbf{x}$, i.e., the number of non-zero elements in $\mathbf{x}$ [40, Definition 2.1]. $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose operations, and $\|\mathbf{A}\|_F$ and $\|\|\mathbf{A}\|\|_2$ denote the Frobenius norm and spectral norm of the matrix $\mathbf{A}$ [39]. For a scalar $a \in \mathbb{R}$ and a vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{n\times 1}$, we define $a^{\mathbf{x}} \triangleq [a^{x_1}, a^{x_2}, \ldots, a^{x_n}]^T$. $\mathbf{A} \succcurlyeq 0$ means that the matrix $\mathbf{A}$ is positive semi-definite [39]. Finally, trace of the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the sum of its n diagonal elements.

II. System Model

A system model inspired by the RT-qPCR mechanism [32] is described in this section. Denote the maximum number of cycles in the PCR process by $c_{\max}$. The cycle number is indexed by t and therefore, $t \in \{1, 2, \ldots, c_{\max}\}$. Also, denote the efficiency of the PCR reaction by q, and let x_0 denote the initial viral load concentration (e.g., DNA molecules per μl). For the sample to be considered positive, the viral load concentration should cross a preset threshold τ in $c \leq c_{\max}$ PCR cycles. Using [32, Equation (7)], the relationship governing the growth of the viral load concentration up to the cycle threshold can be written as:

$$\prod_{t=1}^{\lfloor c \rfloor} (1+q_t)(1+q_{\lfloor c \rfloor+1})^{c-\lfloor c \rfloor} x_0 = \tau.$$

Note that, $c \in \mathbb{R}$ although the cycles are indexed by $\{1, 2, \ldots, c_{\max}\}$. This is because the PCR machine's software performs the interpolation implied by the above equation to output a real-valued CT.

By calibrating the operating protocol, in practice, one can ensure that the variation in q_ts across tests is negligible. Then, the model can be simplified by letting $q_t = q$ for all $t = 1, 2, \ldots, c_{\max}$. Thus, (1) can be simplified as

$$(1+q)^c x_0 = \tau. \tag{1}$$

Further, without loss of generality, we have taken the proportionality constant to be unity [32, see Equation (4)], since one can appropriately scale the threshold τ. The value of q depends on various factors like probe-primer combination [41], dilution of the test solution, whether annealing equilibrium [32] method is used or not, etc. In most lab experiments, the value of q is observed to lie in $\mathcal{X} \triangleq [0.5, 1.0)$ over various probe-primer combinations [32], [41]. As mentioned in Sec. I-A, when the viral load is zero, the threshold τ is not reached even after $c_{\max}$ cycles. This is indicated by setting $c = \infty$. In practice, depending on the PCR kit and sample preparation protocols, manufacturer's instructions etc., $c_{\max} = 40$ to 50 PCR cycles are conducted.

We now extend the model to pool testing based on RT-qPCR and account for noise. Here, n individuals participate in m pool tests, with $m \ll n$. The CT values observed from the m pool tests are collected in a vector as $\mathbf{c} \triangleq [c_1, c_2, \ldots, c_m]^T$, where c_i is the CT of the ith pool test. Similarly, the initial viral loads contributed by the n individuals to the pool tests can be written as $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$.[4] In non-adaptive pool testing, the pool tests are defined by the binary pooling matrix $\mathbf{A} \in \{0,1\}^{m\times n}$, where the (i,j)th element of $\mathbf{A}$ equals 1 if the jth individual participates in the ith test, and equals 0 otherwise. We can now extend (1) to the pool testing framework and write the model for the ith pool's CT measurement, denoted by c_i, as

$$\tau(1+q)^{-c_i} = (1+q)^{\epsilon_i} \mathbf{A}_i^T \mathbf{x}\ , \quad i = 1, 2, \ldots, m, \tag{2}$$

where $\mathbf{A}_i^T$ is the ith row of $\mathbf{A}$ and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is the CT measurement noise with unknown variance σ_ϵ^2. In (2), the

[2]See Appendix A for a note on Kirkman designs.

[3]See Appendix B for a note on Euler designs.

[4]Note that, if the jth individual does not participate in the ith test, we may set $x_j = 0$ for that test. Also note that, in practice, the jth individual may contribute different initial viral loads to the different tests it participates in. However, in the detection regimes of interest, where reliable detection is possible (e.g., where the positive individuals contribute about 100 or more viral particles per μl to the test), these variations do not significantly affect the resulting CT values.

term $\mathbf{A}_i^T\mathbf{x}$ represents the effect of pooling. That is, the total initial viral load in the ith pool test is the sum of all the initial viral loads of the individual samples participating in the ith test, determined by the locations of 1s in the row $\mathbf{A}_i^T$. Also, note that the noise contribution appears as an exponent to the overall process efficiency factor, $(1+q)$. This is due to (1), where the CT depends on the initial viral load via an exponential term. Therefore, any additive noise observed during the measurement of the CT values shows a similar exponential dependency with the initial viral load. A similar model was used in [24, Equation (9)], where the authors expressed system model in terms of the ratio of the initial viral loads in the pooled samples and the viral load corresponding to the minimum observed CT among all pools. In contrast, the (intermediate) model in (2) relates the actual un-normalized initial viral loads of the individual samples with the observed pooled CT values. Further, as we shall see, our final model relates the observed pooled CT values to the individual sample CT values, which we want to estimate.

The goal is to solve the inverse problem of inferring the vector of individual viral loads, $\mathbf{x}$, from (2) given the pooling matrix $\mathbf{A}$ and m pooled CTs, $c_1, c_2, \ldots, c_m$. We note that, the system model as shown in (2) is under-determined since $m \ll n$. When the prevalence rate is low, $\|\mathbf{x}\|_0 \ll n$, sparse signal recovery methods can potentially be used for solving the inverse problem at hand. However, our measurement model in (2) is nonlinear and the noise is multiplicative, unlike the standard sparse signal recovery problem [35], [36]. Further, the values of τ and q are unknown. The range of values over which the viral loads typically vary is large, making the inverse problem numerically hard to solve. Suppose $x_0 = 1$ in (1) (i.e., 1 viral particle per μl) results in $c = 35$ in the noiseless case. Assuming an ideal PCR efficiency of $q = 1$, we obtain $\log_2 \tau = 35$. Thus, an observed $c = 15$ corresponds to a viral load $x_0 = 2^{20} \approx 10^6$. As a result, the range over which the viral load can potentially vary is $[1, 10^6]$. In contrast, the range over which the CT values vary is much smaller, i.e., $[15, 35]$.

In order to solve the problem in a numerically stable manner, we transform the problem from *viral-load*-space into *CT*-space. To keep the notations distinct between the *pooled-sample CTs* and the *individual-sample CTs*, the individual-sample CTs are henceforth denoted by $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T \in \mathbb{R}^{n\times 1}$. From (1), the individual-sample CTs are related to the viral loads as

$$\mathbf{x} = \tau(1+q)^{-\mathbf{u}}, \tag{3}$$

where $(1+q)^{-\mathbf{u}} \triangleq [(1+q)^{-u_1}, (1+q)^{-u_2}, \ldots, (1+q)^{-u_n}]^T \in \mathbb{R}^{n\times 1}$. Hence, we get

$$c_i = -\frac{1}{\log(1+q)} \log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right) + \epsilon_i\,, \quad i = 1, 2, \ldots, m. \tag{4}$$

In matrix-vector form,

$$\begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = -\frac{1}{\log(1+q)} \begin{bmatrix} \log\left(\mathbf{A}_1^T(1+q)^{-\mathbf{u}}\right) \\ \vdots \\ \log\left(\mathbf{A}_m^T(1+q)^{-\mathbf{u}}\right) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{bmatrix},$$

$$\text{or } \mathbf{c} = -\frac{1}{\log(1+q)} \log\left(\mathbf{A}(1+q)^{-\mathbf{u}}\right) + \boldsymbol{\epsilon}. \tag{5}$$

The goal now is to solve the inverse problem of inferring $\mathbf{u}$, the vector of individual CTs, from (5), given the pooling matrix, $\mathbf{A}$, and the vector of pooled CTs, $\mathbf{c}$. We note that while the parameter τ does not appear in the CT-space formulation, q is still unknown.

In summary, the following points:

- Usage of binary pooling matrix and the fact that there are two kinds of pooled test outcomes: negative test outcomes have CT $= \infty$ while the positive test outcomes are non-negative finite real values, *and*
- Multiplicative and non-Gaussian nature of the noise term, $(1+q)^{\epsilon}$ as seen in (2) or additive nature of the noise but with a non-linear model in the $\log$ space as seen in (5),

makes the problem different from standard models considered in sparse signal recovery problems [35], [36]. To this end, we propose a set of novel, robust recovery algorithms to estimate the vector of individual CTs.

III. Proposed Algorithms

A. Preliminaries: Binary Algorithms

In this subsection, we describe two well-known binary pool testing algorithms: combinatorial orthogonal matching pursuit (COMP) and definite defectives (DD) [21], [42] applied to the CT system model after transforming the model to a binary system. We note that there are a variety of binary pool testing algorithms like combinatorial basis pursuit (CBP), smallest satisfying set (SSS), sequential COMP (SCOMP), etc. in the literature [21], [42]. We restrict our comparisons to COMP and DD, because, in a noiseless setting, (1) COMP algorithm does not make a type-2 error (false negative); (2) DD algorithm does not make a type-1 error (false positive). Further, as mentioned in Sec. I-B, COMP is used in the pre-processing stage of the CT algorithms (see Secs. III-B and III-C).

1) Combinatorial Orthogonal Matching Pursuit (COMP): The measured pool CT's are converted to binary values using

$$b_i = \begin{cases} 1, & c_i \neq \infty \\ 0, & c_i = \infty, \end{cases}$$

for $i = 1, 2, \ldots, m$. Let $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ denote the index set of samples. The samples appearing in only negative pool tests (i.e., $b_i = 0$) are declared as definite negatives, denoted by the set $\mathcal{DN}$. Then, the samples remaining in the set $\mathcal{N}\backslash\mathcal{DN}$ are declared as positive samples.

In the sequel, we first pre-process the CT values using the COMP algorithm to filter out the definite negatives. Then, in the system model, we remove the rows of $\mathbf{A}$ corresponding to the negative test outcomes, and the columns of $\mathbf{A}$ and the entries of $\mathbf{u}$ corresponding to the definite negatives. This, in turn, reduces the overall system dimension for further processing. In fact, at low prevalence rates, the reduced system can even be over-determined. The application of COMP also removes the negative tests with CTs equal to ∞ and is thus numerically better suited for optimization-based recovery algorithms. Now, in order to avoid additional notation, we reuse the notations defined earlier to define the model after the COMP stage. For example, n and m will respectively denote

the number of samples and number of tests remaining, after the COMP stage.

2) Definite Defectives (DD): The DD algorithm comprises two stages. In stage-1, similar to the COMP algorithm, the definite negatives and hence the set of *possible positives*, $\mathcal{PP} \triangleq \mathcal{N} \setminus \mathcal{DN}$ is determined. In stage-2, the items from $\mathcal{PP}$ which are sole participants in positive pool tests (i.e., $b_i = 1$) are declared as positive samples and the rest are declared as negative samples.

B. The CT Algorithms: Known Efficiency Factor

In this subsection, we describe two algorithms which use the pooled CTs as the input but assume that the efficiency factor, q, is known.

1) Gradient Descent (GD) - CT Algorithm: From (5), the inverse problem we wish to solve can be written as follows:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \ \frac{1}{2}\left\|\boldsymbol{\epsilon}\right\|_2^2 + \bar{\kappa}\left\|\tau(1+q)^{-\mathbf{u}}\right\|_0, \tag{6}$$

where $\bar{\kappa}$ is a regularization parameter. The first term in (6) arises from a least-squares formulation, while the second term uses (3) with the fact that the vector of viral loads, $\mathbf{x}$, is sparse. We further note that the ℓ_0 optimization problem in (6) is NP-hard (see Appendix C). Define $\tilde{\mathbf{x}} = (1+q)^{-\mathbf{u}}$ such that, $\|\tilde{\mathbf{x}}\|_\infty \leq \tilde{\kappa}$. Compute the convex biconjugate of $f(\tilde{\mathbf{x}}) = \|\tilde{\mathbf{x}}\|_0$ to get $f^{**}(\tilde{\mathbf{x}}) = \frac{1}{\tilde{\kappa}}\|\tilde{\mathbf{x}}\|_1$ [43]. We then replace the ℓ_0 norm term in (6) with its ℓ_1 relaxation term to get

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \ \frac{1}{2}\left\|\boldsymbol{\epsilon}\right\|_2^2 + \kappa\left\|(1+q)^{-\mathbf{u}}\right\|_1, \tag{7}$$

where $\kappa \triangleq \bar{\kappa}|\tau|/\tilde{\kappa} \overset{\tau \geq 0}{=} \bar{\kappa}\tau/\tilde{\kappa}$ is the effective regularization parameter for the optimization problem. We note that (7) is still a hard problem to solve due to its non-linear and non-convex nature. In the literature, the gradient descent (GD) is one of the attractive and practical algorithms available to solve a wide range of optimization problems [44]. It is especially useful when the gradient of the objective function exists and can be efficiently computed. The optimization problem in (7) can be solved using the GD algorithm and is termed as the *GD-CT* algorithm due to the usage of CT values.

In order to derive the GD-CT algorithm, we start by computing the gradient of the objective function in (7) with respect to $\mathbf{u}$, denoted by $\mathbf{g}(\mathbf{u}; \mathbf{A}, \mathbf{c}, q)$. Denoting the (i,j)th element of $\mathbf{A}$ by A_{ij}, the gradient is given by

$$\mathbf{g}(\mathbf{u}; \mathbf{A}, \mathbf{c}, q) = \sum_{i=1}^{m}\left(\frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + c_i\right) \times \frac{-1}{\mathbf{A}_i^T(1+q)^{-\mathbf{u}}}\begin{bmatrix}\frac{A_{i1}}{(1+q)^{u_1}}\\ \vdots \\ \frac{A_{in}}{(1+q)^{u_n}}\end{bmatrix} - \kappa\left(\ln(1+q)\right)(1+q)^{-\mathbf{u}}. \tag{8}$$

The GD-CT procedure is presented in Algorithm 1, where η is the step size and K is the maximum number of iterations.

In Algorithm 1 and henceforth, we denote the vector $\mathbf{u}$ at kth iteration of the algorithm and the jth entry of $\mathbf{u}$ by $\mathbf{u}_k$ and u_j, respectively. Then, the (j,j)th diagonal element of the Hessian $\mathbf{H}(\mathbf{u})$ of the objective function is given by

$$H_{jj}(\mathbf{u}) = \sum_{i=1}^{m}\ \Bigg[\frac{c_i A_{ij}\ln(1+q)(1+q)^{-u_j}}{\mathbf{A}_i^T(1+q)^{-\mathbf{u}}} - \frac{c_i A_{ij}^2\ln(1+q)(1+q)^{-2u_j}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2} - \frac{A_{ij}\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})}{(1+q)^{u_j}\ \mathbf{A}_i^T(1+q)^{-\mathbf{u}}} + \frac{A_{ij}^2(1+q)^{-2u_j}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2} - \frac{A_{ij}^2\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})(1+q)^{-2u_j}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2}\Bigg] - \kappa\ln^2(1+q)(1+q)^{-u_j}, \tag{10}$$

and the (j,j')th element for $j \neq j'$ is given by

$$H_{jj'}(\mathbf{u}) = \sum_{i=1}^{m}\ \frac{A_{ij}A_{ij'}(1+q)^{-u_j-u_{j'}}}{(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})^2} \times\left[c_i + \frac{1}{\log(1+q)} - \log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})\right]. \tag{11}$$

Algorithm 1 Gradient Descent (GD) - CT

Input: $\mathbf{c}$, $\mathbf{A}$, K, κ, η and q
Output: $\hat{\mathbf{u}}$

1: Initialize $\mathbf{u}_1$.
2: **for** $k \leftarrow 1$ to K **do**
3: Compute $\mathbf{g}(\mathbf{u}_k; \mathbf{A}, \mathbf{c}, q)$ using (8) with $\mathbf{u} = \mathbf{u}_k$.
4: Update

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \eta\mathbf{g}(\mathbf{u}_k; \mathbf{A}, \mathbf{c}, q). \tag{9}$$

5: **end for**
6: Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{K+1}$.

Theorem 1. *[45], [46] Given an open convex set, $\mathcal{S}$, let the objective function, $f : \mathcal{S} \mapsto \mathbb{R}$ be twice differentiable. Let η be the step size of the GD-CT algorithm such that $0 < \eta < 1/L$, where $\nabla_{\mathbf{u}} f$ is L-Lipschitz continuous, for $L < \infty$. The GD-CT update as given by* (9)*, generates a sequence of iterates $\{\mathbf{u}_k\}$. If the $\lim_{k\to\infty}\ \mathbf{u}_k$ exists, then $\mathbb{P}\left(\lim_{k\to\infty}\ \mathbf{u}_k = \mathbf{u}^*\right) = 1$, where $\mathbf{u}^*$ is a local minimizer.*

The objective function in (7) is twice differentiable (see (10) and (11)) and is comprised of elementary operations. Hence, it is analytic and satisfies the *Łojasiewicz gradient inequality.*[5] Therefore, the limit of the iterates exists [45]. Further, showing that the gradient is L-Lipschitz is equivalent to bounding the spectral norm of the Hessian, $\|\mathbf{H}\|_2 = \sqrt{\lambda_{\max}(\mathbf{H}^H\mathbf{H})} \leq L < \infty$, where $\lambda_{\max}(\mathbf{H}^H\mathbf{H})$ is the largest eigenvalue of $\mathbf{H}^H\mathbf{H}$ [46]. Since $\mathbf{H}^H\mathbf{H} \succcurlyeq 0$, we have $\lambda_{\max}(\mathbf{H}^H\mathbf{H}) \leq \text{trace}(\mathbf{H}^H\mathbf{H}) = \|\mathbf{H}\|_F^2$. Therefore, it suffices to show that the entries of the Hessian are upper bounded by finite constants.

Recall that the COMP pre-processing removes the tests with pooled CT values $c_i = \infty$. Thus, at least one positive sample with $u_j < \infty$ appears in each test, and the pooled CT is bounded as $c_i \leq c_{\max}$. This also ensures that $\mathbf{A}_i^T(1+q)^{-\mathbf{u}} \neq 0$ in the denominator of (10) or (11). Let $\mathcal{S} = (m_u, M_u)$ such that, $0 < m_u < u_j < M_u$. The positive lower bound assumption is valid since at least one PCR cycle is conducted.

[5]See Appendix D for the formal definition and a note.

Let $\tilde{\mathbf{x}} = (1+q)^{-\mathbf{u}}$, such that the entries, $\tilde{x}_j$, of $\tilde{\mathbf{x}}$ are bounded between $m_x < \tilde{x}_j < M_x < \infty$ for the positive individuals and $\tilde{x}_j = 0$ for the negative individuals, where m_x and M_x are constants depending only on M_u and m_u, respectively. Lastly, note that $\log(m_x \left\|\mathbf{A}_i^T\right\|_1) < \log(\mathbf{A}_i^T\tilde{\mathbf{x}}) < \log(M_x \left\|\mathbf{A}_i^T\right\|_1)$. Since $1 \leq \left\|\mathbf{A}_i^T\right\|_1 \leq n$, we can relax the bound further as $\log(m_x) < \log(\mathbf{A}_i^T\tilde{\mathbf{x}}) < \log(nM_x)$. Define $\gamma \triangleq \max(|\log(m_x)|, |\log(nM_x)|)$ and note that $|\log(\mathbf{A}_i^T\tilde{\mathbf{x}})| < \gamma$. Then, the entries of the Hessian can be bounded as

$$H_{jj'} \leq \begin{cases} C_H\left[1+\gamma(n+1)+nc_{\max}\ln(1+q)\right], & j = j' \\ C_H\left[\gamma + c_{\max} + \frac{1}{\ln(1+q)}\right], & j \neq j', \end{cases}$$

with $C_H \triangleq m(M_x/m_x)^2$. Since $\|\mathbf{H}\|_F^2$ contains n diagonal and $n(n-1)$ off-diagonal entries of $\mathbf{H}$, the trace($\mathbf{H}^H\mathbf{H}$), and hence the $\left\|\left|\mathbf{H}(\mathbf{u})\right|\right\|_2$ is finitely upper bounded, satisfying the last of the sufficient conditions for almost sure convergence of Algorithm 1 to a local minimizer using Theorem 1.

Remark: Instead of using a constant step-size η, we can use adaptive step-sizes. Algorithm 1 can be modified to allow step-size adaptation as follows: Replace *step 4* by

$$\eta_k = \text{ AdaptStep}(\eta_{k-1})$$
$$\mathbf{u}_{k+1} = \mathbf{u}_k - \eta_k \mathbf{g}(\mathbf{u}_k; \mathbf{A}, \mathbf{c}, q),$$

where AdaptStep$(\cdot)$ implements the recipe to adapt the step size [47], [48]. In this case, the L-Lipschitz requirement on the gradient in Theorem 1 can be relaxed.

2) Iterative Mirrored Hard Thresholding (IMHT) - CT Algorithm: Motivated by the literature in non-linear compressed sensing [49], the system model in (5) can be seen as a non-linear transformation relating the pooled CTs vector, $\mathbf{c}$ with the individual CTs vector, $\mathbf{u}$, given by $\mathbf{c} = \Phi(\mathbf{u}) + \boldsymbol{\epsilon}$, where $\Phi(\mathbf{u}) \triangleq -\frac{\log\left(\mathbf{A}\left(\frac{1}{(1+q)^{\mathbf{u}}}\right)\right)}{\log(1+q)}$ is a multi-variable vector-valued non-linear function. A first-order Taylor series approximation of $\Phi(\mathbf{u})$ about a fixed point $\mathbf{u}^*$ can be written as $\Phi(\mathbf{u}) \approxeq \Phi(\mathbf{u}^*) + \mathbf{J}(\mathbf{u}^*)(\mathbf{u}-\mathbf{u}^*)$ and is valid when $\|\mathbf{u}-\mathbf{u}^*\| \leq \delta$ for a sufficiently small $\delta > 0$. In the Taylor series expansion, $\mathbf{J}(\mathbf{u})$ is the $m \times n$ Jacobian matrix of $\Phi(\mathbf{u})$ computed at $\mathbf{u} = \mathbf{u}^*$. The (i,j)th element of $\mathbf{J}(\mathbf{u})$ is

$$\mathbf{J}_{ij}(\mathbf{u}) = \frac{\partial(\Phi_i(\mathbf{u}))}{\partial u_j} = \frac{A_{ij}}{(1+q)^{u_j}}\left(\frac{1}{\mathbf{A}_i^T(1+q)^{-\mathbf{u}}}\right), \quad (12)$$

where $\Phi_i(\mathbf{u})$ is the ith element of $\Phi(\mathbf{u})$.

We introduce the following additional notations before describing the algorithm. Let s denote the number of finite entries in the vector, $\mathbf{u}$ or equivalently, the sparsity of the viral-load vector, $\mathbf{x}$. Let $\mathbf{\Pi}_{u_{th},s}(\mathbf{u})$ denote the *mirrored* hard-thresholding operation, i.e., the operation that sets the $n-s$ largest values of $\mathbf{u}$ to u_{th}. With η denoting the step size and K denoting the maximum number of iterations, the *IMHT-CT* algorithm is described in Algorithm 2.

Although the Algorithm 2 takes the value of s as an input, we note that an overestimated value of s does not affect the simulation performance.

C. The CT Algorithms: Unknown Efficiency Factor

In practice, as stated in Sec. I-B, the efficiency factor, q, of the RT-PCR process may not be precisely known.

Algorithm 2 Iterative Mirrored Hard Thresholding (IMHT) - CT

Input: $\mathbf{c}$, $\mathbf{A}$, s, K, η and q
Output: $\hat{\mathbf{u}}$

1: Initialize $\mathbf{u}_1$.
2: **for** $k \leftarrow 1$ to K **do**
3: Compute the entries of $\mathbf{J}(\mathbf{u}_k)$ using (12) for each $i \in \{1,2,\ldots,m\}$ and $j \in \{1,2,\ldots,n\}$ with $\mathbf{u} = \mathbf{u}_k$.
4: Update

$$\mathbf{u}_{k+1} = \mathbf{\Pi}_{u_{th},s}(\mathbf{u}_k - \eta\mathbf{J}(\mathbf{u}_k)\left(\mathbf{c} - \Phi(\mathbf{u}_k)\right)). \quad (13)$$

5: **end for**
6: Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{K+1}$.

In this subsection, we propose two algorithms whose basic constituents are Algorithm 1 (GD-CT) or Algorithm 2 (IMHT-CT), and are capable of jointly estimating q and $\mathbf{u}$.

1) Block Co-ordinate Descent (BCD) - CT Algorithm: When q is unknown, we modify the overall optimization problem from (7) as

$$\hat{\mathbf{u}}, \hat{q} = \underset{\mathbf{u},q}{\arg\min} \ \frac{1}{2}\|\boldsymbol{\epsilon}\|_2^2 + \kappa\left\|(1+q)^{-\mathbf{u}}\right\|_1 \quad \text{s.t. } q \in \mathcal{X}, \ (14)$$

where $\mathcal{X}$ is the convex box constraint set used to restrict the values that q can take. Since q is a scalar, the box constraint is an interval on the real line. The block co-ordinate descent (BCD) algorithm is well suited for problems where the co-ordinates or variables of optimization show *block-commonality* [50]. From (8), it is clear that the multiplicative factor is common across the entries of the gradient. Therefore, the maximum advantage is obtained in terms of the processing efficiency when the *block-component*s are defined as $\boldsymbol{\alpha} = [\mathbf{u}; q]$, where $\boldsymbol{\alpha}$ denotes the overall parameter vector. The vector $\mathbf{u}$ is considered as one block, and the scalar q is considered as the another block. The BCD algorithm for estimating the values of $\mathbf{u}$ and q in a cyclic fashion is called as the *BCD-CT* algorithm. Denoting the maximum number of iterations by M, the BCD-CT algorithm is described in Algorithm 3.

In Algorithm 3 (BCD-CT), the solution to (15) in each iteration can be obtained using either Algorithm 1 (GD-CT) or Algorithm 2 (IMHT-CT). When the information on whether GD-CT or IMHT-CT algorithms is used is important, we use either BCD-CT-G or BCD-CT-I, respectively.

The projected gradient descent (PGD) algorithm is used to solve sub-problem (16) involving the parameter q in each of the BCD iterations. Let L denote the maximum number of iterations in the PGD algorithm, $\Pi_{\mathcal{X}}(w)$ denote the projection of the point $w \in \mathbb{R}$ into the set $\mathcal{X} \subseteq \mathbb{R}$, β be the step size and $g(q = q_{k,l}; \mathbf{u} = \mathbf{u}_k)$ denote the gradient of objective function in (16) with respect to q evaluated at $q = q_{k,l}$ and $\mathbf{u} = \mathbf{u}_k$. It is easy to show that $g(q; \mathbf{u})$ can be computed using

$$g(q; \mathbf{u}) = \sum_{i=1}^{m}\left(\frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{\log(1+q)} + c_i\right)$$

Algorithm 3 Block Co-Ordinate Descent (BCD) - CT

Input: $\mathbf{c}$, $\mathbf{A}$, M, K, L, κ, η and β
Output: $\hat{\mathbf{u}}$, $\hat{q}$

1: Initialize $\mathbf{u}_1$ and q_1.
2: **for** $k \leftarrow 1$ to M **do**
3: Update

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \frac{1}{2}\left\| \frac{\log\left(\mathbf{A}(1+q_k)^{-\mathbf{u}}\right)}{\log(1+q_k)} + \mathbf{c} \right\|_2^2 + \kappa \left\|(1+q_k)^{-\mathbf{u}}\right\|_1 \quad (15)$$

$$q_{k+1} = \arg\min_{q\in\mathcal{X}} \frac{1}{2}\left\| \frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}_{k+1}}\right)}{\log(1+q)} + \mathbf{c} \right\|_2^2 + \kappa \left\|(1+q)^{-\mathbf{u}_{k+1}}\right\|_1 \quad (16)$$

4: **end for**
5: Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{M+1}$ and set the estimate of q, $\hat{q} = q_{M+1}$.

$$\times \left[-\frac{1}{\log(1+q)\mathbf{A}_i^T(1+q)^{-\mathbf{u}}} \sum_{j=1}^{n} \frac{A_{ij}u_j}{(1+q)^{u_j+1}} - \frac{\log\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)}{(1+q)(\log(1+q))^2} \right] - \sum_{j=1}^{n} \frac{\kappa u_j}{(1+q)^{u_j+1}}. \quad (17)$$

The PGD algorithm is described in Algorithm 4. The second derivative, $H(q)$, of the cost function in (14) w.r.t. q is

$$H(q) = \sum_{i=1}^{m} \left[\frac{1}{\log(1+q)\ \mathbf{A}_i^T(1+q)^{-\mathbf{u}}} \sum_{j=1}^{n} \frac{A_{ij}u_j}{(1+q)^{u_j+1}} + \frac{\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})}{(1+q)(\log(1+q))^2} \right]^2 + \left(c_i + \frac{\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})}{\log(1+q)} \right) \times \left[\frac{\sum_{j=1}^{n} A_{ij}u_j(u_j+1)(1+q)^{-(u_j+2)}}{\log(1+q)\ \mathbf{A}_i^T(1+q)^{-\mathbf{u}}} + \frac{\left(\log(\mathbf{A}_i^T(1+q)^{-\mathbf{u}})\right)^2 (\log(1+q)+2)}{(1+q)^2(\log(1+q))^3} + \frac{\left(\sum_{j=1}^{n} A_{ij}u_j(1+q)^{-(u_j+1)}\right)(\log(1+q)+1)}{(1+q)(\log(1+q))^2\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)} - \frac{\left(\sum_{j=1}^{n} A_{ij}u_j(1+q)^{-(u_j+1)}\right)^2}{\log(1+q)\left(\mathbf{A}_i^T(1+q)^{-\mathbf{u}}\right)^2} \right] + \sum_{j=1}^{n} \frac{\kappa u_j(u_j+1)}{(1+q)^{(u_j+2)}}. \quad (18)$$

Let $q \in \mathcal{X} \triangleq [m_q, M_q),\ m_q < M_q < \infty$. In a positive test where amplification occurs in each PCR cycle, $m_q > 0$. When $u_j < \infty$, $H(q)$ is upper bounded as

$$H(q) \le L_1' \triangleq m \left[\frac{nM_u}{\log(1+m_q)(1+M_q)^{-M_u}(1+m_q)^{m_u+1}} + \frac{\gamma'}{(1+m_q)(\log(1+m_q))^2} \right]^2 + m\left(c_{\max} + \frac{\gamma'}{\log(1+m_q)} \right) \times \left[\frac{n(M_u+1)^2(1+m_q)^{-(m_u+2)}}{\log(1+m_q)(1+M_q)^{-M_u}} + \frac{(\gamma')^2(\log(1+M_q)+2)}{(1+m_q)^2(\log(1+m_q))^3} + \frac{nM_u(1+m_q)^{-(m_u+1)}(\log(1+M_q)+1)}{(1+m_q)(\log(1+m_q))^2(1+M_q)^{-M_u}} \right] + \frac{n\kappa M_u(M_u+1)}{(1+m_q)^{(m_u+2)}}, \quad (19)$$

where $\gamma' \triangleq \max\left(|\log(1+M_q)^{-M_u}|, |\log(n(1+m_q)^{-m_u})|\right)$ and when $u_j = \infty$, as

$$H(q) \le L_2' \triangleq m\left[\frac{\gamma'}{(1+m_q)(\log(1+m_q))^2} \right]^2 + m\left(c_{\max} + \frac{\gamma'}{\log(1+m_q)} \right) \left[\frac{(\gamma')^2(\log(1+M_q)+2)}{(1+m_q)^2(\log(1+m_q))^3} \right]. \quad (20)$$

In summary, $H(q) \le L' \triangleq \max(L_1', L_2') < \infty$. Let the step size, β, be chosen such that $0 < \beta < 1/L'$. Using Theorem 1 for the sequence of scalar iterates $\{q_k\}$, in the context of gradient descent update step in (21), we can conclude that the iterates converge to a local minimizer, q^*.

Algorithm 4 Projected Gradient Descent (PGD)

Input: $\mathbf{c}$, $\mathbf{A}$, $\mathbf{u}_k$, q_k, L and β
Output: q_{k+1}

1: Initialize $q_{k,1} = q_k$.
2: **for** $l \leftarrow 1$ to L **do**
3: Compute $g(q_{k,l}; \mathbf{u}_k)$ using (17).
4: Update

$$q_{k,l+1} = \Pi_{\mathcal{X}}\left(q_{k,l} - \beta g(q_{k,l}; \mathbf{u}_k)\right). \quad (21)$$

5: **end for**
6: Set $q_{k+1} = q_{k,L+1}$.

2) Alternating Direction Method of Multipliers (ADMM) - CT Algorithm: The alternating direction method of multiplier (ADMM) procedure is another well-known parallel or distributed optimization framework [51]. The essence of the ADMM recipe lies in using the advantages of the *dual formulations* and the *augmented Lagrangian* [51]. A modification to (14) on these lines yields the required ADMM optimization problem, given by

$$\hat{\mathbf{u}}, \hat{\mathbf{w}}, \hat{q} = \arg\min_{\mathbf{u},\mathbf{w},q} \frac{1}{2}\|\boldsymbol{\epsilon}\|_2^2 + \kappa\left\|(1+q)^{-\mathbf{w}}\right\|_1 + \frac{\rho}{2}\|\mathbf{u} - \mathbf{w} + \boldsymbol{\mu}\|_2^2 \quad \text{s.t. } q \in \mathcal{X}, \quad (22)$$

where ρ is the penalty parameter, $\boldsymbol{\mu}$ denotes the dual variable and $\mathbf{w}$ is the auxiliary variable. The dual variable is the result of the re-parameterization which converts the standard ADMM problem into its *scaled dual form* [51]. The advantage of the dual formulation, along with the auxiliary variable, is that it allows for the main optimization problem to be decomposed into multiple, simpler sub-problems which are computationally efficient to solve. For example, from (22), at least 3 sub-problems involving $\mathbf{u}$, $\mathbf{w}$ and q can be observed. However, dual methods may have slow convergence rates. A penalty term, $\frac{\rho}{2}\|\mathbf{u} - \mathbf{w} + \boldsymbol{\mu}\|_2^2$, is added to convert the primal problem into a strongly convex objective to overcome this issue. Lastly, from (22), we note that $\boldsymbol{\mu} \to \mathbf{0}$ and $\hat{\mathbf{w}} = \hat{\mathbf{u}}$ on convergence.

Algorithm 5 Alternating Direction Method of Multipliers (ADMM) - CT

Input: $\mathbf{c}$, $\mathbf{A}$, N, K, L, κ, η and β

Output: $\hat{\mathbf{u}}$, $\hat{q}$

1: Initialize $\mathbf{u}_1$, $\mathbf{w}_1$, $\boldsymbol{\mu}_1$ and q_1.

2: **for** $k \leftarrow 1$ to N **do**

3: Update

$$\begin{aligned}
\mathbf{u}_{k+1} &= \arg\min_{\mathbf{u}} \ \frac{1}{2}\left\| \frac{\log\left(\mathbf{A}(1+q_k)^{-\mathbf{u}}\right)}{\log(1+q_k)} + \mathbf{c} \right\|_2^2 \\
&\quad + \frac{\rho}{2}||\mathbf{u} - \mathbf{w}_k + \boldsymbol{\mu}_k||_2^2 \\
q_{k+1} &= \arg\min_{q \in \mathcal{X}} \ \frac{1}{2}\left\| \frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}_{k+1}}\right)}{\log(1+q)} + \mathbf{c} \right\|_2^2 \\
&\quad + \kappa \left\| (1+q)^{-\mathbf{w}_k} \right\|_1 \\
\mathbf{w}_{k+1} &= \arg\min_{\mathbf{w}} \kappa \left\| (1+q_k)^{-\mathbf{w}} \right\|_1 + \frac{\rho}{2}\|\mathbf{u}_k - \mathbf{w} + \boldsymbol{\mu}_k\|_2^2 \\
\boldsymbol{\mu}_{k+1} &= \boldsymbol{\mu}_k + (\mathbf{u}_{k+1} - \mathbf{w}_{k+1}).
\end{aligned} \tag{23}$$

4: **end for**

5: Set the estimate of individual CTs, $\hat{\mathbf{u}} = \mathbf{u}_{N+1}$ and set the estimate of q, $\hat{q} = q_{N+1}$.

Let N denote the maximum number of iterations. The ADMM algorithm for estimating $\mathbf{u}$ and q given $\mathbf{c}$ and $\mathbf{A}$, called *ADMM-CT* algorithm is described in Algorithm 5.

In Algorithm 5, we note that the sub-problem in (23) is convex in $\mathbf{w}$. Hence, it can be solved using any convex optimization package (e.g., CVX [52]). As stated earlier, the sub-problem involving optimization over $\mathbf{u}$ can be solved using GD (Algorithm 1) with a term $\rho(\mathbf{u} - \mathbf{w} + \boldsymbol{\mu})$ added to the gradient in (8). Similarly, the sub-problem of estimating q_{k+1} can be solved using Algorithm 4.

We illustrate the advantage of the ADMM-CT algorithm over the BCD-CT algorithm (with GD algorithm for optimizing $\mathbf{u}$) in terms of the *quality* of the final solution in Fig. 1. For the current discussion, the quality of the final solution is measured as the *gap* between the estimates of the individual CTs for the positive and negative samples. As mentioned in Sec. I-A, the individual CTs for a positive sample is finite and is between $15 - 35$. However, for a negative sample, the individual CT is ∞. Therefore, from the perspective of the recovery algorithms, a larger margin between the estimate of the individual CTs for positive and negative samples is desirable. In addition, a larger margin translates to the robustness of the algorithm to the choice of the decision thresholds. From Fig. 1, we see that there is a well-defined and more prominent margin between the individual CT estimates of the positive and negative samples obtained using the ADMM-CT algorithm compared to the estimates obtained from the BCD-CT algorithm.

D. Convergence to Local Minima

In the optimization problems described above, due to the presence of log and exponential-to-q terms, the objective function is non-convex in $\mathbf{u}$. To overcome the problem of convergence to the local minima, the BCD-CT and ADMM-CT algorithms are run multiple times with *Gaussian randomization* (GR) i.e., the algorithm is run with different initial conditions and the estimates with the lowest objective function value are chosen as the final estimates of q and $\mathbf{u}$ [53]. The term *Gaussian* in GR refers to the fact that the initial samples are drawn from a Gaussian distribution with certain mean and variance. However, our simulations show that the performance of our recovery algorithms is insensitive to the specific distribution used for randomization (see Section IV for more details). The false positive rate (FPR) and false negative rate (FNR) performance improvement obtained from GR in the ADMM-CT algorithm is shown in Fig. 2.

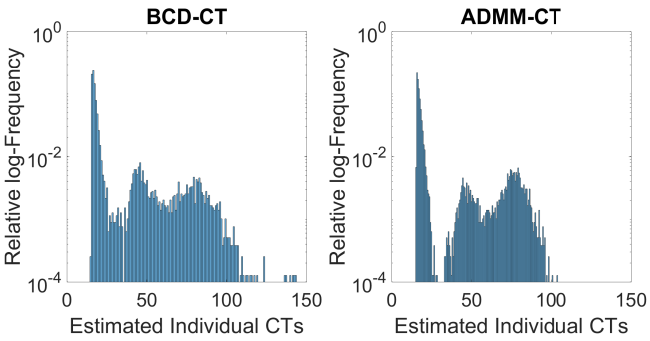

Fig. 1. Comparison of ADMM-CT and BCD-CT algorithms in terms of the distribution of the estimated individual CTs.

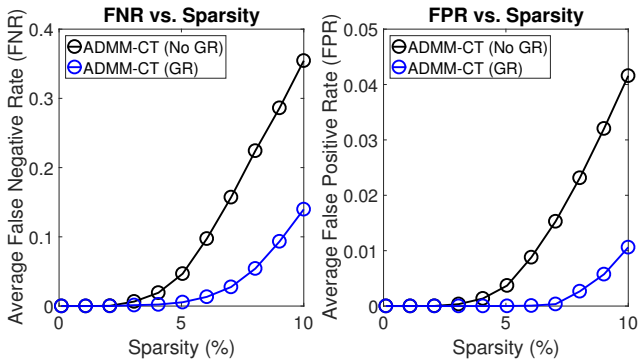

Fig. 2. Comparison of ADMM-CT algorithm with and without Gaussian randomization to overcome local optima.

We note that the complexity of the algorithms is not a major concern, as the RT-qPCR test itself takes several hours to run. All of the recovery algorithms presented here run in a few seconds on a computer, and therefore their relative complexity is not important for this application.

IV. Results and Observations

For generating the simulation data, we use $q = 0.95$ and the Kirkman matrix with $m = 45$ rows and $n = 105$ columns as the pooling matrix, unless stated otherwise.[6] We choose $m = 45$ because a standard PCR plate can accommodate a maximum of 93 test samples at a time (See Sec. I-B).

[6]We obtain Kirkman matrix of size 45×105 by selecting first 105 columns from the full sized 45×285 Kirkman matrix. Other options which can be used are to select columns uniformly randomly without replacement; or select/drop a column if it does not increase/decrease the mutual coherence, etc.

Therefore, using $m = 45$, one can perform two pooled tests in parallel with a 93-capacity plate, making it practically useful in terms of reducing wastage due to empty wells. The indices of nonzero viral loads in $\mathbf{x} \in \mathbb{N}^{105\times 1}$ are picked uniformly at random and stored in an index set $\mathcal{NZI}$ with cardinality $|\mathcal{NZI}| = \lceil ns_x/100 \rceil$, where s_x (%) is the prevalence rate. The viral load in a negative sample is zero. The positive entries of the initial viral-load vector, $[\mathbf{x}]_j$, are drawn from the Poisson(λ_j) distribution where $\lambda_j \sim$ Unif$(100, 10^6)$, for $j \in \mathcal{NZI}$. Finally, using the relationship in (2), the pooled CT vector, $\mathbf{c} \in \mathbb{R}_+^{45\times 1}$ is generated. In the distribution of the hyper-parameter λ_j, the lower and upper limits are the average least and highest viral load (or equivalently, cDNA/ RNA particles) in a positive sample.

We use CT $= 45$ as the decision threshold, i.e., samples with estimated CT > 45 are declared as negative. This parameter can be easily varied based on the actual number of PCR cycles conducted in the RT-qPCR machine. The maximum number of iterations are set as $K = L = 500$ and $M = N = 150$. The efficiency factor is assumed to belong to the set $\mathcal{X} = [0.5, 1.0)$ in Algorithm 4 and the CT threshold value is set as $u_{th} = 100$ in Algorithm 2. Also, in Algorithm 2, the true value of s is assumed to be known in our simulations. In practice, s can be estimated using the prevalence rate curve (e.g., see Fig. 3 in Sec. IV-A). In the initialization step of the algorithms, the starting points $\mathbf{u}_1$ and $\mathbf{w}_1$ are drawn i.i.d. from a Laplace distribution with mean 75 and shape parameter 10 and $q_1 \sim$ Unif$(\mathcal{X})$. Based on the simulations, we note that the distribution function and the associated parameters used do not affect the results much. For example, choosing Normal$(50, 10)$, Laplace$(50, 10)$, Unif$(10, 100)$ etc. give similar results. Finally, the ADMM-CT penalty parameter, $\rho = 0.01$, the regularization parameter, $\kappa = 10^{-3}$, and the step sizes η and β of 0.01 each are chosen via cross-validation. We have noticed in our simulations that the value of κ can be set very close to 0 without significantly changing the performance. This is because, as mentioned in Sec. III-A1, after the COMP stage, the problem is often over-determined. Therefore, the relative importance of the sparsity promoting term is low. Also, since the viral loads are non-negative, the ℓ_1 regularization is not necessary to ensure a unique solution to (7) [54], [55].

A. Advantage of Pool Testing: An Empirical Evaluation Using Covid-19 Data

We start by showing the advantage of using pool testing from the publicly available history of Covid-19 test numbers. Although we focus on non-adaptive pool testing with quantitative measurements in this work, in this experiment, we include one adaptive pool testing and two non-adaptive settings: a binary model and our quantitative model. We consider data from six Indian states: Karnataka, Kerala, Tamil Nadu, Maharashtra, Delhi, and Uttar Pradesh. The data used are primarily sourced by the Indian Council of Medical Research (ICMR), New Delhi, India,[7] and aggregated by a third-party website: https://www.covid19india.org/. As stated in Sec. I, the symptomatic percentage is set equal to 20%. The number of primary contacts per symptomatic individual are considered to be 4. These estimates can be further refined using population density information, data collected from contact tracing applications, etc. Further, the primary contacts of the symptomatic individuals are assumed to be more likely (by a factor of 4) to have the disease. In our analysis, following the standard protocols and triage processes, pool-testing is applied only on asymptomatic individuals and non-primary contacts; symptomatic individuals and primary contacts are tested individually. Dorfman adaptive testing with optimum pool size requires $2\sqrt{(n'k')}$ tests, where n' represents the number of non-symptomatic and non-primary contacts tested, and k' denotes the number of positive cases who are non-symptomatic and non-primary contacts [42]. In addition, we compute the counting bound, which is a lower bound on the number of tests under a binary testing model, as CB $= k' \log_2(\frac{n'}{k'})$ [42]. From [42], the DD algorithm with near-constant column weight pooling matrix design requires roughly $\frac{\text{CB}}{0.45}$ tests for identifying all the sick individuals.

Finally, the total number of tests required by the non-adaptive pool-testing method using deterministic matrix designs like Kirkman and Euler with quantitative measurements is computed empirically, as follows. A set of pooling matrix designs are compiled in $\mathcal{M} = \{\text{Euler} - (15 \times 25), \ldots, \text{Kirkman} - (45 \times 285), \ldots, \text{Kirkman} - (93 \times 1240), \text{Euler} - (361 \times 6859), \ldots\}$. We consider 26 Kirkman and 26 Euler-based design matrices. Thus, the cardinality of the set is $|\mathcal{M}| = 52$. The set is further extended as follows: Denote the *testing rate* of a pooling matrix of size $R \times C$ as $\xi = R/C$ where, C denotes the number of individuals tested, and R denotes the number of tests. Smaller column-truncated matrices are constructed by dropping the last few columns [24] to obtain matrices with rates $[\xi_{\text{round}} : 0.1 : 0.9]$ where ξ_{round} is the value obtained by rounding up 10ξ to the next integer and then dividing by 10. That is, if $\xi = 0.33$, $\xi_{\text{round}} = 0.4$. In this way, we obtain a total of 474 deterministic test matrices.

For each pooling matrix in the set $\mathcal{M}$, a 1000 Monte Carlo run experiment determines the maximum prevalence rate post which our ADMM-CT recovery algorithm makes either a false positive (FP) or a false negative (FN) error. This maximum prevalence rate is denoted by k_e for the given pooling matrix and is added into a look-up table. The non-symptomatic and non-primary contact prevalence rate is computed using the parameters described above. The best rate matrix design is the matrix with the lowest rate whose k_e exceeds the given prevalence rate. The rate thus obtained multiplied by n' gives the number of tests needed if non-adaptive pool-testing with our recovery algorithms and the optimum test matrix from the set $\mathcal{M}$ is used.

We then add the individual tests conducted on symptomatic and on primary contacts to obtain the total number of tests required by each of the three methods: Dorfman with optimal pool size (*Optimized Dorfman*), DD with the near-constant column weight design (*Achievable, DD*) and our approach (the ADMM-CT algorithm). The comparison of the cumulative number of tests from April 2020 till July 2021 is shown in Fig. 3 for all the 6 Indian states, along with the prevalence rate trend over the same duration.

From Fig. 3, the cumulative number of tests required by us-

[7]ICMR: https://www.icmr.gov.in/

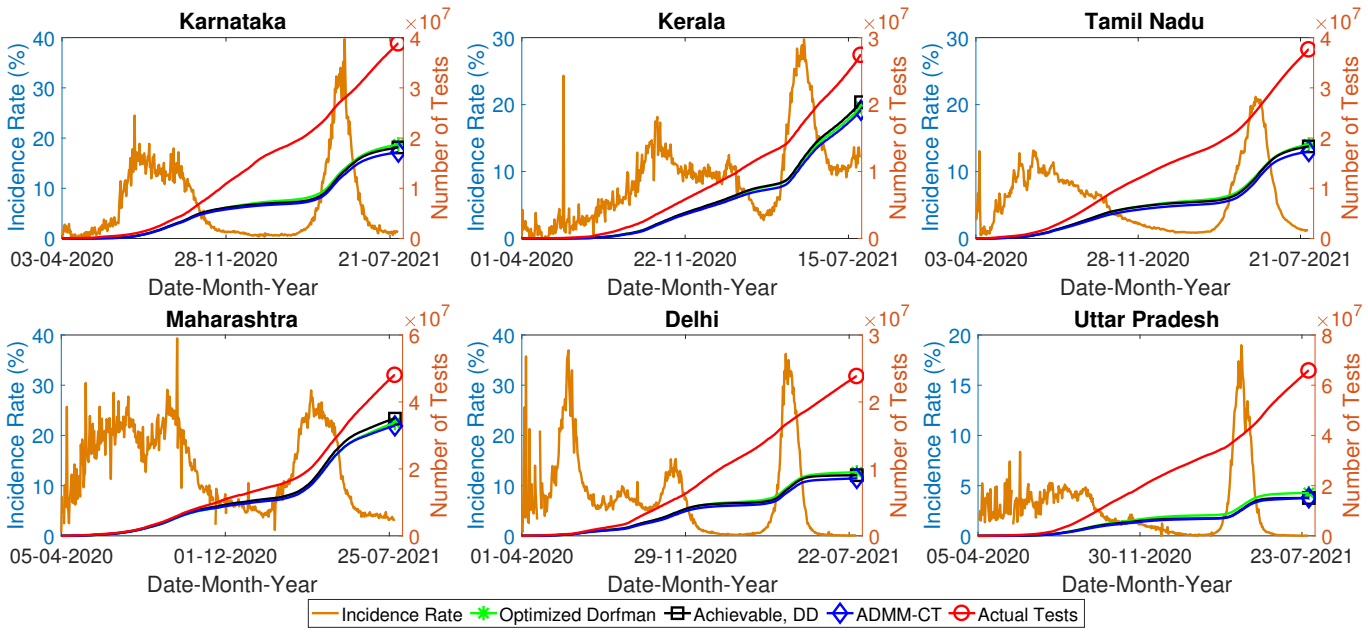

Fig. 3. The prevalence rate trend and the comparison of the cumulative number of tests required by *Optimized Dorfman*, *Achievable, DD* and *ADMM-CT* algorithms for 6 Indian states along with actual cumulative tests conducted from April 2020 till July 2021.

ing non-adaptive pool testing with quantitative measurements and the ADMM-CT algorithm for recovery (blue curve in Fig. 3) is lower than that obtained by using non-adaptive pool testing with binary measurements (black curve), and this is further better than the adaptive testing numbers (green curve). Finally, pool testing methods have significant advantage compared to individual testing (red curve) when the prevalence rates are low (also see Fig. 5 and the associated discussion.) The total number of tests saved translates to (resource) cost and time saving. For instance, using the numbers for the state of Karnataka, the cumulative number of tests saved over the past 1.5 years, if a pool testing method (e.g., the ADMM-CT approach) is used compared to the individual testing is ~ 21.6 million tests. Using the nominal cost per RT-qPCR test as ≈ 14 USD, we obtain a cost-saving of ~ 302.46 million USD. Therefore, pool testing methods have a significant advantage as compared to individual testing. Further, our ADMM-CT approach requires ~ 1.55 million, and ~ 0.93 million fewer tests than the optimized Dorfman and the achievable tests required by DD approaches, respectively. Under the above-mentioned RT-qPCR cost estimate, the cost savings obtained by using the ADMM-CT approach instead of optimized Dorfman and the tests required by DD are ~ 21.78 million and ~ 13.05 million USD, respectively. Thus, our approach is better in terms of the cost savings among the other pool testing methods considered.

B. A Practical Pool Test Protocol

Based on the previous discussion about the benefits of non-adaptive group testing with quantitative measurements, in this sub-section, we address the following aspects:

1) We demonstrate that using the local prevalence rates to design the tests is useful in practice, by using the actual testing numbers and positivity rates from 6 different states of India during different stages of the pandemic. The advantage of the using local prevalence rates was also discussed in [56].
2) We empirically characterize the testing rates achievable using deterministic pooling matrices for different prevalence rates. That is, we provide insight into the question of which deterministic pooling matrix should be chosen to guarantee a near-zero errors at each prevalence rate.
3) We collect empirical results on the prevalence rates at which adaptive, non-adaptive binary measurements-based, and non-adaptive quantitative measurements-based group testing perform the best, in terms of the testing rates achieved.

The authors in [56] give a lower bound on the number of tests required given the heterogeneity profile (i.e., prevalence rate, risk profile, contact maps etc.) of the local population. In particular, they focus on the two-stage group testing algorithms like Bernoulli sampling, Dorfman, constant tests per sample, the doubly constant algorithm etc., with random pooling matrices. Our work complements this approach, since we focus on single-stage group testing with deterministic pooling matrices.

To this end, first, we empirically show that using the *local* (e.g., state-wise) prevalence rate to select the pooling matrix is advantageous over using the *global* prevalence rate using actual testing data. The total number of tests conducted across the 6 states are aggregated to obtain the total number of individual tests conducted and the total number of tests required if our algorithm is used. Also, using the raw aggregated data from these 6 states, a *global* prevalence rate is computed. Finally, the number of tests required by our method using the global prevalence rate information is also computed. The comparison of the cumulative number of tests required under individual testing and when the local/global prevalence rates are used is shown, along with the global prevalence rate from April 2020 till July 2021 in Fig. 4.

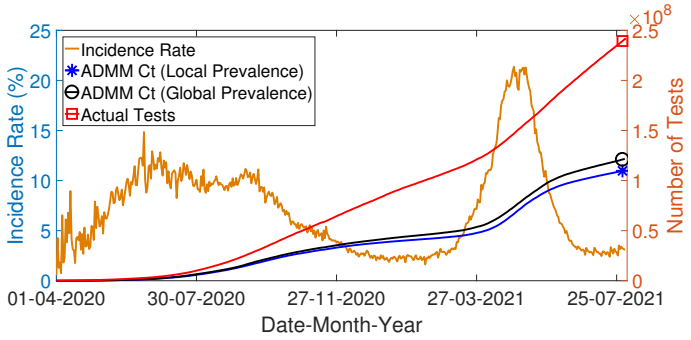

Fig. 4. Comparison of the cumulative number of tests required by our approach using the local prevalence rates versus using the aggregated numbers across 6 states to estimate a global prevalence rate, along with actual cumulative tests conducted and the variation of aggregated positivity rate over time from April 2020 till July 2021.

It can be observed from Fig. 4 that there is a clear advantage of using local prevalence rate information. Usage of the local prevalence rate to design the pool test saves ~ 11.44 million tests and hence, a cost-saving of $\sim$ 160.21 million USD, compared to the global prevalence rate based design.

Next, we illustrate how choosing the test matrix based on the prevalence rate helps. Fig. 5 shows the testing rate achieved by selecting the best member of the Euler and Kirkman family of matrices at each prevalence rate. The figure also shows the testing rate achieved by the Dorfman method with the optimum pool size for each prevalence rate and the rate achieved by the DD algorithm, computed approximately from the counting bound as mentioned earlier.

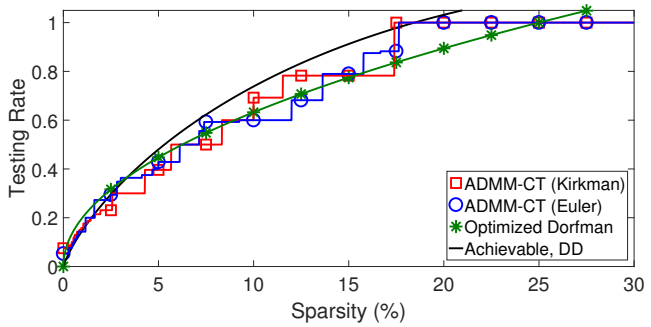

Fig. 5. Comparison of two-stage Dorfman testing rate with single-stage rates and optimum choice of deterministic matrices: Kirkman and Euler types available across the prevalence rates, i.e., sparsity levels (%).

From Fig. 5, we can make several interesting observations. First, pool testing is beneficial compared to individual testing when the prevalence rate is $< 25\%$. Second, non-adaptive pool testing methods have an advantage over the adaptive Dorfman style testing when the prevalence rate is $< 13.5\%$. Third, the quantitative (or CT) measurement-based method has an advantage over DD, a binary model-based approach, when the prevalence rate $\sim 1.6 - 17.5\%$. Fourth, the adaptive Dorfman and the quantitative measurement methods provide similar testing rates when the prevalence rate $\sim 0.25 - 1.6\%$. Further from Fig. 5, we note that the testing rates shown in the blue and red curves, corresponding to Euler and Kirkman designs, respectively, overlap at some prevalence rates. Thus, it indicates the existence of multiple non-adaptive deterministic pooling strategies.

C. Performance of the Algorithms

The goal of this subsection is to illustrate the performance of the proposed algorithms. All the results are averaged over $10,000$ Monte-Carlo runs. We use the following metrics: average NMSE in the CT values recovered, the average FN rate (FNR), and the average FP rate (FPR), for the prevalence rate (sparsity) up to 10%. The NMSE is the mean squared error (MSE) between the estimate $\hat{\mathbf{u}}$ and the true vector $\mathbf{u}$ normalized by the mean ℓ_2 norm-squared of the true vector, i.e., NMSE $\triangleq \mathbb{E}\left[\|\mathbf{u}-\hat{\mathbf{u}}\|_2^2\right]/\mathbb{E}\left[\|\mathbf{u}\|_2^2\right]$. Ignoring the role of the infinities in the computation of ℓ_2 norm of the CT vector, we note that the average ℓ_2 norm of the CT vector increases as the number of defective items in the population increases, thereby proportionally scaling un-normalized error metrics like MSE. In order to remove this bias in the error performance when comparing across the sparsity levels, the NMSE is considered instead of the MSE. We compare the algorithms in two sets, the first where the efficiency factor, q is known, and the second set where q is unknown and has to be estimated. In both the sets, the comparison is performed across two scenarios: firstly, under the various noise standard deviations, $\sigma_\epsilon \in \{0, 3.0\}$ [41] at $q = 0.95$ and secondly, under different efficiency factors, $q \in \{0.5, 0.95\}$ at $\sigma_\epsilon = 0$. In addition, the algorithms are run with Gaussian randomization, as mentioned earlier.

We illustrate the NMSE, FNR and FPR performances of GD-CT and IMHT-CT algorithms in Fig. 6. From subplots (B) and (C) in Fig. 6, we see that the FPR and FNR performances of both GD-CT and IMHT-CT algorithms are not affected by the noise levels. Further, the FNR and FPR performances of the GD-CT algorithm are better than that of the IMHT-CT algorithm. From the subplot (A), we see that the NMSE curves shift up when the noise $\sigma_\epsilon > 0$. From the subplots (B), (C), (E) and (F) in Fig. 6, we can observe that the FNR and FPR of the IMHT-CT algorithm show observable variation across the efficiency factors, whereas, the GD-CT algorithm shows relatively low variation. Also, the GD-CT algorithm's FPR is better, and the FNR is similar to that of the IMHT-CT algorithm. From the subplot (D), the NMSE performance of GD-CT across efficiency factors is better than that of the IMHT-CT algorithm. Finally, the relative variation in the performance of algorithms across the two efficiency factors (q) is $\approx 20\%$ for the IMHT-CT algorithm and $\approx 9\%$ for the GD-CT algorithm at sparsity level of 10%. Thus, the performance of the GD-CT algorithm is more stable with respect to variations in the efficiency factor.

The update step in (13) of the IMHT-CT algorithm executes a projection step, namely, the mirrored-hard thresholding operation, $\mathbf{\Pi}_{u_{th},s}(\mathbf{u})$. Compared to the GD-CT algorithm where iterates are computed optimally using the direction of the negative gradient, the projection step in IMHT-CT algorithm does not guarantee that the projected iterate exactly satisfies the optimality condition. Therefore, there is a trade-off between

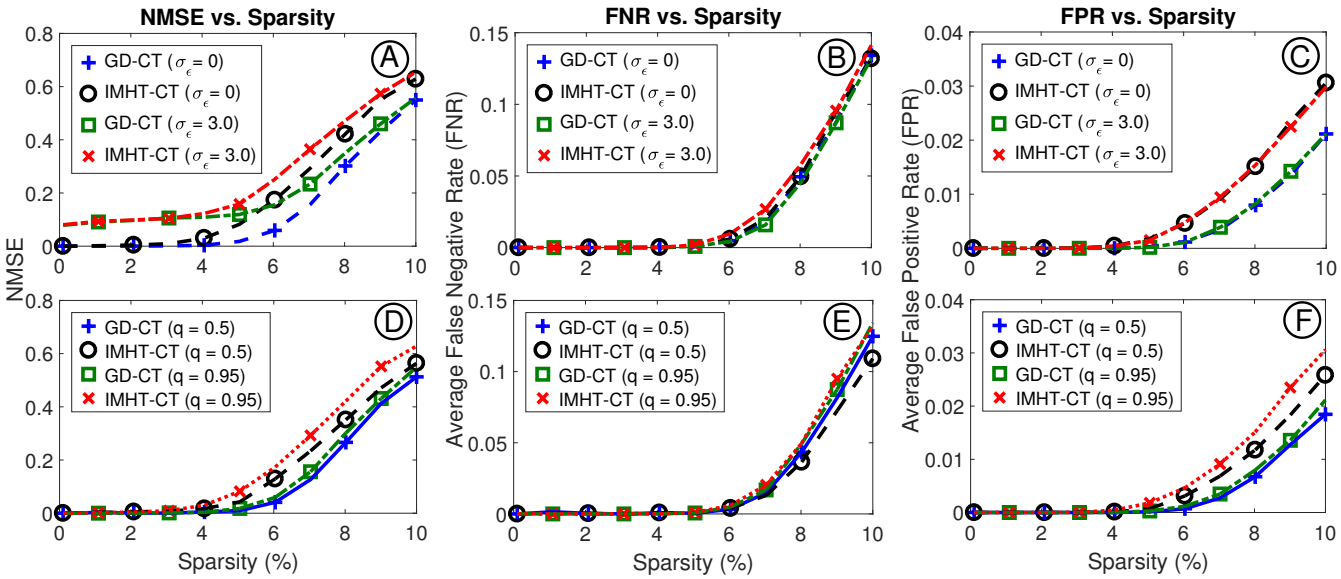

Fig. 6. Comparison of the NMSE, average FNR and average FPR performance across GD and IMHT algorithms over different sparsity (%) levels at various noise levels and efficiency factors. The plots in the top row compares the algorithm performances at various noise standard deviations, σ_ϵ, and the plots in the bottom row compares the performances at various efficiency factors, q, used to generate the data.

ensuring that the iterates are feasible (as in IMHT-CT) vs. the iterates satisfying the optimality condition (as in GD-CT). Overall, in the balance, the GD-CT algorithm outperforms the IMHT-CT algorithm. Next, we consider the case where the

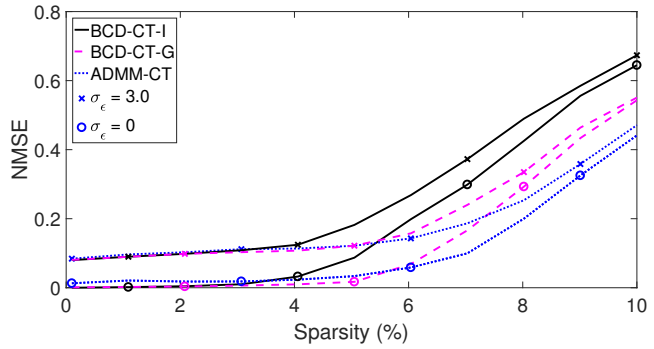

Fig. 7. Comparison of the average NMSE across ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms over different sparsity (%) and noise levels.

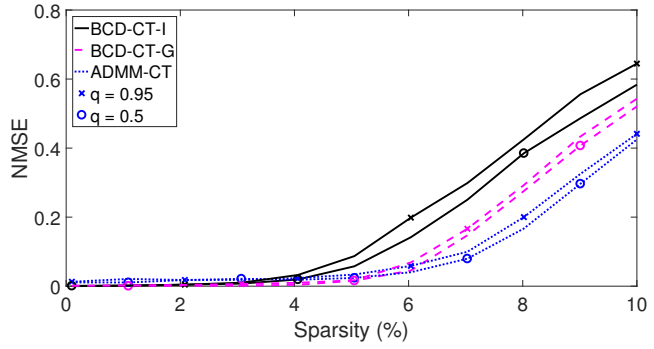

Fig. 8. Comparison of the average NMSE across ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms for different sparsity (%) and efficiency factors.

efficiency factor, q, is unknown and is estimated using the PGD algorithm. We compare the NMSE performances of the algorithms in Fig. 7 across various noise levels and in Fig. 8 across various efficiency factors.

From Figs. 7 and 8, the NMSE performance of the ADMM-CT algorithm is the best, followed by that of BCD-CT-G and BCD-CT-I algorithms. Although ADMM-CT shows a nonzero NMSE at low sparsity levels, there is no effect on the FPR and FNR performances, as we shall see later. Further, from Fig. 7, the NMSE curves shift up as the noise level increases. In addition, from Fig. 8, we see that the performance at $q = 0.5$ is slightly better with relatively notable difference for the BCD-CT-I algorithm. Thus, in line with our previous observations from Fig. 6, the IMHT based algorithms exhibit a wider variation in performance across different values of q compared to the GD based algorithms, even when q is unknown and is estimated from the observed CT values.

We define the mean squared error between q and its estimate, $\hat{q}$, as $\text{MSE}_q \triangleq \mathbb{E}[|q - \hat{q}|^2]$. When the sparsity $< 5\%$, we observed that $\text{MSE}_q \approx 0.052, 0.101$ and 0.26 for the $\sigma_\epsilon = 0, 1.0$ and 3.0, respectively. Also, while small errors in the estimate of q do not significantly affect the algorithm's FPR or FNR, larger errors cause the algorithm to make FP or FN errors at lower sparsity levels (see Table II in Sec. IV-D).

Lastly, we compare the FNR and FPR performances of the algorithms in Fig. 9 across various noise levels and efficiency factors. We include the performance of binary measurement-based algorithms: COMP and DD.

We can observe from the subplots (A) and (C) in Fig. 9 that the FNR of ADMM-CT, BCD-CT-I, and BCD-CT-G are similar. The FPR of the ADMM-CT algorithm is the best, followed by BCD-CT-G and BCD-CT-I algorithms, as seen in the subplots (B) and (D) across the noise levels and efficiency factors, respectively. Further, the FNR and FPR performances are not significantly different across the efficiency factors for each algorithm despite the variations

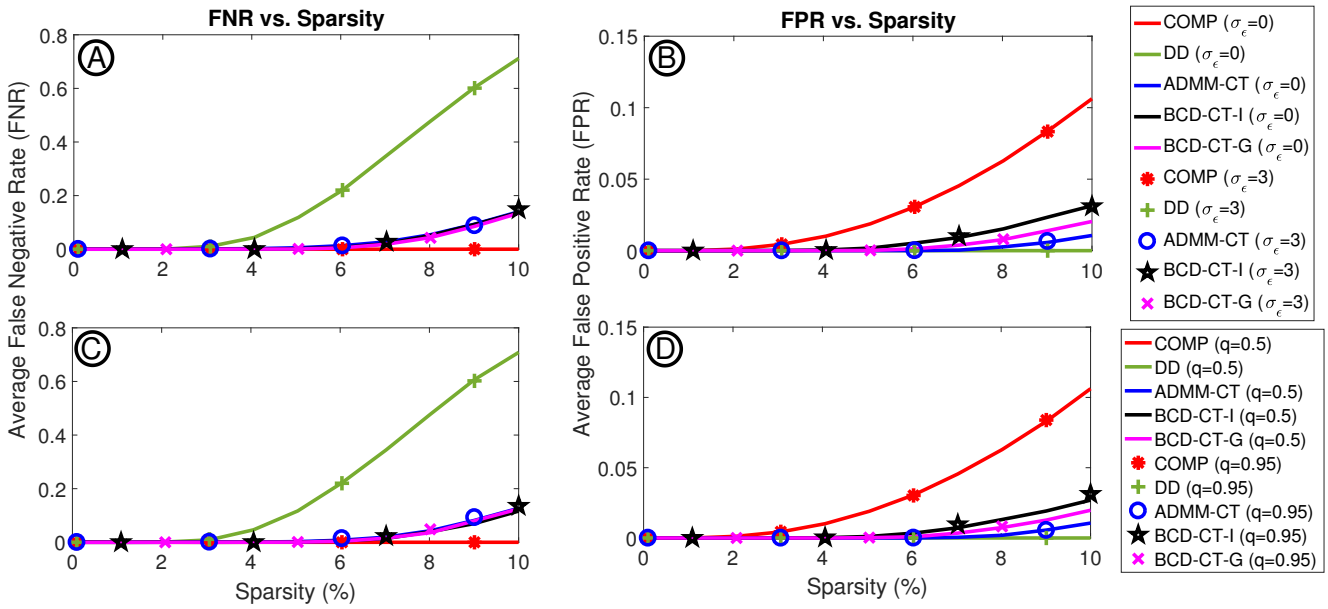

Fig. 9. Comparison of the average FNR and average FPR across ADMM-CT, BCD-CT-G, and BCD-CT-I algorithms over different sparsity (%) at different noise levels (top row) and efficiency factors (bottom row).

observed in NMSE plots. The ADMM-CT approach is able to match the FNR of COMP while simultaneously matching the FPR of the DD algorithms. Although COMP and DD optimize for FNR and FPR, respectively, the corresponding FPR and FNR performances are poor. Thus, this highlights the advantage of quantitative measurement-based algorithms over binary pool testing algorithms like COMP and DD. Also, pooled CT-based algorithms can estimate the individual CTs, which could be independently valuable in practice.

In Algorithm 3, the BCD iterations are not tied together, since each block is optimized by independently solving the sub-problems in a cyclic manner (see (15) and (16)). In contrast, the iterates are better coupled in the ADMM-CT algorithm. In Algorithm 5, 1) an auxiliary variable, $\mathbf{w}_k$, similar in behavior to the individual-sample CTs, $\mathbf{u}_k$, is used; 2) the sub-problem that solves for $\mathbf{u}_k$ uses a previous value of q_k, whereas, the sub-problem which solves for $\mathbf{w}_k$ uses the updated $q_k \leftarrow q_{k+1}$; and 3) the auxiliary variables $\mathbf{w}_k$ and $\mathbf{u}_k$ are tied together using a dual variable, $\boldsymbol{\mu}_k$, in the ADMM-CT algorithm. Due to these reasons, the ADMM-CT outperforms the BCD-CT algorithm.

D. Comparison with CS-based Methods

In this final subsection, we compare the performance of the ADMM-CT algorithm with the existing CS-based algorithms in the literature [24], [29]. These previous studies directly apply CS-based recovery techniques (with or without an initial COMP stage) to the observation model in (2). Certain CS-based algorithms like LASSO are known to yield the optimal solution even in the non-linear regime, under certain conditions [57], [58]. In particular, the authors in [57] show that when the entries of the pooling matrix are drawn i.i.d. from $\sim \mathcal{N}(0,1)$, the performance of LASSO with non-linear measurements characterized by a non-linear mapping $g(\cdot)$ is asymptotically the same as if the measurements were of the linear form $c_i = \mu \mathbf{A}_i^T \mathbf{x} + \sigma \epsilon_i$, where $\mu \triangleq \mathbb{E}[\bar{\gamma} g(\bar{\gamma})]$, $\sigma \triangleq \mathbb{E}[(g(\bar{\gamma}) - \mu\bar{\gamma})^2]$ and $\bar{\gamma} \sim \mathcal{N}(0,1)$. Similarly, the authors in [58] derive upper and lower bounds for the MSE under a Poisson measurement model and when the entries of the pooling matrix satisfy certain boundedness conditions. The simulation results in this sub-section shows that our proposed approach is better under the non-asymptotic regime with a deterministic binary pooling matrix. Note that, in order to use (2) directly with CS-based algorithms, one needs to assume a nominal value of q to compute the left hand side of the measurement equation of the observed CT values [24]. We also mention that one can use cross-validation to choose the value of q, instead of assuming a nominal value.

We demonstrate the robustness of the ADMM-CT algorithm in the following two aspects. The first one is the robustness to the CT measurement noise. The second aspect is the robustness to the unknown efficiency factor, q. We denote the efficiency factor used for the CT-to-viral-load conversion by q_{conv}, which could be different from the actual q of the PCR process. We consider the minimum sparsity percentage (denoted by k_e) at which the recovery algorithm makes at least one FP or FN out of $10,000$ experiments as our metric for comparison. Table I summarizes the values of k_e for each of the recovery algorithms considered, at different noise (σ_ϵ) levels.

In CS-based algorithms, a threshold is fixed, below which the estimates are set to 0. We fix the threshold to balance out the FP and FN errors. Further, the hyper-parameters (regularization constants etc.) required by the CS based algorithms are chosen via cross-validation [24, S.III]. We compare against the NN-LASSO, NN-LS, NN-LAD, NN-OMP and SBL algorithms proposed for RT-PCR based pool testing for Covid-19 in the literature [24], [29]. From Table I, we note that as the noise level increases, k_e obtained by CS-based algorithms degrades. In particular, at practical noise levels [41], ADMM-CT has a clear advantage over the CS-based algorithms, even

TABLE I
COMPARISON OF k_e ACROSS RECOVERY ALGORITHMS AT VARIOUS NOISE VARIANCES

Noise σ_ϵ	0 (Noiseless)	0.1	0.5	1.0
NN-LASSO	3	1	1	1
NN-LS	3	1	1	1
NN-LAD	4	2	1	1
NN-OMP	4	1	1	1
SBL	8	4	2	1
ADMM-CT	4	4	4	4

TABLE II
COMPARISON OF k_e ACROSS RECOVERY ALGORITHMS AT VARIOUS PROCESS EFFICIENCY FACTORS, q, AT NOISE $\sigma_\epsilon = 0.1$ AND $q_{\text{CONV}} = 0.95$

q	0.99	0.95	0.9	0.8	0.65	0.5
NN-LASSO	1	1	1	1	1	1
NN-LS	1	1	1	1	1	1
NN-LAD	1	2	1	1	1	1
NN-OMP	1	1	1	1	1	1
SBL	4	4	4	3	3	3
ADMM-CT	4	4	4	4	4	4

though SBL outperforms ADMM-CT in the noiseless case.

Next, we summarize the effect of mismatch between the q and q_{conv} in Table II. Similar to the noise performance analysis, we use the metric k_e to demonstrate the robustness of different algorithms. For this purpose, we fix the $q_{\text{conv}} = 0.95$ as in [24] and vary the actual $q \in \{0.99, 0.95, 0.9, 0.8, 0.65, 0.5\}$ of the RT-qPCR process [32], [41]. From Table II, we note that as the mismatch between q and q_{conv} increases, the performance of the CS-based algorithms degrades. The ADMM-CT algorithm estimates both $\mathbf{u}$ and q, and hence, the parameter q_{conv} is not used by it. Although SBL is robust to mismatch in q, the ADMM-CT still has an advantage over the existing algorithms. We also observed that the NMSE in the CT values returned by SBL are of the order 0.5, while that for ADMM-CT are of the order $0.01 - 0.04$ at $q = 0.99$ and $k_e = 4$. Thus, the SBL algorithm performs poorer than ADMM-CT in terms of estimating the individual CT values. The ADMM-CT algorithm is robust across the values of PCR efficiency factors observed in practice, and outputs the individual CTs with low NMSE.

In summary, the proposed algorithms are robust to the CT measurement noise levels observed in practical RT-qPCR. Also, they do not require knowledge of the machine-specific parameters, and the performance is similar across the spectrum of PCR (amplification) efficiency factors seen in practice.

V. CONCLUSION

We considered the problem of sparse recovery in a non-adaptive pool testing given the quantitative CT-values obtained from a non-linear model for the RT-qPCR test used in Covid-19 detection and a deterministic binary-valued pooling matrix. We proposed two novel and robust sparse recovery algorithms: ADMM-CT and BCD-CT, to jointly estimate the individual CT values and the process efficiency parameter, given the pooled CTs and the pooling matrix. Gradient descent and iterative mirrored hard-thresholding algorithms were used to solve the sub-problem involving the vector of individual CTs. Similarly, a projected gradient descent method was used to estimate the process efficiency parameter. We empirically showed the advantages of pool testing and, more specifically, that of non-adaptive pool testing with quantitative outcomes, using actual testing-numbers datasets. We also illustrated the algorithms' NMSE, false-positive rate, and false-negative rate performances. Finally, we demonstrated the advantage of the algorithms over the CS-based algorithms available in the literature. The future work could involve further theoretical analysis of the algorithms in a probably approximately correct (PAC) framework. In addition, the design of the decision logic (e.g., list decoding-based, clustering-based, etc.) and its effect on the testing rate are worth studying in greater depth.

APPENDIX A
KIRKMAN MATRIX DESIGNS

Consider $m = 15$ schoolgirls who walk out three abreast for seven days in succession. Is it possible to arrange them daily, so that no two girls walk abreast twice? This problem is called *Kirkman's schoolgirl problem*. Reverend T. P. Kirkman posed this question in 1850 and wondered about the existence of the solution [37, Example 2.76]. The answer to this problem is known as the *Kirkman Triple System* of order m, KTS(m), or more generally as a Steiner triple system of order m, or STS(m). An STS(m) consists of $\binom{m}{2}/3$ m-length boolean column vectors such that each member vector has exactly three 1s and the dot product of any two vectors is ≤ 1. The KTS(m) satisfies these conditions and in addition, possess *resolvability* property i.e., the member vectors can be arranged such that the sum of vectors from i to $i + m/3 - 1$ equals $\mathbf{1} \in \mathbb{R}^m$ for every $i \equiv 1$ modulo $m/3$. This property of KTS ensures that any l such group of vectors can be chosen from KTS to form an $m \times n$ *Kirkman matrix*, $n > m$ with $n = lm/3,\ 3 < l \leq (m-1)/2$, while keeping the number of 1s in each row fixed.

In addition, KTS and STS are solutions for other problems like social golfer problem [37]. For smaller m, the construction of Kirkman matrices can be done via greedy methods. For more information about Kirkman-based designs, see [24], [37]. A Kirkman matrix can be used for exact recovery of up to 3 positive samples [24]. From our simulations, we observe that the usage of quantitative measurements enable us to do better.

APPENDIX B
EULER MATRIX DESIGNS

The Euler matrix designs considered in this paper are based on *generalized Euler squares* (GES) [38]. In particular, we consider the following construction: let p be a prime; let $n \geq 1$ and $k \leq p^n$ denote the number of tests that an item participates in. If R is the number of items allowed in each test and d is a theoretical bound on the identifiable number of defective items, i.e., the d-disjunctness property is satisfied, then it is possible to construct a binary matrix of dimension $p^n k \times p^{n(r+1)}$, with $R = p^{nr}$ and $d = \lfloor \frac{k-1}{r} \rfloor + 1$, for $r \in \{1, 2\}$ and coherence at most r/k [38].

APPENDIX C
NP-HARDNESS OF (6)

In this sub-subsection, we show the NP-hardness of (6). We note that solving the optimization problem with ℓ_0 norm as given in (6) entails enumeration of all the candidate solutions and evaluating the cost function to find the solution.

We present a formal proof of NP-hardness below. First, we transform the cost function in (6) as

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \ \frac{1}{2}\left\|\boldsymbol{\epsilon}\right\|_2^2 + \psi \left\|(1+q)^{-\mathbf{u}}\right\|_0. \tag{24}$$

The above uses the fact that $\tau > 0$ and $\psi \triangleq \bar{\kappa}\tau$. Using (5), solving (24) is equivalent to solving

$$\min_{\mathbf{u}} \ \left\|(1+q)^{-\mathbf{u}}\right\|_0 \ \text{s.t.} \ \left\|\mathbf{c} + \frac{\log\left(\mathbf{A}(1+q)^{-\mathbf{u}}\right)}{\log(1+q)}\right\|_2 \leq \nu. \tag{25}$$

We define $\nu' \triangleq \nu\log(1+q)$, $\bar{\mathbf{x}} \triangleq (1+q)^{-\mathbf{u}}$ and $\bar{\mathbf{y}} \triangleq (1+q)^{-\mathbf{c}}$ and rewrite (25) to get

$$\min_{\bar{\mathbf{x}}} \ \left\|\bar{\mathbf{x}}\right\|_0 \ \text{s.t.} \ \left\|-\log(\bar{\mathbf{y}}) + \log(\mathbf{A}\bar{\mathbf{x}})\right\|_2 \leq \nu'. \tag{26}$$

It suffices to show the NP-hardness of (26) for, say, $\nu' = 0$ and the Kirkman pooling matrix. Therefore, we get

$$\min_{\bar{\mathbf{x}}} \ \left\|\bar{\mathbf{x}}\right\|_0 \ \text{s.t.} \ \bar{\mathbf{y}} = \mathbf{A}\bar{\mathbf{x}}. \tag{27}$$

We are now ready to use the proof steps similar to [40, Theorem 2.17 in Section 2.3] in the noiseless case. The overall idea is to transform a known NP-hard problem (e.g., exact cover by 3-set problem) to the problem in (27) in polynomial time. To this end, we start by taking $\bar{\mathbf{y}} = [1, 1, \ldots, 1]^T$. Then, using the constraint in (27), we get $\left\|\mathbf{A}\bar{\mathbf{x}}\right\|_0 = m$. From Appendix A, we use $n = lm/3,\ 3 < l \leq (m-1)/2$ to note that $n \leq m(m-1)/6 < \binom{m}{3},$ for $m \geq 3$. Therefore, the pooling matrix construction can be done in polynomial time. Further, Kirkman matrices emerge from KTS which are the solutions to the Kirkman's schoolgirls problem. Hence, the column sums of Kirkman matrices are equal to 3, implying that $\left\|\mathbf{A}\bar{\mathbf{x}}\right\|_0 \leq 3\left\|\bar{\mathbf{x}}\right\|_0$. In conclusion, we get $\left\|\bar{\mathbf{x}}\right\|_0 \geq m/3$. We can run the ℓ_0 normalization problem for the two cases: $\left\|\bar{\mathbf{x}}\right\|_0 = m/3$, and $\left\|\bar{\mathbf{x}}\right\|_0 > m/3$, to conclude that solving the ℓ_0 minimization problem enables one to solve the exact cover by 3-sets problem [40]. Therefore, the problem in (27) and hence, the original problem in (6) is NP-hard in general.

APPENDIX D
ŁOJASIEWICZ GRADIENT INEQUALITY

Given a real analytical function, f, the Łojasiewicz gradient inequality gives an upper bound for the difference between $f(\mathbf{x})$ and $f(\mathbf{x}^*)$, where $\mathbf{x}$ is any point in the open neighborhood of a critical point, $\mathbf{x}^*$, of f. More formally, a critical point $\mathbf{x}^*$ of function, f satisfies the Łojasiewicz gradient inequality if there exists a neighborhood $\mathcal{V}$, $0 \leq a < 1$, and $m,\ \epsilon > 0$ such that $\|\nabla f(\mathbf{x})\| \geq m|f(\mathbf{x}) - f(\mathbf{x}^*)|^a \ \forall\ \mathbf{x}$ in $\{\mathbf{x} \in \mathcal{V} : f(\mathbf{x}^*) < f(\mathbf{x}) < f(\mathbf{x}^*) + \epsilon\}$ [45]. This gradient inequality is useful in proving the global linear convergence of gradient descent-based algorithms [45], [59].

REFERENCES

[1] J. P. A. Ioannidis, "Infection fatality rate of COVID-19 inferred from seroprevalence data," *Bulletin of the World Health Organization*, vol. 99, no. 1, pp. 19–33F, Jan. 2021. [Online]. Available: https://www.who.int/bulletin/volumes/99/1/20-265892/en/

[2] V. Caturano, B. Manti *et al.*, "Estimating asymptomatic SARS-CoV-2 infections in a geographic area of low disease incidence," *BMC Infect. Dis.*, vol. 21, no. 350, Apr. 2021. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/variants/variant.html

[3] R. Subramanian, Q. He *et al.*, "Quantifying asymptomatic infection and transmission of COVID-19 in New York city using observed cases, serology, and testing capacity," *Proc. Natl. Acad. Sci. U.S.A (PNAS)*, vol. 118, no. 9, Mar. 2021.

[4] S. A. McDonald, F. Miura *et al.*, "Estimating the asymptomatic proportion of SARS-CoV-2 infection in the general population: Analysis of a nationwide serosurvey in the Netherlands," *medRxiv*, Mar. 2021. [Online]. Available: https://www.medrxiv.org/content/early/2021/03/31/2021.03.29.21254334

[5] S. Liu, H. Luo *et al.*, "Characteristics and associations with severity in COVID-19 patients: A multicentre cohort study from Jiangsu province, China," *The Lancet PrePrint*, Feb. 2020. [Online]. Available: https://ssrn.com/abstract=3548753

[6] K. Mizumoto, K. Kagaya *et al.*, "Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan," *Euro Surveill.*, vol. 25, no. 10, Feb. 2020.

[7] M. A. Johansson, T. M. Quandelacy *et al.*, "SARS-CoV-2 transmission from people without COVID-19 symptoms," *JAMA*, vol. 4, no. 1, pp. 1406–1407, Jan. 2021.

[8] Y. Bai, L. Yao *et al.*, "Presumed asymptomatic carrier transmission of COVID-19," *JAMA*, vol. 323, no. 14, pp. 1406–1407, Feb. 2020.

[9] E. Sanchez, "COVID-19 science: Why testing is so important," *American Heart Association*, Apr. 2020. [Online]. Available: https://www.heart.org/en/news/2020/04/02/covid-19-science-why-testing-is-so-important

[10] Centre for Disease Control and Prevention (CDC), "About Variants of the Virus that Causes COVID-19," *Centre for Disease Control and Prevention*, Feb 2022. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/variants/variant.html

[11] A. Atkeson, M. Droste *et al.*, "Economic benefits of COVID-19 screening tests," *medRxiv*, Oct. 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/11/01/2020.10.22.20217984

[12] B. Udugama, P. Kadhiresan *et al.*, "Diagnosing COVID-19: The disease and tools for detection," *ACS Nano.*, vol. 14, no. 4, pp. 3822–3835, Mar. 2020.

[13] J. H. Schefe, K. E. Lehmann *et al.*, "Quantitative real-time RT-PCR data analysis: current concepts and the novel "gene expressions' CT difference" formula," *J. Mol. Med.*, vol. 84, no. 11, pp. 901–910, Nov. 2006.

[14] B. W. Buchan, J. S. Hoff *et al.*, "Distribution of SARS-CoV-2 PCR cycle threshold values provide practical insight into overall and target-specific sensitivity among symptomatic patients," *Am. J. Clin. Pathol.*, vol. 154, no. 4, Sep. 2020.

[15] P. K. Chatki and A. G. Pise, "Perspective on the SARS-CoV-2 and its diagnostic tests," *Int. J. Curr. Res. (IJCRR)*, vol. 12, no. 19, Oct. 2020.

[16] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*, 2nd ed. World Scientific, 1999.

[17] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1880–1901, 2012.

[18] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Stat.*, vol. 14, no. 4, pp. 436–440, 1943.

[19] A. R. Bharti, S. L. Letendre *et al.*, "Malaria diagnosis by a polymerase chain reaction-based assay using a pooling strategy," *Americal Journal of Tropical Medicine and Hygiene*, vol. 81, no. 5, pp. 754–757, Nov. 2009.

[20] J. N. Eberhardt, N. P. Breuckmann *et al.*, "Multi-stage group testing improves efficiency of large-scale COVID-19 screening," *J. Clin. Virol.*, vol. 128, pp. 3775–3797, Jul. 2020.

[21] C. L. Chan, P. H. Che *et al.*, "Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms," in *Proc. Allerton Conf. on Commun., Control and Comput.*, 2011, pp. 1832–1839.

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

[23] J. Scarlett and O. Johnson, "Noisy non-adaptive group testing: A (near-)definite defectives approach," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3775–3797, 2020.

[24] S. Ghosh, R. Agarwal *et al.*, "A compressed sensing approach to pooled RT-PCR testing for COVID-19 detection," *IEEE Open J. Signal Process.*, 2021.

[25] I. Yelin, N. Aharony *et al.*, "Evaluation of COVID-19 RT-qPCR test in multi sample pools." *Clin. Infect. Dis.*, vol. 71, no. 16, pp. 2073–2078, Nov. 2020, first published in PrePrint on 27-March, 2020.

[26] B. Abdalhamid, C. R. Bilder *et al.*, "Assessment of specimen pooling to conserve SARS CoV-2 testing resources," *Am. J. Clin. Pathol.*, vol. 153, no. 6, pp. 715–718, Apr. 2020.

[27] A. Cohen, N. Shlezinger *et al.*, "Multi-level group testing with application to one-shot pooled COVID-19 tests," in *Proc. ICASSP*. IEEE, 2021, pp. 1030–1034.

[28] C. Schumacher and M. Täufer, "The statistics of noisy one-stage group testing in outbreaks," 2020. [Online]. Available: https://arxiv.org/abs/2012.02101

[29] H. B. Petersen, B. Bah *et al.*, "Practical high-throughput, non-adaptive and noise-robust SARS-CoV-2 testing," 2020. [Online]. Available: https://arxiv.org/abs/2007.09171

[30] J. Yi, R. Mudumbai *et al.*, "Low-cost and high-throughput testing of COVID-19 viruses and antibodies via compressed sensing: System concepts and computational experiments," 2020. [Online]. Available: https://arxiv.org/abs/2004.05759

[31] J. Yi, M. Cho *et al.*, "Error correction codes for COVID-19 virus and antibody testing: Using pooled testing to increase test reliability," 2020. [Online]. Available: https://arxiv.org/abs/2007.14919

[32] J. L. Gevertz, S. M. Dunn *et al.*, "Mathematical model of real-time PCR kinetics," *Biotechnol. Bioeng.*, vol. 92, no. 3, pp. 346–355, Nov. 2005.

[33] K. A. Waldstein, J. Yi *et al.*, "Use of compressed sensing to expedite high-throughput diagnostic testing for COVID-19 and beyond," *medRxiv*, 2021.

[34] H.-P. Wang, R. Gabrys *et al.*, "Tropical group testing," *aRxiv*, 01 2022.

[35] D. L. Donoho, "Compressed Sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[36] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[37] C. J. Colbourn and J. H. Dinitz, *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications)*. Chapman and Hall/CRC, 2006.

[38] P. Sasmal, P. Jampana *et al.*, "Sparse recovery guarantees for block orthogonal binary matrices constructed via generalized Euler squares," 2019. [Online]. Available: https://arxiv.org/abs/1907.07396

[39] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge Univ. Press, 1990.

[40] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Springer New York, 2013.

[41] C. B. F. Vogels, A. F. Brito *et al.*, "Analytical sensitivity and efficiency comparisons of SARS-CoV-2 RT–qPCR primer–probe sets," *Nat. Microbiol.*, vol. 5, no. 10, pp. 1299—-1305, Jul. 2020.

[42] M. Aldridge, O. Johnson *et al.*, *Group Testing: An Information Theory Perspective*. now, 2019.

[43] M. J. Abdi, *Cardinality Optimization Problems*. University of Birmingham, 2013. [Online]. Available: https://books.google.co.in/books?id=8d3-oAEACAAJ

[44] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge Univ. Press, 2014, pp. 150––166.

[45] J. D. Lee, M. Simchowitz *et al.*, "Gradient descent only converges to minimizers," in *Proc. 29th Ann. Conf. on Learn. Theory*, vol. 49. PMLR, 23–26 Jun 2016, pp. 1246–1257.

[46] I. Panageas and G. Piliouras, "Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions," in *Proc. 8th Innovations in Theor. Comp. Sc. (ITCS)*, vol. 67, Dagstuhl, Germany, 2017, pp. 2:1–2:12.

[47] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[48] Y. Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures*, G. Montavon, G. B. Orr *et al.*, Eds. Springer Berlin Heidelberg, 2012.

[49] T. Blumensath, "Compressed sensing with nonlinear observations and related nonlinear optimization problems," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3466–3474, 2013.

[50] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. USA: Prentice-Hall, Inc., 1989.

[51] S. Boyd, N. Parikh *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Machine Learning.*, vol. 3, no. 1, Jan. 2011.

[52] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[53] Z.-Q. Luo, W.-K. Ma *et al.*, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010.

[54] R. Kueng and P. Jung, "Robust nonnegative sparse recovery and the nullspace property of 0/1 measurements," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 689–703, 2018.

[55] M. A. Khajehnejad, A. G. Dimakis *et al.*, "Sparse recovery of nonnegative signals with minimal expansion," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 196–208, 2011.

[56] M. A. Attia, W.-T. Chang *et al.*, "Heterogeneity aware two-stage group testing," *IEEE Trans. Signal Process.*, vol. 69, pp. 3977–3990, 2021.

[57] C. Thrampoulidis, E. Abbasi *et al.*, "LASSO with non-linear measurements is equivalent to one with linear measurements," in *Proc. 28th Int. Conf. on Neur. Inf. Process. Syst. (NeurIPS)*, ser. NIPS '15, vol. 2. Cambridge, MA, USA: MIT Press, 2015, pp. 3420–3428.

[58] X. Jiang, G. Raskutti *et al.*, "Minimax optimal rates for poisson inverse problems with physical constraints," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4458–4474, 2015.

[59] H. Karimi, J. Nutini *et al.*, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," *arXiv e-prints*, Aug. 2016.

Sameera Bharadwaja H. (S'19) received his B. E. degree in Telecommunication Engineering from P. E. S. Institute of Technology, India in 2009 and his M. Tech. degree in Communication Systems from the Indian Institute of Technology Roorkee, India in 2011. From 2011 to 2013, he worked as a systems engineer in Innovation Labs, Tata Consultancy Services, Delhi, and in 2013, he joined Samsung Semiconductor India R&D (SSIR), Bangalore, India where he is currently working as an associate director. His work at SSIR involves design and development of machine learning algorithm modules for semiconductor FAB equipment diagnosis. He is currently pursuing his Ph. D. as an external registrant from SSIR at Signal Processing for Communications (SPC) lab in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. His research interests include group testing, sparse signal recovery and machine learning.

Chandra R. Murthy (S'03–M'06–SM'11) is a professor in the Electrical Communication Engineering department at the Indian Institute of Science, Bangalore, India. His research interests are in the areas of massive MIMO systems, beyond 5G communications, and sparse signal recovery.

He was an associate editor for the IEEE SIGNAL PROCESSING LETTERS during 2012-16, SADHANA JOURNAL during 2016-18, and IEEE TRANSACTIONS ON COMMUNICATIONS during 2017-22. He was an elected member of the IEEE SPCOM Technical Committee for the years 2014-19. He is currently serving as an area editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and as an associate editor for the IEEE TRANSACTIONS ON INFORMATION THEORY.